Ricerca Operativa

Exercise Solutions

Lucas Simonetti

Università degli Studi di Udine · A.A. 2025–2026

Contents

CHAPTER 1

Introduction & Modelling

Exercise 1. Define *Operations Research* in your own words. What distinguishes it from pure mathematics and from computer science?

Solution. Operations Research (OR) is the discipline of applying advanced analytical methods (such as mathematical modeling, optimization, and statistics) to help make better decisions in complex systems. It differs from:

- **Pure mathematics**: which focuses on abstract structures, theorems, and proofs for their own sake, whereas OR is inherently applied and focused on decision-making in real-world systems.
- **Computer science**: which focuses on the design, analysis, and implementation of algorithms, software, and hardware. OR uses computer science as a tool to solve models, but its core concern is the modeling of decisions and system optimization.

Exercise 2. True or false: "Operations Research always requires a computer to find a solution." Justify your answer.

Solution. False. While computers are practically essential for solving large-scale, real-world instances, simple optimization problems can be solved analytically or by hand using methods like the graphical method for 2-variable linear programs, or the manual execution of the simplex algorithm on small tableaus.

Exercise 3. List three real-world domains in which OR techniques are routinely applied, and for each domain name one concrete decision that OR helps to optimize.

Solution. Three domains and optimized decisions:

1. **Logistics and Transportation**: Optimizing vehicle routing to minimize delivery costs and travel time.
2. **Healthcare**: Optimizing nurse scheduling to cover demand under shift constraints while minimizing labor cost.

3. **Manufacturing**: Optimizing production scheduling to decide which products to manufacture in what quantities to maximize profit.

Exercise 4. What are the three main components of a *mathematical programming model*? Briefly explain the role of each component.

Solution. The three main components are:

1. **Decision Variables**: Mathematical symbols representing the decisions to be made (e.g., quantities to produce).

2. **Objective Function**: A mathematical expression of the goal (e.g., maximize profit, minimize cost) written in terms of the decision variables.

3. **Constraints**: Inequalities or equations that restrict the values the decision variables can take, representing resource limits or physical rules.

Exercise 5. Define the term *instance* of a problem and explain how it differs from the problem itself. Give one example from operations research.

Solution. A **problem** is a generic description of a class of optimization situations (containing parameters like cost, demand, weights). An **instance** is a specific case of the problem where all parameters are assigned concrete numerical values.

Example: The Knapsack Problem is the general problem. A specific knapsack instance has a capacity of $W = 10$ kg and 3 items with weights $w = [3, 5, 2]$ and values $v = [10, 20, 15]$.

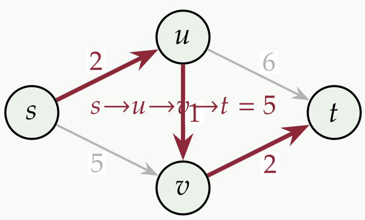
Exercise 6. The *Shortest Path Problem* asks, given a weighted directed graph $G = (V, A)$ and two nodes $s, t \in V$, for a minimum-cost directed path from s to t.

(a) Write down a specific instance of this problem (you may use a small graph with $|V| \leq 5$).

(b) Identify which part of your description belongs to the *problem* and which part belongs to the *instance*.

Solution.

(a) Let $V = \{s, u, v, t\}$, $A = \{(s, u), (s, v), (u, v), (u, t), (v, t)\}$, with arc weights $c(s, u) = 2$, $c(s, v) = 5$, $c(u, v) = 1$, $c(u, t) = 6$, $c(v, t) = 2$. We seek the shortest path from s to t.



(b) **Problem part**: The definition of nodes, arcs, arc costs, and the goal of finding a path of minimum total weight from s to t.
Instance part: The specific set of 4 vertices, 5 arcs, and their numerical costs.

Exercise 7. Explain why a single algorithm is said to *solve a problem* rather than solve an instance, even though the algorithm operates on a specific input each time it is run.

Solution. An algorithm is a generic, step-by-step procedure designed to find the optimal solution for *any* valid input instance of the problem. It is defined in terms of the problem parameters (e.g., Dijkstra's algorithm works for any graph with non-negative edge costs), meaning its correctness holds universally across all possible instances of that problem.

Exercise 8. True or false: "Two instances of the same problem can have different feasible sets." Justify your answer with an example.

Solution. True. Feasible sets depend on the instance parameters.

Example: In the Knapsack Problem, if instance 1 has knapsack capacity $W = 5$ and items of weight $w = [3, 4]$, the feasible set of binary selection vectors is $\{(0,0), (1,0), (0,1)\}$. If instance 2 has capacity $W = 10$ and same weights, the feasible set is $\{(0,0), (1,0), (0,1), (1,1)\}$, which includes the combined selection.

Exercise 9. Formally define an *optimization problem*. Your definition must include: the feasible set X, the objective function f, and the notion of an optimal solution.

Solution. An optimization problem is a triple (X, f, goal), where:

- $X \subseteq \mathbb{R}^n$ (or some discrete domain) is the **feasible set** representing all valid solutions.
- $f : X \to \mathbb{R}$ is the **objective function** mapping each solution to a real value.
- $\text{goal} \in \{\text{minimize}, \text{maximize}\}$.

An **optimal solution** is a point $x^* \in X$ such that:

- $f(x^*) \leq f(x)$ for all $x \in X$ (if goal is minimize).
- $f(x^*) \geq f(x)$ for all $x \in X$ (if goal is maximize).

Exercise 10. Consider the following optimization problem:

$$\max\ 3x_1 + 2x_2 \quad \text{s.t.} \quad x_1 + x_2 \leq 4,\ x_1, x_2 \geq 0.$$

(a) Write the feasible set X in set-builder notation.

(b) Find an optimal solution by inspection and state the optimal value.

Solution.

(a) $X = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 + x_2 \leq 4,\ x_1 \geq 0,\ x_2 \geq 0\}$.

(b) Because the coefficient of x_1 (3) is larger than the coefficient of x_2 (2), we should make x_1 as large as possible within the constraint $x_1 + x_2 \leq 4$. Setting $x_1 = 4$ and $x_2 = 0$ is feasible. The objective value is $3(4) + 2(0) = 12$. Any other point (x_1, x_2) has $3x_1 + 2x_2 \leq 3x_1 + 3x_2 = 3(x_1 + x_2) \leq 12$. So $(x_1^*, x_2^*) = (4, 0)$ is optimal with value $z^* = 12$.

Exercise 11. The *furniture workshop* produces tables (x_1) and chairs (x_2). Each table requires 3 hours of carpentry and 2 hours of finishing; each chair requires 2 hours of carpentry and 1 hour of finishing. Available capacity is 120 carpentry hours and 70 finishing hours per week. Profit is \$90 per table and \$60 per chair.

(a) Write the LP formulation (decision variables, objective, constraints).

(b) Identify the type of the model (LP, ILP, MILP, or NLP).

Solution.

(a) Let $x_1 \geq 0$ be the number of tables produced per week, and $x_2 \geq 0$ the number of chairs.

$$\begin{aligned} \text{maximize} \quad & 90x_1 + 60x_2 \\ \text{subject to} \quad & 3x_1 + 2x_2 \leq 120 \quad \text{(Carpentry hours)} \\ & 2x_1 + x_2 \leq 70 \quad \text{(Finishing hours)} \\ & x_1, x_2 \geq 0 \end{aligned}$$

(b) Assuming table and chair production can be fractional (e.g., average rates per week), the model is a **Linear Program (LP)**. If we enforce integer values, it is an **Integer Linear Program (ILP)**.

Exercise 12. What is the difference between a *feasible solution* and an *optimal solution*? Is every optimal solution feasible? Is every feasible solution optimal?

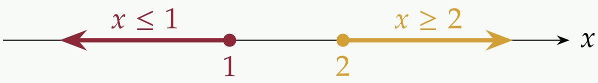
Solution. A **feasible solution** is any point that satisfies all constraints of the model. An **optimal solution** is a feasible solution that achieves the best possible objective function value among all feasible points.

- **Yes**, by definition, an optimal solution must belong to the feasible set X.
- **No**, a feasible solution does not necessarily have the best objective value (e.g., $(0, 0)$ is feasible in the furniture workshop but not optimal).

Exercise 13. True or false: "An optimization problem always has at least one feasible solution." Justify with a brief example or counterexample.

Solution. False. Some problems are infeasible (their feasible set is empty).

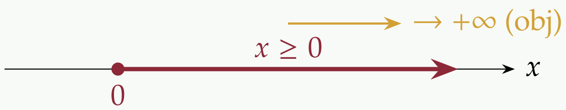
Counterexample: $\max x$ s.t. $x \leq 1, x \geq 2$. No real number is simultaneously ≤ 1 and ≥ 2.



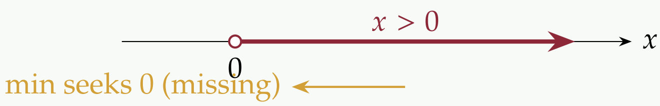
Exercise 14. True or false: "An optimization problem with a non-empty feasible set always has at least one optimal solution." Justify with a brief example or counterexample.

Solution. False. The problem can be unbounded, or the feasible set is open and the supremum/infimum is not attained.

Counterexample 1 (Unbounded): $\max x$ s.t. $x \geq 0$. The objective value can grow to infinity; there is no finite optimal solution.



Counterexample 2 (Open set): $\min x$ s.t. $x > 0$. The infimum is 0, but 0 is not in the feasible set. No optimal solution exists.



Exercise 15. Under what conditions on the feasible set X and the objective function f is an optimal solution guaranteed to exist for a minimization problem? (Hint: recall Weierstrass theorem from calculus).

Solution. By the extreme value theorem (Weierstrass theorem), an optimal solution is guaranteed to exist if:

1. The feasible set X is **compact** (closed and bounded in $\mathbb{R}^n$).
2. The objective function $f : X \to \mathbb{R}$ is **continuous**.

Exercise 16. Describe the difference between *local optimality* and *global optimality*. Under what circumstances is a local optimum guaranteed to be a global optimum?

Solution. A feasible solution x^* is a **local optimum** if it has the best objective value in its immediate neighborhood (i.e., $f(x^*) \leq f(x)$ for all feasible x within a distance $\epsilon > 0$). It is a **global optimum** if it has the best objective value over the entire feasible set X.

A local optimum is guaranteed to be a global optimum in **convex optimization**:

- The feasible set X is convex.

- The objective function f to minimize is convex (or concave if maximizing).

Exercise 17. Classify the following optimization problems into LP, ILP, MILP, or NLP. Justify your classification.

(a) $\min 3x_1 - 2x_2$ s.t. $x_1^2 + x_2 \leq 5$, $x_1, x_2 \geq 0$.

(b) $\max 5x_1 + 4x_2$ s.t. $x_1 + x_2 \leq 10$, $x_1 \geq 0$ integer, $x_2 \geq 0$ continuous.

(c) $\min \sum_{j=1}^{n} c_j x_j$ s.t. $x_j \in \{0, 1\}$.

Solution.

(a) **NLP (Nonlinear Program)**: The constraint contains a nonlinear term x_1^2.

(b) **MILP (Mixed Integer Linear Program)**: The objective and constraints are linear, but variable x_1 must be integer while x_2 is continuous.

(c) **ILP (Integer Linear Program)**: The objective and constraints are linear, and all variables x_j are restricted to be binary (integers).

Exercise 18. Why are nonlinear programming (NLP) problems generally harder to solve to global optimality than linear programming (LP) problems?

Solution. In LPs, the feasible region is a convex polytope and the objective is linear (hence convex), meaning any local optimum is guaranteed to be a global optimum, and the optimum always lies on a vertex. In NLPs, the feasible region and/or objective function can be non-convex, which introduces multiple local optima, saddle points, and flat regions. Algorithms can easily get trapped in a local optimum with no simple way to verify if a better solution exists elsewhere.

Exercise 19. Define the *linear relaxation* of an integer linear program. Why is it useful?

Solution. The **linear relaxation** (or LP relaxation) of an ILP is the linear program obtained by removing the integrality constraints on all variables (e.g., replacing $x_j \in \{0, 1\}$ with $0 \leq x_j \leq 1$, or $x_i \in \mathbb{Z}$ with $x_i \in \mathbb{R}$).

It is useful because:

- It can be solved very quickly using algorithms like Simplex or Barrier methods.
- Its optimal value provides a valid mathematical bound (upper bound for maximization, lower bound for minimization) on the optimal integer value.
- It forms the basis of exact algorithms like Branch and Bound and Cutting Planes.

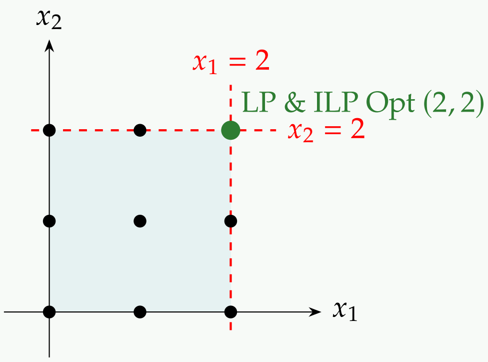
Exercise 20. Give an example of a 2-variable ILP whose LP relaxation has an integer optimal solution. Draw the feasible region and the LP optimum.

Solution. Consider the ILP:

$$\max\ x_1 + x_2 \quad \text{s.t.} \quad x_1 \le 2,\ x_2 \le 2,\ x_1, x_2 \ge 0 \text{ integer}.$$

Its LP relaxation has the unique optimal solution $(x_1^*, x_2^*) = (2, 2)$, which is integer.

Below is the TikZ visualization of the feasible region (dotted points are the integer solutions, shaded region is the LP relaxation):



Exercise 21. Give an example of a 2-variable ILP where rounding the optimal solution of the LP relaxation to the nearest integer yields an *infeasible* solution.

Solution. Consider the ILP:

$$\max\ x_2 \quad \text{s.t.} \quad 2x_1 + 2x_2 \le 3,\ x_1, x_2 \in \{0, 1\}.$$

The LP relaxation allows $0 \le x_1, x_2 \le 1$. The optimal LP solution is $(x_1^*, x_2^*) = (0, 1.5)$, which has objective value 1.5. Rounding the LP solution to the nearest integer yields $(0, 2)$ or $(0, 1)$. If we round to nearest integer $(0, 2)$, this violates $2x_1 + 2x_2 \le 3$ since $2(0) + 2(2) = 4 > 3$. Thus, rounding yields an infeasible point.

Exercise 22. A student claims: "If we round the LP relaxation solution to the nearest feasible integer point, we always get the optimal solution of the ILP." Disprove this claim with a counterexample.

Solution. Consider the ILP:

$$\max\ x_1 + 11x_2 \quad \text{s.t.} \quad x_1 + 10x_2 \le 10,\ x_1 \le 9,\ x_1, x_2 \ge 0 \text{ integer}.$$

The LP relaxation optimum is at $(9, 0.1)$ with objective value $9 + 11(0.1) = 10.1$. The nearest feasible integer point is $(9, 0)$ with objective value 9. The optimal integer solution is $(0, 1)$ with objective value 11, which is far from $(9, 0)$ and has a much better objective value. This disproves the student's claim.

Exercise 23. What is the difference between a *decision problem* and an *optimization problem*? Explain how one can use an algorithm for the decision version to solve the optimization version.

Solution.

- An **optimization problem** asks for the best feasible solution (e.g., "Find a path from s to t of minimum cost z").
- A **decision problem** asks a yes/no question (e.g., "Is there a path from s to t of cost at most K?").

We can solve the optimization version using a decision algorithm by performing a **binary search** on the objective value K. If the optimal value is bounded within $[L, U]$, we query the decision algorithm for $K = (L + U)/2$. If "yes", we set $U = K$; if "no", we set $L = K$. We repeat this until the interval is small enough (or exactly, if values are discrete integers).

Exercise 24. Why is the binary search approach for solving optimization problems via decision problems only practical when the objective value lies in a bounded range?

Solution. Binary search requires starting with a finite interval $[L, U]$. If the objective value is unbounded (e.g., can go to infinity), we cannot directly choose a midpoint. Furthermore, the number of steps in binary search is proportional to $\log(U - L)$. If $U - L$ is double-exponentially large or infinite, the search will not terminate in polynomial time.

Exercise 25. True or false: "If a decision problem is NP-complete, then the corresponding optimization problem is NP-hard." Justify your answer.

Solution. True. The optimization problem is at least as hard as the decision problem. If we had a polynomial-time algorithm for the optimization problem (e.g., finding the minimum cost path or maximum clique), we could solve the decision problem in polynomial time by simply finding the optimal value and comparing it to K. Therefore, NP-completeness of the decision version implies NP-hardness of the optimization version.

Exercise 26. A manager says: "Since our scheduling problem is NP-hard, we should give up on finding any solution and just guess." Critique this statement from an operations research perspective.

Solution. This statement is incorrect. NP-hardness means that finding the *guaranteed optimal* solution for *arbitrary, large-scale* instances in polynomial time is highly unlikely. However, OR offers several powerful alternatives:

1. **Heuristics and Metaheuristics** (e.g., Genetic Algorithms, Tabu Search, Simulated Annealing) which find very high-quality (often near-optimal) solutions quickly.
2. **Approximation Algorithms** which run in polynomial time and guarantee a solution within a provable factor of the optimum.
3. **Exact Methods** (like Branch and Bound, MIP solvers) which often solve real-world instances to optimality very quickly despite

theoretical worst-case complexity.

Exercise 27. Formulate the following statement as a linear constraint: "Either factory A must be open, or factory B must be open, or both." (Use binary variables $y_A, y_B \in \{0, 1\}$ where 1 means open).

Solution. The requirement is that at least one of y_A, y_B must be 1. This is written as:

$$y_A + y_B \geq 1.$$

Exercise 28. Formulate: "Factory A can only be open if factory B is open."

Solution. If factory A is open ($y_A = 1$), then B must be open ($y_B = 1$). If A is closed ($y_A = 0$), B can be open or closed. This logical implication $y_A \implies y_B$ is written as:

$$y_A \leq y_B.$$

Exercise 29. Formulate: "If factory A is open, then factory B must be closed."

Solution. If $y_A = 1$, we must have $y_B = 0$. We cannot have both $y_A = 1$ and $y_B = 1$. This is written as:

$$y_A + y_B \leq 1.$$

Exercise 30. Formulate: "Exactly two of the four factories $\{A, B, C, D\}$ must be open." (Use binary variables y_A, y_B, y_C, y_D).

Solution. The sum of the binary variables must equal 2:

$$y_A + y_B + y_C + y_D = 2.$$

Exercise 31. Formulate: "At most three of the five factories $\{A, B, C, D, E\}$ can be open."

Solution. The sum of the binary variables must be less than or equal to 3:

$$y_A + y_B + y_C + y_D + y_E \leq 3.$$

Exercise 32. Suppose we have a production variable $x \geq 0$ (continuous) and a binary variable $y \in \{0, 1\}$ indicating whether the factory is open. Write a constraint that ensures $x = 0$ when the factory is closed ($y = 0$), and permits x to be at most M when open ($y = 1$).

Solution. This is a classic "Big-M" linking constraint:

$$x \leq My.$$

If $y = 0$, the constraint forces $x \leq 0$, which since $x \geq 0$ implies $x = 0$. If $y = 1$, the constraint becomes $x \leq M$, permitting production up to the capacity limit M.

Exercise 33. For the directed graph in the previous exercise, assign the following arc capacities: $c(s, u) = 3$, $c(s, v) = 2$, $c(u, v) = 1$, $c(u, t) = 2$, $c(v, t) = 3$. By inspection, find the maximum flow from s to t.

Solution. By inspection, we can route:

- 2 units along the path $s \to u \to t$ (capacity is limited by $c(u, t) = 2$).
- 2 units along the path $s \to v \to t$ (capacity is limited by $c(s, v) = 2$ and $c(v, t) = 3$, so we route 2).

Total flow is $2 + 2 = 4$ units. The remaining capacities are: $s \to u$ has 1 unit capacity left, but $u \to t$ is saturated. Routing flow $s \to u \to v \to t$ is not possible because $s \to v$ is already saturated at 2. Hence, the maximum flow from s to t is **4**.

Exercise 34. Write down the LP formulation for the maximum flow problem instance described in the previous exercise.

Solution. Let $f_{ij} \geq 0$ be the flow on arc $(i, j) \in A$.

$$\begin{array}{rll} \text{maximize} & f_{ut} + f_{vt} & \\ \text{subject to} & f_{su} = f_{uv} + f_{ut} & \text{(Flow conservation at } u) \\ & f_{sv} + f_{uv} = f_{vt} & \text{(Flow conservation at } v) \\ & f_{su} \leq 3, \ f_{sv} \leq 2 & \\ & f_{uv} \leq 1 & \\ & f_{ut} \leq 2, \ f_{vt} \leq 3 & \\ & f_{ij} \geq 0 & \forall (i, j) \in A \end{array}$$

Exercise 35. Consider a knapsack problem with 6 items. The weights are $w = [2, 3, 4, 5, 6, 7]$ and the values are $v = [3, 4, 5, 8, 10, 11]$. The knapsack capacity is $W = 15$. Formulate this as an ILP.

Solution. Let $x_j \in \{0, 1\}$ be a binary variable where $x_j = 1$ if item j is selected, and 0 otherwise, for $j \in \{1, 2, 3, 4, 5, 6\}$.

$$\begin{array}{rl} \text{maximize} & 3x_1 + 4x_2 + 5x_3 + 8x_4 + 10x_5 + 11x_6 \\ \text{subject to} & 2x_1 + 3x_2 + 4x_3 + 5x_4 + 6x_5 + 7x_6 \leq 15 \\ & x_j \in \{0, 1\} \quad \forall j \in \{1, \ldots, 6\} \end{array}$$

Exercise 36. For the 6-item knapsack instance in the previous exercise, compute the value-to-weight ratios of the items and sort them in descending order.

Solution. We compute $r_j = v_j / w_j$ for each item j:

- Item 1: $r_1 = 3/2 = 1.5$
- Item 2: $r_2 = 4/3 \approx 1.33$
- Item 3: $r_3 = 5/4 = 1.25$

- Item 4: $r_4 = 8/5 = 1.6$
- Item 5: $r_5 = 10/6 \approx 1.67$
- Item 6: $r_6 = 11/7 \approx 1.57$

Sorting the items in descending order of their ratios:

Item 5 (1.67) > Item 4 (1.6) > Item 6 (1.57) > Item 1 (1.5) > Item 2 (1.33) > Item 3 (1.25).

Exercise 37. Use the greedy heuristic (select items in descending order of ratio until no more fit) to find a feasible solution for the knapsack instance. What is its total value and weight?

Solution. Using the sorted order: Item 5, Item 4, Item 6, Item 1, Item 2, Item 3. Capacity $W = 15$.

- Select **Item 5** ($w_5 = 6$, remaining capacity $15 - 6 = 9$). Value = 10.
- Select **Item 4** ($w_4 = 5$, remaining capacity $9 - 5 = 4$). Value = $10 + 8 = 18$.
- Consider Item 6 ($w_6 = 7 > 4$): does not fit.
- Select **Item 1** ($w_1 = 2$, remaining capacity $4 - 2 = 2$). Value = $18 + 3 = 21$.
- Consider Item 2 ($w_2 = 3 > 2$): does not fit.
- Consider Item 3 ($w_3 = 4 > 2$): does not fit.

The greedy solution selects **Items {1, 4, 5}**.
Total weight = $2 + 5 + 6 = 13 \leq 15$.
Total value = $3 + 8 + 10 =$ **21**.

Exercise 38. Formulate the *fractional knapsack problem* (where items can be subdivided) as a linear program. Find the optimal solution for the 6-item instance.

Solution. In the fractional knapsack problem, we let $x_j \in [0, 1]$ represent the fraction of item j selected:

$$\begin{aligned} \text{maximize} \quad & 3x_1 + 4x_2 + 5x_3 + 8x_4 + 10x_5 + 11x_6 \\ \text{subject to} \quad & 2x_1 + 3x_2 + 4x_3 + 5x_4 + 6x_5 + 7x_6 \leq 15 \\ & 0 \leq x_j \leq 1 \quad \forall j \in \{1, \ldots, 6\} \end{aligned}$$

To solve it, we greedily select items in descending order of ratio:

- Select $x_5 = 1$ (weight used = 6, remaining = 9)
- Select $x_4 = 1$ (weight used = $6 + 5 = 11$, remaining = 4)
- Select fraction of Item 6: since $w_6 = 7 > 4$, we take $x_6 = 4/7$.

The remaining capacity is 0, so $x_1 = x_2 = x_3 = 0$.

Optimal fractional solution: $x^* = (0, 0, 0, 1, 1, 4/7)$.
Total value $= 10(1) + 8(1) + 11(4/7) = 18 + 44/7 = 170/7 \approx$ **24.29**.

Exercise 39. For the LP formulation you wrote in the previous exercise:

(a) Which constraints are active at the optimal solution?

(b) How does the optimal value of the fractional knapsack compare to the greedy integer solution?

Solution.

(a) The active (binding) constraints are:

- The knapsack capacity constraint: $2x_1 + 3x_2 + 4x_3 + 5x_4 + 6x_5 + 7x_6 = 15$.
- The upper bounds for selected items: $x_4 = 1$, $x_5 = 1$.
- The lower bounds for unselected items: $x_1 = 0$, $x_2 = 0$, $x_3 = 0$.

(b) The optimal fractional value (24.29) is strictly greater than the greedy integer value (21.0). This is expected since the fractional LP relaxation relaxes the integrality constraints, expanding the feasible region.

Exercise 40. Formulate the *Facility Location Problem*: we must decide which warehouses to open from a set J (opening cost f_j) and how to serve customers I from open warehouses (shipping cost c_{ij}). Each customer must be served by exactly one warehouse.

Solution. Let $y_j \in \{0, 1\}$ be 1 if warehouse $j \in J$ is opened, 0 otherwise. Let $x_{ij} \in \{0, 1\}$ (or $x_{ij} \geq 0$) be the fraction of customer i's demand served by warehouse j.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{subject to} \quad & \sum_{j \in J} x_{ij} = 1 & \forall\, i \in I \\
& x_{ij} \leq y_j & \forall\, i \in I,\ j \in J \\
& y_j \in \{0, 1\} & \forall\, j \in J \\
& x_{ij} \geq 0 & \forall\, i \in I,\ j \in J
\end{aligned}
$$

Exercise 41. Explain the role of the constraint $x_{ij} \leq y_j$ in the facility location formulation. What happens if this constraint is omitted?

Solution. The constraint $x_{ij} \leq y_j$ ensures that customer i can only receive goods from warehouse j if warehouse j is actually opened ($y_j = 1$). If $y_j = 0$, the constraint forces $x_{ij} = 0$.

If this constraint is omitted, the model will minimize shipping costs by assigning customers to warehouses with the lowest shipping costs c_{ij} without opening them (leaving $y_j = 0$), thus avoiding the opening costs f_j. This would yield an invalid, physically impossible solution.

Exercise 42. Formulate the *Uncapacitated Facility Location Problem* (UFL) where warehouses have unlimited capacity.

Solution. The formulation given in the solution to Exercise 41 is exactly the Uncapacitated Facility Location Problem. Because there are no constraints limiting the total demand served by a warehouse j (e.g., $\sum_{i \in I} d_i x_{ij} \leq K_j y_j$), each warehouse is assumed to have unlimited capacity.

Exercise 43. Formulate the *Capacitated Facility Location Problem* (CFL) where each warehouse j has a maximum capacity K_j.

Solution. Let d_i be the demand of customer $i \in I$. We add a capacity constraint for each warehouse $j \in J$:

$$\begin{aligned}
\text{minimize} \quad & \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \\
\text{subject to} \quad & \sum_{j \in J} x_{ij} = 1 && \forall\, i \in I \\
& \sum_{i \in I} d_i x_{ij} \leq K_j y_j && \forall\, j \in J \\
& x_{ij} \leq y_j && \forall\, i \in I,\ j \in J \\
& y_j \in \{0, 1\} && \forall\, j \in J \\
& x_{ij} \geq 0 && \forall\, i \in I,\ j \in J
\end{aligned}$$

Exercise 44. In the CFL formulation, is the constraint $x_{ij} \leq y_j$ redundant if we already have $\sum_{i \in I} d_i x_{ij} \leq K_j y_j$? Explain.

Solution. Mathematically, $x_{ij} \leq y_j$ is redundant because if $y_j = 0$, then $\sum_i d_i x_{ij} \leq 0 \implies x_{ij} = 0$ (since $d_i > 0, x_{ij} \geq 0$).

However, in practice, keeping $x_{ij} \leq y_j$ is highly recommended because it provides a much tighter LP relaxation. Solver algorithms (like Branch and Bound) can solve the model much faster because the individual constraints $x_{ij} \leq y_j$ restrict the fractional values of y_j much more effectively than the aggregated capacity constraint.

Exercise 45. Formulate the following logical condition: "If we open warehouse A, we cannot open warehouse B."

Solution. This means that at most one of warehouse A or B can be opened, which translates to:

$$y_A + y_B \leq 1.$$

Exercise 46. A small airline must assign pilots to routes. There are m routes and n pilots; each pilot j is qualified for a subset $Q_j \subseteq \{1, \ldots, m\}$ of routes. Each route must be covered by exactly one pilot. Formulate a *set-cover* integer program for this assignment.

Solution. Let $x_{ij} \in \{0,1\}$ be a binary variable where $x_{ij} = 1$ if pilot $j \in \{1,\dots,n\}$ is assigned to route $i \in \{1,\dots,m\}$, and 0 otherwise.

We must cover each route exactly once, and a pilot can only be assigned to a route if they are qualified.

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & \sum_{j:i\in Q_j} x_{ij} = 1 \quad \forall\, i \in \{1,\dots,m\} \\ & x_{ij} = 0 \qquad\quad \forall\, j \in \{1,\dots,n\},\ i \notin Q_j \\ & x_{ij} \in \{0,1\} \quad\ \forall\, i,j \end{array}$$

Alternatively, we can define binary $z_j = 1$ if pilot j is employed/selected, and minimize the number of pilots:

$$\begin{array}{lll} \text{minimize} & \sum_{j=1}^{n} z_j & \\ \text{subject to} & \sum_{j:i\in Q_j} x_{ij} = 1 & \forall\, i \in \{1,\dots,m\} \\ & x_{ij} \le z_j & \forall\, i,j \\ & x_{ij} = 0 & \forall\, j,\ i \notin Q_j \\ & z_j,\ x_{ij} \in \{0,1\} & \end{array}$$

Exercise 47. A company ships goods from two warehouses (W_1, W_2) to three customers (C_1, C_2, C_3). Supply at W_1 is 80 units and at W_2 is 60 units. Demand at C_1, C_2, C_3 is 50, 40, and 50 units respectively. Shipping cost per unit from W_i to C_j is given by the matrix:

$$\begin{pmatrix} 2 & 3 & 1 \\ 5 & 4 & 8 \end{pmatrix}.$$

Formulate this as a linear program (the *transportation problem*).

Solution. Let $x_{ij} \ge 0$ be the number of units shipped from warehouse W_i to customer C_j, for $i \in \{1,2\}$ and $j \in \{1,2,3\}$.

The objective is to minimize total shipping cost:

$$\begin{array}{lll} \text{minimize} & 2x_{11} + 3x_{12} + x_{13} + 5x_{21} + 4x_{22} + 8x_{23} & \\ \text{subject to} & x_{11} + x_{12} + x_{13} \le 80 & (\text{Supply } W_1) \\ & x_{21} + x_{22} + x_{23} \le 60 & (\text{Supply } W_2) \\ & x_{11} + x_{21} \ge 50 & (\text{Demand } C_1) \\ & x_{12} + x_{22} \ge 40 & (\text{Demand } C_2) \\ & x_{13} + x_{23} \ge 50 & (\text{Demand } C_3) \\ & x_{ij} \ge 0 & \forall\, i,j \end{array}$$

Exercise 48. Formulate the *maximum weight independent set* problem on an undirected graph $G = (V,E)$ with node weights $w_v \ge 0$ as an integer linear program. (A set $S \subseteq V$ is *independent* if no two nodes in S are adjacent.)

Solution. Let $x_v \in \{0, 1\}$ be a binary variable where $x_v = 1$ if node $v \in V$ is selected in the independent set, and 0 otherwise.

We want to maximize total weight, subject to the constraint that we cannot select both endpoints of any edge $\{u, v\} \in E$:

$$\begin{aligned} \text{maximize} \quad & \sum_{v \in V} w_v x_v \\ \text{subject to} \quad & x_u + x_v \leq 1 \quad \forall \{u, v\} \in E \\ & x_v \in \{0, 1\} \quad \forall v \in V \end{aligned}$$

Exercise 49. A project consists of n tasks. Task j has a duration d_j and a set P_j of *predecessor tasks* that must all finish before task j can start. Define start time variables $s_j \geq 0$ and formulate the problem of minimizing the project makespan as a linear program.

Solution. Let $s_j \geq 0$ be the start time of task $j \in \{1, \ldots, n\}$. Let T be a continuous variable representing the project makespan (the completion time of the entire project).

For each task j, its completion time is $s_j + d_j$. The project makespan T must be at least the completion time of every task:

$$T \geq s_j + d_j \quad \forall j \in \{1, \ldots, n\}.$$

Furthermore, task j cannot start until all its predecessor tasks $i \in P_j$ are finished:

$$s_j \geq s_i + d_i \quad \forall j \in \{1, \ldots, n\},\ i \in P_j.$$

The linear program is:

$$\begin{aligned} \text{minimize} \quad & T \\ \text{subject to} \quad & T \geq s_j + d_j \quad \forall j \in \{1, \ldots, n\} \\ & s_j \geq s_i + d_i \quad \forall j \in \{1, \ldots, n\},\ i \in P_j \\ & s_j \geq 0 \quad \forall j \in \{1, \ldots, n\} \end{aligned}$$

Exercise 50. Explain the difference between *continuous* and *discrete* optimization. Give one example of each type from the running examples in this chapter.

Solution.

- **Continuous Optimization**: The decision variables can take any real value within a continuous interval (e.g., $x \in \mathbb{R}$). Feasible regions are typically smooth, continuous spaces.

  Example: The Transportation Problem (Exercise 47), where shipping quantities can be fractional values.

- **Discrete Optimization**: Some or all of the decision variables are restricted to take values from a discrete set (such as integers or binary choices, e.g., $x \in \mathbb{Z}$ or $x \in \{0, 1\}$). Feasible regions consist of isolated points.

Example: The Knapsack Problem (Exercise 35), where we must make binary yes/no choices on item selections.

Exercise 51. An objective function $f(x) = c^\top x$ is *linear*. Explain why replacing it with $f(x) = \|x\|_2^2$ changes the problem class from LP to NLP. Does the feasible set change?

Solution. Replacing the objective with $f(x) = \|x\|_2^2 = \sum_j x_j^2$ makes the objective function quadratic (quadratic terms like x_j^2 are nonlinear). By definition, a Linear Program must have a linear objective function. A nonlinear objective function shifts the problem into the class of **Nonlinear Programs (NLP)**.

The feasible set X does **not** change, because we only altered the objective function, leaving the linear constraints defining the feasible region unchanged.

Exercise 52. Consider the following claim: "Solving the decision version of an NP-hard problem is easier than solving its optimization version." Argue for or against this claim, providing a formal justification.

Solution. We argue **against** the claim in terms of polynomial-time complexity classes (P vs. NP). In computational complexity theory, both versions are in the same complexity class under polynomial-time Turing reductions:

- The decision version is NP-complete, which means it cannot be solved in polynomial time unless $P = NP$.
- If we had an algorithm that solves the decision version in polynomial time, we could solve the optimization version in polynomial time by binary search (as shown in Exercise 22), which requires only a polynomial number of calls to the decision oracle.
- Thus, if the decision version is "easy" (P), the optimization version is "easy" (P). If the decision version is "hard" (NP-complete), the optimization version is "hard" (NP-hard).

Therefore, from a formal complexity class perspective, one is not "easier" than the other; they are computationally equivalent.

Exercise 53. Let z^*_{LP} denote the optimal value of the LP relaxation of an ILP, and z^*_{ILP} the optimal value of the ILP itself (both are maximization problems). Prove or disprove: $z^*_{\text{LP}} \geq z^*_{\text{ILP}}$.

Solution. Proof. Let X_{ILP} be the feasible region of the ILP, and X_{LP} be the feasible region of its LP relaxation. By definition, $X_{\text{ILP}} \subseteq X_{\text{LP}}$ because any solution that satisfies the integrality constraints also satisfies the relaxed continuous constraints.

Since we are maximizing the same objective function $f(x) = c^\top x$ over both sets, and X_{ILP} is a subset of X_{LP}:

$$z^*_{\text{ILP}} = \max_{x \in X_{\text{ILP}}} f(x) \leq \max_{x \in X_{\text{LP}}} f(x) = z^*_{\text{LP}}.$$

Thus, $z^*_{\text{LP}} \geq z^*_{\text{ILP}}$ must hold.

Exercise 54. Suppose an optimization problem over a finite feasible set has 2^{50} feasible solutions. Explain why exhaustive enumeration is infeasible in practice, and discuss what OR provides as an alternative.

Solution. The size $2^{50} \approx 1.125 \times 10^{15}$ is an astronomical number of solutions. Even if a computer could evaluate 1 billion (10^9) solutions per second, it would take:

$$\frac{1.125 \times 10^{15}}{10^9 \text{ sec}} = 1.125 \times 10^6 \text{ seconds} \approx 13 \text{ days}.$$

While 13 days is theoretically possible, larger sizes like 2^{100} would take billions of years.

OR provides several alternative approaches:

1. **Implicit Enumeration** (e.g., Branch and Bound): Instead of checking every solution, it uses bounds from LP relaxations to prune large subtrees of the solution space that cannot possibly contain a better solution.
2. **Duality and Cutting Planes**: Adding mathematical inequalities to shrink the continuous feasible set toward the convex hull of integer solutions, identifying the optimum without scanning individual points.
3. **Local Search and Metaheuristics**: Navigating the solution space strategically via local improvements rather than scanning the whole space.

Exercise 55. Define what it means for a constraint to be *binding* (or *active*) at a feasible solution x^0. For the LP

$$\max\ 2x_1 + x_2 \quad \text{s.t.} \quad x_1 + x_2 \leq 5,\ x_1 \leq 3,\ x_1, x_2 \geq 0,$$

identify which constraints are binding at the point $(3, 2)$.

Solution. A constraint $g(x) \leq b$ is **binding** (or active) at a feasible solution x^0 if it holds with strict equality at that point, i.e., $g(x^0) = b$.

Let's check the constraints at $(3, 2)$:

1. $x_1 + x_2 \leq 5$: at $(3, 2)$, $3 + 2 = 5$. Since $5 = 5$, this constraint is **binding**.
2. $x_1 \leq 3$: at $(3, 2)$, $3 = 3$. Since $3 = 3$, this constraint is **binding**.
3. $x_1 \geq 0$: at $(3, 2)$, $3 > 0$. Not binding.
4. $x_2 \geq 0$: at $(3, 2)$, $2 > 0$. Not binding.

Thus, the binding constraints at $(3, 2)$ are $x_1 + x_2 \leq 5$ and $x_1 \leq 3$.

Exercise 56. Explain how OR can be used to support *multi-objective* decisions where a decision-maker wants to simultaneously minimize cost and maximize service level. Why is there generally no single optimal solution in such settings?

Solution. In multi-objective optimization, objectives are often in conflict: improving one (e.g., service level) worsens the other (e.g., cost). Therefore, there is no single solution that simultaneously optimizes all objectives. Instead, OR identifies the set of **Pareto-optimal** (or non-dominated) solutions. A solution is Pareto-optimal if no objective can be improved without worsening at least one other objective.

OR support methods include:

1. **Weighted Sum Method**: Combining objectives into a single function: $\min w_1 \cdot \text{Cost} - w_2 \cdot \text{Service}$, varying the weights to explore trade-offs.

2. **ϵ-Constraint Method**: Optimizing one objective (e.g., minimize Cost) while setting the others as constraints (e.g., Service Level $\geq \epsilon$).

CHAPTER 2

Linear Programming

Exercise 1. Consider the LP

$$\max\ 3x_1 + 2x_2 \quad \text{s.t.} \quad x_1 + x_2 \le 4,\ 2x_1 + x_2 \le 6,\ x_1, x_2 \ge 0.$$

Draw the feasible region, identify all vertices, evaluate the objective at each vertex, and state the optimal solution.

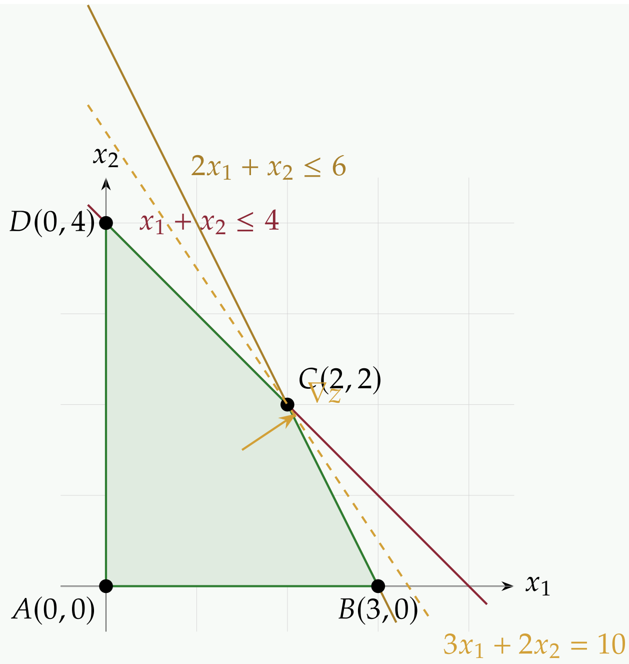
Solution. The feasible region is bounded by the constraints $x_1 + x_2 \le 4$, $2x_1 + x_2 \le 6$, and $x_1, x_2 \ge 0$. We find the vertices by intersecting the boundary lines:

1. Intersection of $x_1 = 0$ and $x_2 = 0$: $A = (0, 0)$.
2. Intersection of $2x_1 + x_2 = 6$ and $x_2 = 0$: $B = (3, 0)$.
3. Intersection of $x_1 + x_2 = 4$ and $2x_1 + x_2 = 6$: Subtracting the first equation from the second gives $x_1 = 2$, which implies $x_2 = 2$. So $C = (2, 2)$.
4. Intersection of $x_1 + x_2 = 4$ and $x_1 = 0$: $D = (0, 4)$.

We evaluate the objective function $z(x_1, x_2) = 3x_1 + 2x_2$ at each vertex:

- $z(A) = 3(0) + 2(0) = 0$.
- $z(B) = 3(3) + 2(0) = 9$.
- $z(C) = 3(2) + 2(2) = 10$.
- $z(D) = 3(0) + 2(4) = 8$.

The optimal solution is $x^* = (2, 2)$ with optimal value $z^* = 10$.



Exercise 2. Solve the following LP graphically:

$$\min \; x_1 - 2x_2 \quad \text{s.t.} \quad x_1 + 2x_2 \leq 8, \; x_1 - x_2 \leq 3, \; x_1, x_2 \geq 0.$$

Identify the optimal vertex and the optimal objective value.

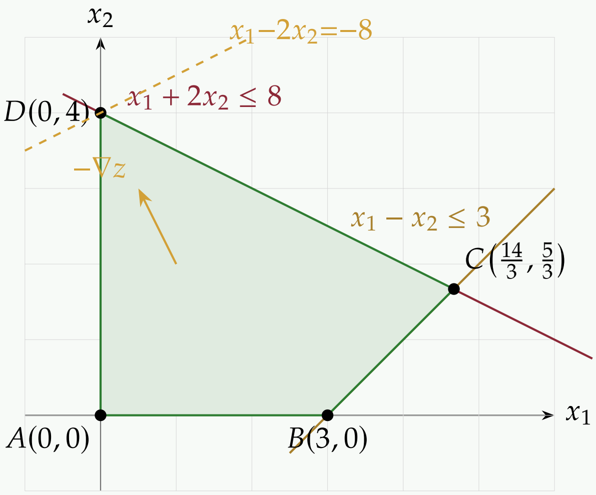
Solution. The feasible region is bounded by $x_1 + 2x_2 \leq 8$, $x_1 - x_2 \leq 3$, and $x_1, x_2 \geq 0$. Let's find the vertices:

1. Intersection of $x_1 = 0$ and $x_2 = 0$: $A = (0,0)$.
2. Intersection of $x_1 - x_2 = 3$ and $x_2 = 0$: $B = (3,0)$.
3. Intersection of $x_1 - x_2 = 3$ and $x_1 + 2x_2 = 8$: Subtracting the first equation from the second yields $3x_2 = 5 \implies x_2 = 5/3$. Then $x_1 = 14/3$. So $C = (14/3, 5/3) \approx (4.67, 1.67)$.
4. Intersection of $x_1 + 2x_2 = 8$ and $x_1 = 0$: $D = (0,4)$.

We evaluate the objective function $z(x_1, x_2) = x_1 - 2x_2$ at each vertex:

- $z(A) = 0 - 2(0) = 0$.
- $z(B) = 3 - 2(0) = 3$.
- $z(C) = 14/3 - 10/3 = 4/3 \approx 1.33$.
- $z(D) = 0 - 2(4) = -8$.

Since we are minimising, the optimal solution is $x^* = (0,4)$ with optimal value $z^* = -8$.



Exercise 3. Consider the LP

$$\max\ 2x_1 + 2x_2 \quad \text{s.t.} \quad x_1 + x_2 \leq 5,\ x_1 \leq 3,\ x_2 \leq 3,\ x_1, x_2 \geq 0.$$

Solve it graphically. How many optimal solutions does it have? Describe the set of all optimal solutions.

Solution. The feasible region is bounded by $x_1 + x_2 \leq 5$, $x_1 \leq 3$, $x_2 \leq 3$, and $x_1, x_2 \geq 0$. The vertices of the feasible region are: $A = (0, 0)$, $B = (3, 0)$, $C = (3, 2)$, $D = (2, 3)$, $E = (0, 3)$.

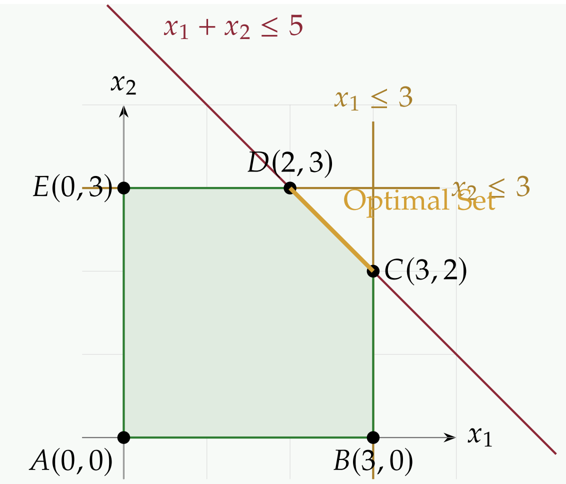
We evaluate the objective $z(x_1, x_2) = 2x_1 + 2x_2$:

- $z(A) = 0$.
- $z(B) = 6$.
- $z(C) = 2(3) + 2(2) = 10$.
- $z(D) = 2(2) + 2(3) = 10$.
- $z(E) = 6$.

Since the maximum value of 10 is achieved at two distinct vertices $C(3, 2)$ and $D(2, 3)$, any point on the line segment connecting them is also optimal. The set of all optimal solutions is the segment:

$$X^* = \{\lambda(3,2)+(1-\lambda)(2,3) \mid 0 \leq \lambda \leq 1\} = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1+x_2 = 5,\ 2 \leq x_1 \leq 3\}.$$

Thus, there are infinitely many optimal solutions.



Exercise 4. Use the graphical method to determine whether the LP

$$\max\ x_1 + x_2 \quad \text{s.t.} \quad -x_1 + x_2 \leq 1,\ x_1 - 2x_2 \leq 2,\ x_1, x_2 \geq 0$$

is feasible, infeasible, or unbounded. Justify your answer with a sketch.

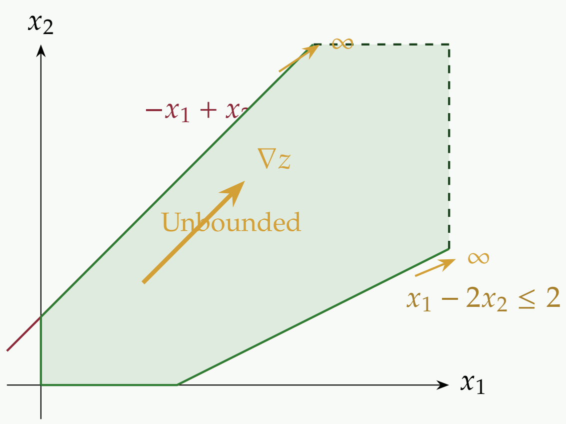
Solution. The feasible region is defined by $-x_1 + x_2 \leq 1$, $x_1 - 2x_2 \leq 2$, and $x_1, x_2 \geq 0$. The boundary lines are:

- $-x_1 + x_2 = 1 \implies x_2 = x_1 + 1$ (passes through $(0,1)$ and $(1,2)$).
- $x_1 - 2x_2 = 2 \implies x_1 = 2x_2 + 2$ (passes through $(2,0)$ and $(4,1)$).

The feasible region is non-empty (for instance, $(0,0)$ is feasible since $0 \leq 1$ and $0 \leq 2$). However, the region is unbounded in the direction of the vector $v = (2,1)$. Indeed, for any $x_2 \geq 0$, if we set $x_1 = 2x_2 + 2$, we get:

- $-x_1 + x_2 = -(2x_2 + 2) + x_2 = -x_2 - 2 \leq 1$ (always satisfied).
- $x_1 - 2x_2 = 2 \leq 2$ (satisfied).

Since x_2 can be made arbitrarily large, x_1 also grows without bound, and the objective function $z(x_1, x_2) = x_1 + x_2 = 3x_2 + 2$ approaches $+\infty$. Thus, the LP is **unbounded**.



Exercise 5. Solve graphically:

$$\min\ 4x_1 + 3x_2 \quad \text{s.t.} \quad 2x_1 + x_2 \geq 6,\ x_1 + 2x_2 \geq 6,\ x_1, x_2 \geq 0.$$

Identify the optimal vertex and confirm it is feasible.

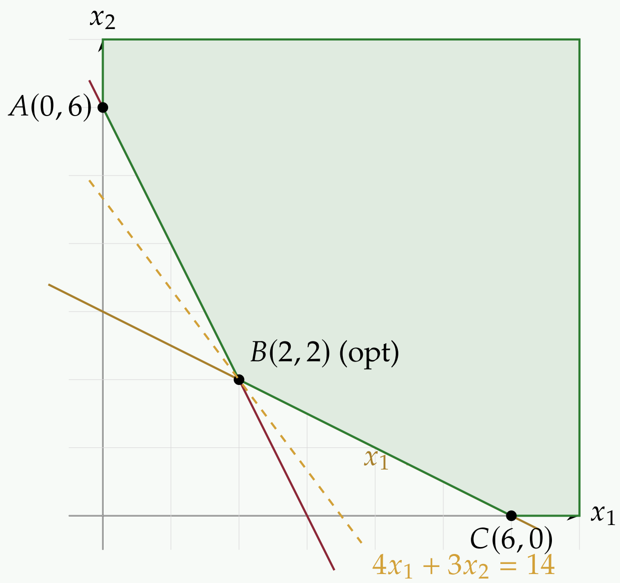
Solution. The constraints are $2x_1 + x_2 \geq 6$, $x_1 + 2x_2 \geq 6$, and $x_1, x_2 \geq 0$. The feasible region is the area above both constraint lines in the first quadrant. Let's find the boundary vertices:

1. Intersection of $2x_1 + x_2 = 6$ and $x_1 = 0$: $A = (0, 6)$.
2. Intersection of $2x_1 + x_2 = 6$ and $x_1 + 2x_2 = 6$: Subtracting gives $x_1 = x_2 \implies 3x_1 = 6 \implies x_1 = 2, x_2 = 2$. So $B = (2, 2)$.
3. Intersection of $x_1 + 2x_2 = 6$ and $x_2 = 0$: $C = (6, 0)$.

Evaluating the objective $z(x_1, x_2) = 4x_1 + 3x_2$:

- $z(A) = 4(0) + 3(6) = 18$.
- $z(B) = 4(2) + 3(2) = 14$.
- $z(C) = 4(6) + 3(0) = 24$.

The minimum is at $B = (2, 2)$ with optimal value $z^* = 14$. Feasibility check: $2(2) + 2 = 6 \geq 6$ and $2 + 2(2) = 6 \geq 6$ and $2, 2 \geq 0$. Thus, B is feasible.



Exercise 6. For the LP

$$\max\ 5x_1 + 4x_2 \quad \text{s.t.} \quad 6x_1 + 4x_2 \leq 24,\ x_1 + 2x_2 \leq 6,\ x_1, x_2 \geq 0,$$

sketch the feasible region, find all corner points algebraically (by solving pairs of binding constraints), and determine the optimal solution.

Solution. The feasible region is bounded by $6x_1 + 4x_2 \leq 24$, $x_1 + 2x_2 \leq 6$, and $x_1, x_2 \geq 0$. The corner points are found by setting pairs of boundary lines to equality:

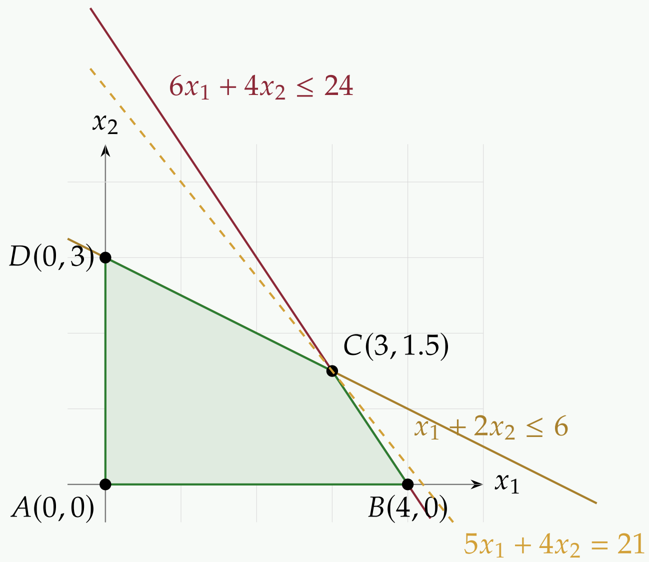
- $x_1 = 0$ and $x_2 = 0$: $A = (0, 0)$.

- $6x_1 + 4x_2 = 24$ and $x_2 = 0 \implies x_1 = 4$. So $B = (4, 0)$.
- $6x_1 + 4x_2 = 24$ and $x_1 + 2x_2 = 6$: Multiplying the second equation by 2 yields $2x_1 + 4x_2 = 12$. Subtracting this from the first yields $4x_1 = 12 \implies x_1 = 3$. Then $3 + 2x_2 = 6 \implies x_2 = 1.5$. So $C = (3, 1.5)$.
- $x_1 + 2x_2 = 6$ and $x_1 = 0 \implies x_2 = 3$. So $D = (0, 3)$.

We evaluate the objective $z(x_1, x_2) = 5x_1 + 4x_2$:

- $z(A) = 0$.
- $z(B) = 5(4) + 4(0) = 20$.
- $z(C) = 5(3) + 4(1.5) = 15 + 6 = 21$.
- $z(D) = 5(0) + 4(3) = 12$.

The optimal solution is $x^* = (3, 1.5)$ with optimal value $z^* = 21$.



Exercise 7. Convert the following LP to standard form (minimisation, equality constraints, all variables ≥ 0):

$$\max\ 3x_1 - x_2 + 2x_3 \quad \text{s.t.} \quad x_1 + x_2 \leq 5,\ 2x_1 - x_3 \geq 1,\ x_1 + x_2 + x_3 = 4,\ x_1, x_2, x_3 \geq 0.$$

Name every slack or surplus variable you introduce.

Solution. To convert to standard form:

1. Convert the maximization objective to minimization by multiplying by -1:
$$\text{minimize } -3x_1 + x_2 - 2x_3$$
2. Convert $x_1 + x_2 \leq 5$ to an equality by introducing a non-negative slack variable $s_1 \geq 0$:
$$x_1 + x_2 + s_1 = 5$$
3. Convert $2x_1 - x_3 \geq 1$ to an equality by introducing a non-negative

surplus variable $e_1 \geq 0$:

$$2x_1 - x_3 - e_1 = 1$$

4. The third constraint $x_1 + x_2 + x_3 = 4$ is already an equality, so it remains unchanged.

5. All variables x_1, x_2, x_3, s_1, e_1 are non-negative.

The standard form model is:

$$\begin{aligned} \text{minimize} \quad & -3x_1 + x_2 - 2x_3 \\ \text{subject to} \quad & x_1 + x_2 + s_1 = 5 \\ & 2x_1 - x_3 - e_1 = 1 \\ & x_1 + x_2 + x_3 = 4 \\ & x_1, x_2, x_3, s_1, e_1 \geq 0. \end{aligned}$$

Here, s_1 is the slack variable for the first constraint, and e_1 is the surplus variable for the second constraint.

Exercise 8. Convert to standard form:

$$\min \; x_1 + 2x_2 \quad \text{s.t.} \quad x_1 - x_2 \geq 3, \; x_1 + 3x_2 \leq 9, \; x_1 \geq 0, \; x_2 \text{ unrestricted.}$$

Show the substitution used for the unrestricted variable.

Solution. The variable x_2 is unrestricted (free), so we substitute $x_2 = x_2^+ - x_2^-$ with $x_2^+, x_2^- \geq 0$. Now we convert the inequalities:

1. Convert $x_1 - x_2 \geq 3 \implies x_1 - (x_2^+ - x_2^-) \geq 3$ to an equality by subtracting a surplus variable $e_1 \geq 0$:

$$x_1 - x_2^+ + x_2^- - e_1 = 3$$

2. Convert $x_1 + 3x_2 \leq 9 \implies x_1 + 3(x_2^+ - x_2^-) \leq 9$ to an equality by adding a slack variable $s_1 \geq 0$:

$$x_1 + 3x_2^+ - 3x_2^- + s_1 = 9$$

The standard form model is:

$$\begin{aligned} \text{minimize} \quad & x_1 + 2x_2^+ - 2x_2^- \\ \text{subject to} \quad & x_1 - x_2^+ + x_2^- - e_1 = 3 \\ & x_1 + 3x_2^+ - 3x_2^- + s_1 = 9 \\ & x_1, x_2^+, x_2^-, e_1, s_1 \geq 0. \end{aligned}$$

Exercise 9. Convert to standard form:

$$\max \; -2x_1 + x_2 + 3x_3 \quad \text{s.t.} \quad x_1 + x_2 + x_3 \leq 10, \; x_1 - x_2 + 2x_3 \geq 2, \; x_1 \geq 0, \; x_2 \geq 0, \; x_3 \leq 0.$$

Introduce $x_3' = -x_3 \geq 0$ and write the resulting standard-form problem.

Solution. First, since $x_3 \le 0$, we substitute $x_3 = -x_3'$ where $x_3' \ge 0$. Next, we convert the maximization to minimization by multiplying the objective by -1:

$$\text{minimize } 2x_1 - x_2 - 3x_3 = 2x_1 - x_2 + 3x_3'$$

Now convert the inequalities:

1. For $x_1 + x_2 + x_3 \le 10 \implies x_1 + x_2 - x_3' \le 10$, add a slack variable $s_1 \ge 0$:
$$x_1 + x_2 - x_3' + s_1 = 10$$
2. For $x_1 - x_2 + 2x_3 \ge 2 \implies x_1 - x_2 - 2x_3' \ge 2$, subtract a surplus variable $e_1 \ge 0$:
$$x_1 - x_2 - 2x_3' - e_1 = 2$$

The standard form model is:

$$\begin{aligned} \text{minimize} \quad & 2x_1 - x_2 + 3x_3' \\ \text{subject to} \quad & x_1 + x_2 - x_3' + s_1 = 10 \\ & x_1 - x_2 - 2x_3' - e_1 = 2 \\ & x_1, x_2, x_3', s_1, e_1 \ge 0. \end{aligned}$$

Exercise 10. A linear programme has variables $x_1 \ge 0$, x_2 unrestricted, and $x_3 \le 0$. Write the substitutions needed to express all three variables as non-negative variables, and state how many new variables are introduced in the worst case.

Solution. The substitutions required are:

- x_1 is already non-negative ($x_1 \ge 0$), so no substitution is needed.
- x_2 is unrestricted, so we substitute $x_2 = x_2^+ - x_2^-$ with $x_2^+, x_2^- \ge 0$.
- x_3 is non-positive ($x_3 \le 0$), so we substitute $x_3 = -x_3'$ with $x_3' \ge 0$.

Thus, we replace the original variables x_2, x_3 with three new non-negative variables x_2^+, x_2^-, x_3'. The total number of variables increases by 1 (we started with 3 and end with 4). In general, for each unrestricted variable we introduce 2 non-negative variables (an increase of 1 variable), and for each bounded variable we use a direct transformation (no increase in variable count). Thus, in the worst case, if all n variables are unrestricted, we introduce n new variables, doubling the variable count.

Exercise 11. Convert the following LP to canonical form (all inequality constraints of the same sense, right-hand side ≥ 0, no equality constraints):

$$\min \ x_1 + x_2 \quad \text{s.t.} \quad 2x_1 + x_2 = 4, \ x_1 - x_2 \le 1, \ x_1, x_2 \ge 0.$$

Solution. To convert to canonical form (minimisation, all constraints of the sense $\ge$, variables non-negative):

1. Split the equality constraint $2x_1 + x_2 = 4$ into two inequalities:

$$2x_1 + x_2 \geq 4 \quad \text{and} \quad 2x_1 + x_2 \leq 4 \implies -2x_1 - x_2 \geq -4$$

2. Convert the inequality $x_1 - x_2 \leq 1$ to $\geq$ form by multiplying by -1:

$$-x_1 + x_2 \geq -1$$

The canonical form model is:

$$\begin{aligned} \text{minimize} \quad & x_1 + x_2 \\ \text{subject to} \quad & 2x_1 + x_2 \geq 4 \\ & -2x_1 - x_2 \geq -4 \\ & -x_1 + x_2 \geq -1 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Exercise 12. Let $P = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 + x_2 \leq 3,\ x_1 \geq 0,\ x_2 \geq 0\}$.

(a) List all vertices of P.

(b) For each vertex, state which constraints are active (tight).

(c) State the dimension of P.

Solution.

(a) The polyhedron P is a right triangle in the first quadrant. Its vertices are:

$$A = (0, 0), \quad B = (3, 0), \quad C = (0, 3).$$

(b) The constraints defining P are:

$$\begin{aligned} (1) \quad & x_1 + x_2 \leq 3 \\ (2) \quad & -x_1 \leq 0 \\ (3) \quad & -x_2 \leq 0 \end{aligned}$$

For each vertex, the active constraints (where the inequality holds as an equality) are:

- For $A(0, 0)$: Constraints (2) and (3) are active ($x_1 = 0$, $x_2 = 0$).
- For $B(3, 0)$: Constraints (1) and (3) are active ($x_1 + x_2 = 3$, $x_2 = 0$).
- For $C(0, 3)$: Constraints (1) and (2) are active ($x_1 + x_2 = 3$, $x_1 = 0$).

(c) The polyhedron P has a non-empty interior in $\mathbb{R}^2$ (for example, the point $(1, 1)$ satisfies all constraints strictly: $1 + 1 = 2 < 3, 1 > 0, 1 > 0$). Thus, the dimension of P is 2.

Exercise 13. Consider the polyhedron

$$P = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 \geq 0,\ x_2 \geq 0,\ x_1 + 2x_2 \leq 6,\ 2x_1 + x_2 \leq 6\}.$$

(a) Find all vertices of P by solving pairs of binding constraints.

(b) List all edges (1-dimensional faces) and the inequalities that define them.

Solution.

(a) The constraints are:

$$\begin{aligned}
&(1) \quad -x_1 \le 0 \\
&(2) \quad -x_2 \le 0 \\
&(3) \quad x_1 + 2x_2 \le 6 \\
&(4) \quad 2x_1 + x_2 \le 6
\end{aligned}$$

We check all pairs of binding constraints:

- (1) and (2): $x_1 = 0, x_2 = 0$. Feasible because $0 + 2(0) \le 6$ and $2(0) + 0 \le 6$. Vertex: $A = (0, 0)$.
- (2) and (4): $x_2 = 0, 2x_1 + x_2 = 6 \implies x_1 = 3$. Feasible because $3 + 2(0) \le 6$. Vertex: $B = (3, 0)$.
- (3) and (4): $x_1 + 2x_2 = 6$ and $2x_1 + x_2 = 6$. Solving this system yields $x_1 = 2, x_2 = 2$. Feasible because $2 \ge 0, 2 \ge 0$. Vertex: $C = (2, 2)$.
- (1) and (3): $x_1 = 0, x_1 + 2x_2 = 6 \implies x_2 = 3$. Feasible because $2(0) + 3 \le 6$. Vertex: $D = (0, 3)$.
- (1) and (4): $x_1 = 0, 2x_1 + x_2 = 6 \implies x_2 = 6$. Infeasible because $x_1 + 2x_2 = 0 + 12 = 12 > 6$.
- (2) and (3): $x_2 = 0, x_1 + 2x_2 = 6 \implies x_1 = 6$. Infeasible because $2x_1 + x_2 = 12 + 0 = 12 > 6$.

Thus, the vertices are $A(0, 0)$, $B(3, 0)$, $C(2, 2)$, and $D(0, 3)$.

(b) The edges (1-dimensional faces) connect adjacent vertices and are defined by setting one of the defining inequalities to equality over the feasible domain:

- Edge AB: Defined by constraint $x_2 = 0$ for $0 \le x_1 \le 3$.
- Edge BC: Defined by constraint $2x_1 + x_2 = 6$ for $2 \le x_1 \le 3$.
- Edge CD: Defined by constraint $x_1 + 2x_2 = 6$ for $0 \le x_1 \le 2$.
- Edge DA: Defined by constraint $x_1 = 0$ for $0 \le x_2 \le 3$.

Exercise 14. Prove that the feasible region of any LP with only inequality constraints (no equality constraints) and non-negative variables is a convex set.

Solution. The feasible region of such an LP can be written as:

$$P = \{x \in \mathbb{R}^n \mid Ax \le b,\ x \ge 0\}.$$

To prove P is convex, we must show that for any two points $y, z \in P$ and any $\lambda \in [0, 1]$, the convex combination $w = \lambda y + (1 - \lambda)z$ also belongs to P.

Let $y, z \in P$ and $\lambda \in [0, 1]$.

1. First, we check the non-negativity of w: Since $y \ge 0$ and $z \ge 0$, and

since $\lambda \geq 0$ and $1 - \lambda \geq 0$, we have:

$$w = \lambda y + (1 - \lambda)z \geq 0.$$

2. Next, we verify that w satisfies the inequality constraints $Aw \leq b$: Using the linearity of matrix multiplication:

$$Aw = A(\lambda y + (1 - \lambda)z) = \lambda Ay + (1 - \lambda)Az.$$

Since $y \in P \implies Ay \leq b$ and $z \in P \implies Az \leq b$, and since $\lambda \geq 0, 1 - \lambda \geq 0$:

$$Aw \leq \lambda b + (1 - \lambda)b = (\lambda + 1 - \lambda)b = b.$$

Since $w \geq 0$ and $Aw \leq b$, we conclude that $w \in P$. Hence, P is a convex set.

Exercise 15. A point $\bar{x}$ in a polyhedron P is a *vertex* (extreme point) if and only if it cannot be written as $\bar{x} = \lambda y + (1 - \lambda)z$ with $y, z \in P$, $y \neq z$, $0 < \lambda < 1$. Using this definition, show that $(0, 0)$ is a vertex of the polyhedron $P = \{x \in \mathbb{R}^2 \mid x_1 + x_2 \leq 1,\ x_1 \geq 0,\ x_2 \geq 0\}$.

Solution. Suppose $\bar{x} = (0, 0)$ can be written as a convex combination of two points $y, z \in P$ with $0 < \lambda < 1$. That is:

$$(0, 0) = \lambda y + (1 - \lambda)z, \quad \text{where } y = (y_1, y_2) \in P,\ z = (z_1, z_2) \in P.$$

This vector equation gives two scalar equations:

$$0 = \lambda y_1 + (1 - \lambda)z_1$$
$$0 = \lambda y_2 + (1 - \lambda)z_2$$

Since $y, z \in P$, we have $y_1, y_2, z_1, z_2 \geq 0$. Also, we have $\lambda > 0$ and $1 - \lambda > 0$. Since the sum of two non-negative terms λy_1 and $(1 - \lambda)z_1$ is 0, each term must be individually 0:

$$\lambda y_1 = 0 \implies y_1 = 0 \quad (\text{since } \lambda > 0)$$

$$(1 - \lambda)z_1 = 0 \implies z_1 = 0 \quad (\text{since } 1 - \lambda > 0)$$

Similarly, for the second equation:

$$\lambda y_2 = 0 \implies y_2 = 0$$

$$(1 - \lambda)z_2 = 0 \implies z_2 = 0$$

This implies $y = (0, 0)$ and $z = (0, 0)$, which means $y = z$. This contradicts the requirement that $y \neq z$. Thus, $(0, 0)$ cannot be written as a convex combination of two distinct points in P, proving it is a vertex.

Exercise 16. Let $P \subset \mathbb{R}^2$ be the unit square $[0, 1]^2$.

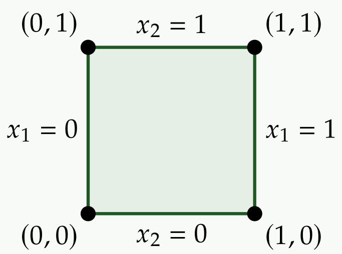
(a) List all faces of P (vertices, edges, the whole square, and the empty face), and for each face give the supporting hyperplane that defines it.

(b) How many facets does P have?

Solution. Write $P = \{x \in \mathbb{R}^2 : 0 \le x_1 \le 1,\ 0 \le x_2 \le 1\}$. Its proper non-empty faces are:

Face	Supporting hyperplane
$(0,0)$	$x_1 + x_2 = 0$
$(1,0)$	$x_1 - x_2 = 1$
$(0,1)$	$-x_1 + x_2 = 1$
$(1,1)$	$x_1 + x_2 = 2$
left and right edges	$x_1 = 0,\ x_1 = 1$
bottom and top edges	$x_2 = 0,\ x_2 = 1$

The whole square P and $\varnothing$ are the two improper faces. Under the standard definition, a supporting hyperplane has a non-zero normal and must meet P, so these two improper faces have no supporting hyperplane. If degenerate exposing equations are allowed, $0^\top x = 0$ exposes all of P, while the valid but non-tight inequality $x_1 \le 2$ has equality set disjoint from P and represents the empty face. In dimension two, facets are edges, so P has **four facets**.



Exercise 17. Give an example of a polyhedron in $\mathbb{R}^2$ that has infinitely many feasible points but *no* vertices. Describe it by its half-space representation and explain why no vertex exists.

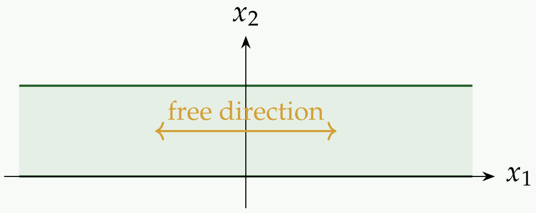
Solution. Take the horizontal strip

$$P = \{(x_1, x_2) \in \mathbb{R}^2 : 0 \le x_2 \le 1\} = \{x : -x_2 \le 0,\ x_2 \le 1\}.$$

For every $x \in P$ and $\varepsilon > 0$,

$$x = \tfrac{1}{2}(x - \varepsilon e_1) + \tfrac{1}{2}(x + \varepsilon e_1),$$

where both points on the right are distinct and remain in P. Thus no point is extreme and P has no vertices.



Exercise 18. Consider the half-spaces $H_1 = \{x : a^\top x \le b\}$ and $H_2 = \{x : c^\top x \le d\}$ in $\mathbb{R}^n$. Prove that $H_1 \cap H_2$ is convex.

Solution. Let $x, y \in H_1 \cap H_2$ and $\lambda \in [0,1]$. Then

$$a^\top(\lambda x + (1-\lambda)y) \le \lambda b + (1-\lambda)b = b,$$
$$c^\top(\lambda x + (1-\lambda)y) \le \lambda d + (1-\lambda)d = d.$$

The convex combination satisfies both inequalities and therefore belongs to $H_1 \cap H_2$.

Exercise 19. Let $P = \{x \in \mathbb{R}^3 : x_1 + x_2 + x_3 = 1,\ x_1, x_2, x_3 \ge 0\}$.

(a) State the dimension of P.

(b) Find all vertices of P.

(c) Describe the edges of P.

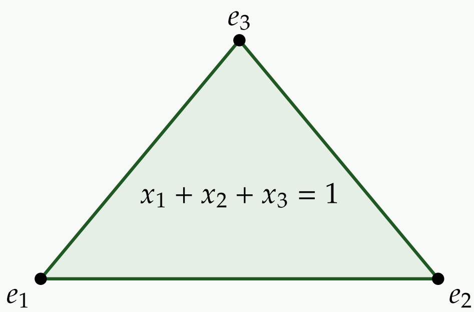
Solution.

(a) The equality confines P to a plane, while $(1/3, 1/3, 1/3)$ is relatively interior. Hence $\dim P = 2$.

(b) Its vertices are $e_1 = (1,0,0)$, $e_2 = (0,1,0)$, and $e_3 = (0,0,1)$.

(c) Its edges are the three segments

$$E_{12} = \{(\lambda, 1-\lambda, 0) : 0 \le \lambda \le 1\},$$
$$E_{13} = \{(\lambda, 0, 1-\lambda) : 0 \le \lambda \le 1\},$$
$$E_{23} = \{(0, \lambda, 1-\lambda) : 0 \le \lambda \le 1\}.$$



Exercise 20. Prove that the intersection of any finite collection of convex sets is convex.

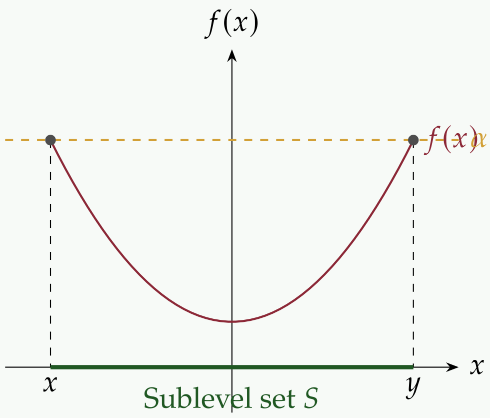
Solution. Let $C = \bigcap_{i=1}^m C_i$, with every C_i convex. If $x, y \in C$, then $x, y \in C_i$ for every i. Hence $\lambda x + (1-\lambda)y \in C_i$ for every i and every $\lambda \in [0,1]$. The convex combination therefore belongs to C.

Exercise 21. Let $f : \mathbb{R}^n \to \mathbb{R}$ be convex and $\alpha \in \mathbb{R}$. Show that the sublevel set $S = \{x : f(x) \le \alpha\}$ is convex.

Solution. For $x, y \in S$ and $\lambda \in [0, 1]$, convexity gives

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \leq \lambda\alpha + (1 - \lambda)\alpha = \alpha.$$

Thus the whole segment between x and y lies in S.

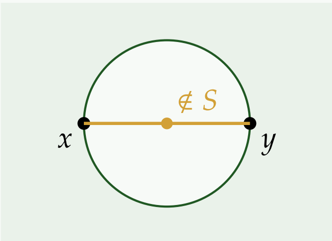


Exercise 22. Give an example of a set that is *not* convex and explain which convexity condition it violates. Use a set in $\mathbb{R}^2$ defined by a simple inequality.

Solution. Let

$$S = \{(x_1, x_2) \in \mathbb{R}^2 : x_1^2 + x_2^2 \geq 1\}.$$

Both $(-1, 0)$ and $(1, 0)$ belong to S, but their midpoint $(0, 0)$ does not. Hence S fails the line-segment condition.



Exercise 23. Let $S = \{(0, 0), (2, 0), (1, 2)\} \subset \mathbb{R}^2$.

(a) Describe $\text{conv}(S)$ by half-space inequalities.

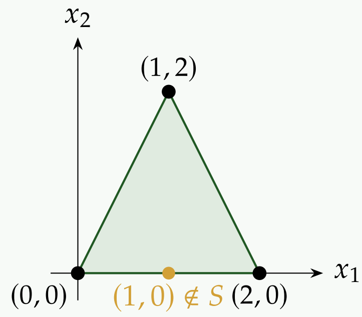
(b) Is S itself convex? Justify.

Solution.

(a) The convex hull is the triangle

$$\text{conv}(S) = \{x : x_2 \geq 0,\ x_2 \leq 2x_1,\ x_2 \leq -2x_1 + 4\}.$$

(b) No. The midpoint $(1, 0)$ of $(0, 0)$ and $(2, 0)$ does not belong to the three-point set S.



Exercise 24. Prove that every polytope is equal to the convex hull of its vertices. You may appeal to Minkowski–Weyl.

Solution. Minkowski–Weyl gives a finite representation

$$P = \operatorname{conv}(V) + \operatorname{cone}(R)$$

for every polyhedron. If P is bounded, it has no non-zero recession direction, so $\operatorname{cone}(R) = \{0\}$ and $P = \operatorname{conv}(V)$. Removing redundant points from the finite set V leaves exactly the vertices of P. This is the bounded polyhedron direction of Minkowski–Weyl.

Exercise 25. Express every point of $P = \operatorname{conv}\{(0,0),(4,0),(0,4),(2,2)\}$ as a convex combination, using coefficients $\lambda_1, \ldots, \lambda_4 \geq 0$ with $\sum_i \lambda_i = 1$.

Solution. The point $(2,2) = \frac{1}{2}(4,0) + \frac{1}{2}(0,4)$ is a redundant generator, not a vertex. Thus

$$P = \{x : x_1 \geq 0,\ x_2 \geq 0,\ x_1 + x_2 \leq 4\}.$$

For any $(x_1, x_2) \in P$, choose

$$\lambda_1 = 1 - \frac{x_1 + x_2}{4}, \qquad \lambda_2 = \frac{x_1}{4}, \qquad \lambda_3 = \frac{x_2}{4}, \qquad \lambda_4 = 0.$$

They are non-negative, sum to one, and satisfy

$$(x_1, x_2) = \lambda_1(0,0) + \lambda_2(4,0) + \lambda_3(0,4) + \lambda_4(2,2).$$

Exercise 26. For

$$P = \{(x_1, x_2) : x_1 + x_2 \geq 1,\ x_1 \geq 0,\ x_2 \geq 0\},$$

identify the lineality/recession directions and decompose P into the sum of a polytope and a cone.

Solution. The recession cone is

$$\operatorname{rec}(P) = \mathbb{R}^2_{\geq 0},$$

because increasing either coordinate preserves every constraint. The lineality space is $\operatorname{rec}(P) \cap (-\operatorname{rec}(P)) = \{0\}$.

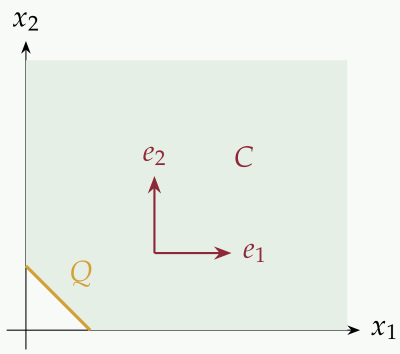
Let

$$Q = \text{conv}\{(1,0),(0,1)\}, \qquad C = \mathbb{R}^2_{\geq 0}.$$

Then

$$\boxed{P = Q + C}.$$

Indeed, every $q + c$ has non-negative coordinates and sum at least one. Conversely, from any $x \in P$ one can decrease non-negative coordinates until their sum is exactly one, obtaining $x = q + c$ with $q \in Q, c \in C$.



Exercise 27. Show that $v = (1,0,0)$ is an extreme point of $\Delta = \{x \in \mathbb{R}^3 : x_1 + x_2 + x_3 = 1,\ x_i \geq 0\}$ using the active-constraint characterisation.

Solution. At v the equality and the constraints $x_2 = 0, x_3 = 0$ are active. Their normals form

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

whose determinant is 1. The active normals therefore have rank 3, equal to the ambient dimension, and their equalities have the unique solution $(1,0,0)$. Hence v is a vertex.

Exercise 28. For each statement, state whether it is **true** or **false**.

(a) Every LP with a bounded feasible region has a finite optimum.

(b) If a minimisation LP is feasible and its objective is bounded below, then an optimum exists.

(c) The optimal value of an LP is always attained at a vertex.

(d) An LP can have exactly two optimal solutions.

Solution.

(a) **False as stated.** The empty set is bounded, but an infeasible LP has no optimum. With non-emptiness added, it is true.

(b) **True.** A non-empty polyhedron is closed, and a linear objective bounded below on it attains its infimum.

(c) **False.** Minimise the constant objective 0 on the line $\{(x_1, x_2) : x_2 = 0\}$. Every point is optimal and the line has no vertices.

(d) **False.** If two distinct points are optimal, their entire segment is feasible and optimal by convexity and linearity.

Exercise 29. State whether each statement is **true** or **false**.

(a) The feasible region of every LP is convex.

(b) Adding a constraint can only decrease or maintain the optimal value for maximisation.

(c) The set of optimal solutions of an LP is convex.

(d) If an LP in $\mathbb{R}^2$ has exactly one optimum, it is a vertex.

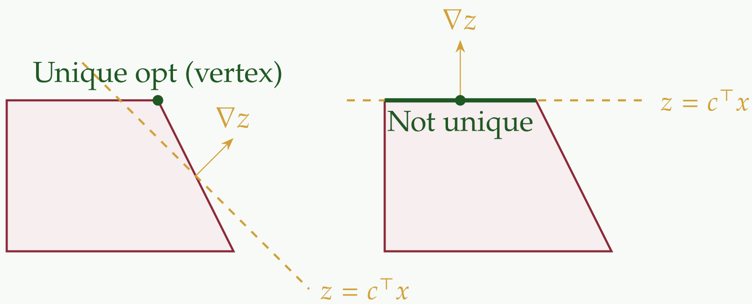
Solution.

(a) **True**: it is an intersection of half-spaces and hyperplanes.

(b) **True**: the new feasible region is a subset of the old one.

(c) **True**: a convex combination of two optima has the same objective value.

(d) **True**: if the optimum were not extreme, it would be a strict convex combination of two distinct feasible points; linearity would make both endpoints optimal as well.



Exercise 30. State whether each statement is **true** or **false**.

(a) A polyhedron defined by m inequalities in $\mathbb{R}^n$ has at most $\binom{m}{n}$ vertices.

(b) Every bounded polyhedron is a polytope.

(c) The empty set is a valid polyhedron.

(d) One equality $a^\top x = b$ in $\mathbb{R}^n$ always defines a set of dimension $n - 1$.

Solution.

(a) **True.** At each vertex one can select n independent active inequalities, and any fixed selection has at most one common solution. Degeneracy can only cause overcounting.

(b) **True**, by the definition/equivalent characterisation of a polytope.

(c) **True**; for example $\{x : x \leq 0,\ x \geq 1\} = \varnothing$.

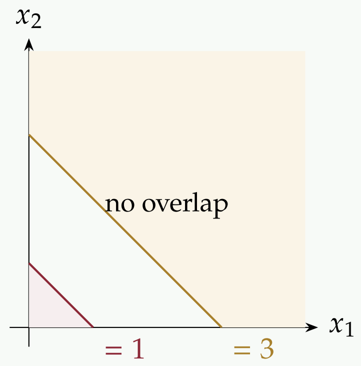
(d) **False as stated.** It is true when $a \neq 0$. If $a = 0, b = 0$, the set is $\mathbb{R}^n$; if $a = 0, b \neq 0$, it is empty.

Exercise 31. Write a specific infeasible LP in two variables. Sketch its empty feasible region and explain why no feasible point exists.

Solution. For example,

$$\max x_1 + x_2 \quad \text{s.t.} \quad x_1 + x_2 \le 1, \quad x_1 + x_2 \ge 3, \quad x_1, x_2 \ge 0.$$

The same quantity cannot be at most 1 and at least 3, so the feasible set is empty.

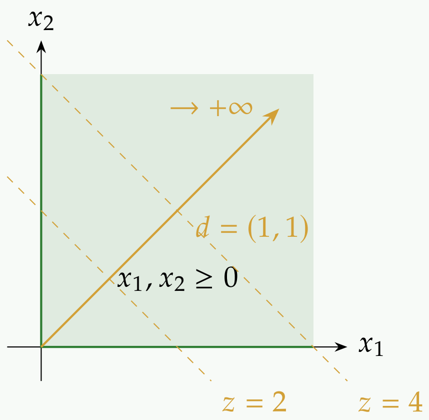


Exercise 32. Write a specific LP in two variables that is unbounded. Sketch the feasible region and show an improving direction.

Solution. Take

$$\max x_1 + x_2 \quad \text{s.t.} \quad x_1, x_2 \ge 0.$$

Along $x(\lambda) = (\lambda, \lambda)$, $\lambda \ge 0$, the objective is $2\lambda \to +\infty$.



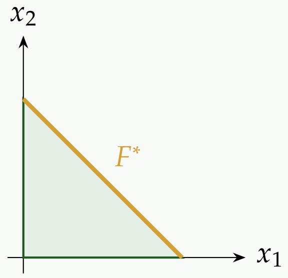
Exercise 33. Write an LP with infinitely many optimal solutions forming a line segment. Identify its optimal vertices and entire optimal face.

Solution. Consider

$$\max x_1 + x_2 \quad \text{s.t.} \quad x_1 + x_2 \le 2, \quad x_1, x_2 \ge 0.$$

The optimum is 2. The optimal vertices are $(2,0)$ and $(0,2)$, and

$$F^* = \{x : x_1 + x_2 = 2,\ x \geq 0\} = \{\lambda(2,0) + (1-\lambda)(0,2) : 0 \leq \lambda \leq 1\}.$$

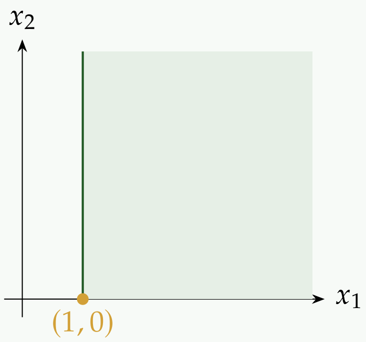


Exercise 34. Write an LP with an unbounded feasible region but a finite optimum. Explain why boundedness of the region is not necessary.

Solution. For example,

$$\min x_1 + x_2 \quad \text{s.t.} \quad x_1 \geq 1,\ x_2 \geq 0.$$

The region is unbounded, but $x_1 + x_2 \geq 1$, with equality uniquely at $(1,0)$. Thus $z^* = 1$. An unbounded feasible region causes an unbounded objective only when it has a recession direction that improves the objective; here every recession direction $d \geq 0$ has $(1,1)^\top d \geq 0$.



Exercise 35. For the polyhedron $0 \leq x_i \leq 2$, $i = 1,2,3$:

(a) What is its dimension?

(b) How many vertices does it have?

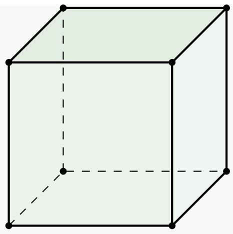
(c) How many facets does it have?

Solution. This is the cube $[0,2]^3$.

(a) It has non-empty interior, so its dimension is 3.

(b) Each coordinate independently equals 0 or 2 at a vertex: there are $2^3 = 8$ vertices.

(c) Each of the six bounds $x_i = 0$ or $x_i = 2$ exposes a facet: there are 6 facets.



Exercise 36. A point $x^* \in \mathbb{R}^3$ satisfies four out of six inequality constraints with equality. Under what conditions on the active-constraint matrix is x^* guaranteed to be a vertex?

Solution. Let A_I contain the rows (normal vectors) of the four constraints active at x^*. The point is guaranteed to be a vertex when

$$\operatorname{rank}(A_I) = 3.$$

Equivalently, among the four active constraints there must be three with linearly independent normals. Then the active equalities determine a unique point in $\mathbb{R}^3$. Merely having four active constraints is not enough: if their normals span only a line or a plane, the common active set can still contain a segment.

Exercise 37. A nutritionist must design a minimum-cost daily diet using bread, milk, and eggs. Per unit, their protein contents are $3, 8, 6$ grams; fat contents are $1, 5, 5$ grams; carbohydrate contents are $15, 12, 1$ grams; and costs are €0.10, €0.20, and €0.15. Minimum requirements are 55 g protein, 33 g fat, and 70 g carbohydrates. Formulate an LP.

Solution. Let $x_B, x_M, x_E \geq 0$ denote slices of bread, cups of milk, and eggs. The three nutrient requirements are lower bounds, so the model is

$$\begin{array}{lll} \text{minimize} & 0.10x_B + 0.20x_M + 0.15x_E & \\ \text{subject to} & 3x_B + 8x_M + 6x_E \geq 55 & \text{(protein)}, \\ & x_B + 5x_M + 5x_E \geq 33 & \text{(fat)}, \\ & 15x_B + 12x_M + x_E \geq 70 & \text{(carbohydrates)}, \\ & x_B, x_M, x_E \geq 0. & \end{array}$$

The variables are continuous because the exercise asks for an LP. If whole eggs or whole slices were required, the corresponding variables would have to be integer and the model would become an ILP or MILP.

Exercise 38. Extend the diet model by requiring that no more than three eggs are consumed per day. Is the new model still an LP?

Solution. Add the linear bound

$$x_E \leq 3.$$

Nothing else changes. The model remains an LP because both the new constraint and all previous expressions are linear. The phrase "number of eggs" might motivate $x_E \in \mathbb{Z}_{\geq 0}$ in a more realistic model, but that integrality condition is not part of the stated extension.

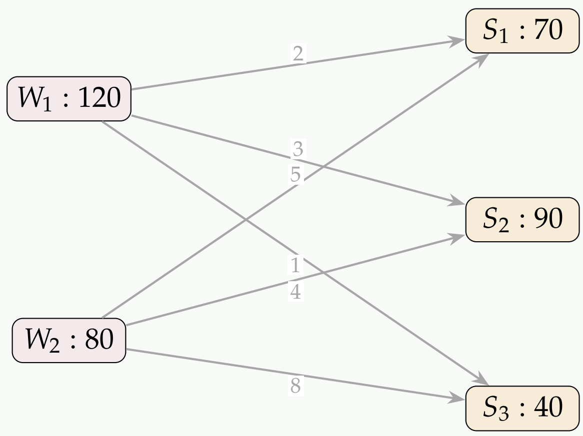
Exercise 39. A company has warehouses W_1, W_2 with supplies $120, 80$ and shops S_1, S_2, S_3 with demands $70, 90, 40$. Unit costs are

$$C = \begin{pmatrix} 2 & 3 & 1 \\ 5 & 4 & 8 \end{pmatrix}.$$

Formulate a minimum-cost transportation LP.

Solution. Let $x_{ij} \geq 0$ be the quantity shipped from warehouse W_i to shop S_j. Since total supply and demand are equal, we may use equality constraints:

$$\begin{aligned} \text{minimize} \quad & 2x_{11} + 3x_{12} + x_{13} + 5x_{21} + 4x_{22} + 8x_{23} \\ \text{subject to} \quad & x_{11} + x_{12} + x_{13} = 120 && (W_1), \\ & x_{21} + x_{22} + x_{23} = 80 && (W_2), \\ & x_{11} + x_{21} = 70 && (S_1), \\ & x_{12} + x_{22} = 90 && (S_2), \\ & x_{13} + x_{23} = 40 && (S_3), \\ & x_{ij} \geq 0 \qquad i = 1,2, \; j = 1,2,3. \end{aligned}$$



Exercise 40. For the preceding transportation problem, determine whether total supply equals total demand. If not, explain how to add a dummy node.

Solution. Total supply is

$$120 + 80 = 200,$$

and total demand is

$$70 + 90 + 40 = 200.$$

The instance is already **balanced**; no dummy node is needed.

In general, if supply exceeds demand by Δ, add a dummy shop with demand Δ and usually zero shipping cost. If demand exceeds supply by Δ, add a dummy warehouse with supply Δ. Costs on dummy arcs can instead represent disposal or shortage penalties.

Exercise 41. A paint manufacturer uses resources R_1, R_2 to make products P_1, P_2, P_3. Their resource requirements are respectively $(0.4, 0.6)$, $(0.7, 0.3)$, and $(0.5, 0.5)$ litres per litre. Supplies are 600 and 500 litres, and profits are €8, €6, €7 per litre. Formulate a profit-maximising LP.

Solution. Let $x_j \geq 0$ be litres of product P_j. Then

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 6x_2 + 7x_3 \\ \text{subject to} \quad & 0.4x_1 + 0.7x_2 + 0.5x_3 \leq 600 \quad (R_1), \\ & 0.6x_1 + 0.3x_2 + 0.5x_3 \leq 500 \quad (R_2), \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Each column of the constraint matrix describes one product's consumption of the two scarce raw materials.

Exercise 42. Crude oils A and B cost €30 and €20 per barrel. Petrol must contain at least 60% crude A, and at least 1000 barrels must be produced. Formulate a minimum-cost LP.

Solution. Let $x_A, x_B \geq 0$ be barrels of the two crude oils. The fraction condition is linear after multiplying by total volume:

$$x_A \geq 0.6(x_A + x_B) \iff 0.4x_A - 0.6x_B \geq 0.$$

Thus

$$\begin{aligned} \text{minimize} \quad & 30x_A + 20x_B \\ \text{subject to} \quad & x_A + x_B \geq 1000, \\ & 0.4x_A - 0.6x_B \geq 0, \\ & x_A, x_B \geq 0. \end{aligned}$$

Because both costs are positive, an optimum produces exactly 1000 barrels. The cheaper oil B is used as much as quality permits, giving $(x_A, x_B) = (600, 400)$ and cost €26,000.

Exercise 43. Prove or disprove: if P and Q are polyhedra in $\mathbb{R}^n$, then $P \cup Q$ is also a polyhedron.

Solution. False. In $\mathbb{R}$, let

$$P = \{x : x \leq -1\}, \qquad Q = \{x : x \geq 1\}.$$

Both are polyhedra, but $P \cup Q$ is not convex: it contains -1 and 1 but not their midpoint 0. Every polyhedron is convex, so this union cannot be a polyhedron.

Exercise 44. Prove or disprove: if P and Q are polytopes, then their Minkowski sum $P + Q = \{x + y : x \in P,\ y \in Q\}$ is a polytope.

Solution. True. Write $P = \operatorname{conv}\{p_1, \ldots, p_r\}$ and $Q = \operatorname{conv}\{q_1, \ldots, q_s\}$.

Then

$$P + Q = \operatorname{conv}\{p_i + q_j : i = 1, \ldots, r,\ j = 1, \ldots, s\}.$$

To see the non-trivial inclusion, if $x = \sum_i \alpha_i p_i$ and $y = \sum_j \beta_j q_j$, then

$$x + y = \sum_{i,j} \alpha_i \beta_j (p_i + q_j), \qquad \sum_{i,j} \alpha_i \beta_j = 1.$$

Hence $P + Q$ is the convex hull of finitely many points and is a polytope.

Exercise 45. Prove that every face of a polyhedron is itself a polyhedron.

Solution. Let $P = \{x : Ax \le b\}$ and let F be a non-empty face exposed by a valid inequality $c^\top x \le \delta$:

$$F = \{x \in P : c^\top x = \delta\}.$$

An equality is equivalent to two inequalities, so

$$F = \{x : Ax \le b,\ c^\top x \le \delta,\ -c^\top x \le -\delta\}.$$

This is an intersection of finitely many half-spaces, hence a polyhedron. The empty face is also a polyhedron.

Exercise 46. Prove or disprove: the projection of a polyhedron onto a coordinate subspace is always a polyhedron.

Solution. **True.** Suppose

$$P = \{(x, y) : Ax + By \le b\}.$$

Projecting onto the x-coordinates means describing

$$\pi_x(P) = \{x : \text{there exists } y \text{ with } Ax + By \le b\}.$$

Fourier–Motzkin elimination removes the components of y one at a time. At each step it combines finitely many linear inequalities and produces another finite system of linear inequalities. After all y-variables have been eliminated, the remaining finite system describes exactly $\pi_x(P)$. Therefore the projection is a polyhedron.

Exercise 47. Show that $\max\{c^\top x : x \in P\}$ is equivalent to $-\min\{-c^\top x : x \in P\}$. Convert

$$\max\ 5x_1 - 3x_2 \quad \text{s.t.} \quad x_1 + x_2 \le 4, \quad x_1, x_2 \ge 0$$

to an equivalent minimisation LP in standard form.

Solution. For every feasible x, maximizing $c^\top x$ ranks points in exactly the reverse order of minimizing $-c^\top x$. Hence

$$\max_{x \in P} c^\top x = -\min_{x \in P}(-c^\top x).$$

Introduce slack $s \geq 0$ in the inequality. A standard-form minimisation model is

$$\begin{aligned} \text{minimize} \quad & -5x_1 + 3x_2 \\ \text{subject to} \quad & x_1 + x_2 + s = 4, \\ & x_1, x_2, s \geq 0. \end{aligned}$$

If the minimum value is $\hat{z}$, the original maximum value is $-\hat{z}$ and the optimizer is the same.

Exercise 48. An LP has the constraint $|x_1 - x_2| \leq 3$. Replace it with linear constraints.

Solution. The elementary equivalence $|u| \leq 3 \iff -3 \leq u \leq 3$, with $u = x_1 - x_2$, gives

$$x_1 - x_2 \leq 3, \qquad -x_1 + x_2 \leq 3.$$

These two half-spaces form a diagonal strip of width 3 on either side of the line $x_1 = x_2$. They can be inserted directly into the LP.

Exercise 49. Given

$$\max\ 2x_1 + x_2 \quad \text{s.t.} \quad x_1 + x_2 + s_1 = 5, \quad 2x_1 - x_2 + s_2 = 4, \quad x_1, x_2, s_1, s_2 \geq 0,$$

identify which constraints are active at $(x_1, x_2) = (2, 1)$. Is it a basic feasible solution? Find the slacks.

Solution. Substitution gives

$$s_1 = 5 - 2 - 1 = 2, \qquad s_2 = 4 - 2(2) + 1 = 1.$$

Both slacks are positive, so neither original inequality $x_1 + x_2 \leq 5$ nor $2x_1 - x_2 \leq 4$ is active. Also $x_1, x_2 > 0$, so no non-negativity constraint is active.

The standard-form matrix has rank 2, whereas the feasible vector $(2, 1, 2, 1)$ has four positive components. A basic feasible solution can have at most two basic positive components here. Equivalently, no two independent constraints pin down $(2, 1)$ in the (x_1, x_2) plane. Thus $(2, 1)$ is feasible but **not** a basic feasible solution.

Exercise 50. For $3x_1 + 2x_2 \geq 12$, introduce a surplus variable $s \geq 0$. If $x_1 = 3, x_2 = 1$, find s and determine feasibility.

Solution. Subtracting a surplus variable gives

$$3x_1 + 2x_2 - s = 12, \qquad s \geq 0.$$

At $(3, 1)$ the equality would require

$$s = 3(3) + 2(1) - 12 = -1.$$

This violates $s \geq 0$. Equivalently, the original left-hand side is $11 < 12$, so the point is infeasible. A negative "surplus" is a useful signal that the lower-bound constraint has not been met.

Exercise 51. How many linearly independent constraints are active at a vertex of a polyhedron in $\mathbb{R}^n$? Give an example in $\mathbb{R}^3$ where exactly three are active.

Solution. At a vertex, the active constraint normals must span $\mathbb{R}^n$, so their rank is n. There must therefore be at least n active constraints and at least n linearly independent ones. A degenerate vertex may have more than n active constraints, but only n can be linearly independent.

For the cube $[0,1]^3$, the origin is a vertex at which exactly

$$x_1 = 0, \qquad x_2 = 0, \qquad x_3 = 0$$

are active. Their normals are the three coordinate vectors and are linearly independent.

Exercise 52. Consider

$$\max\ x_1 + x_2 + x_3 + x_4 \quad \text{s.t.} \quad x_1 + x_2 + x_3 + x_4 = 1, \quad x_i \geq 0.$$

(a) What is the dimension?

(b) List all vertices.

(c) What is the optimal value, and is the optimum unique?

Solution.

(a) The feasible set is the standard simplex in an affine hyperplane of $\mathbb{R}^4$, so its dimension is $4 - 1 = 3$.

(b) Its vertices are

$$e_1 = (1,0,0,0),\ e_2 = (0,1,0,0),\ e_3 = (0,0,1,0),\ e_4 = (0,0,0,1).$$

(c) The objective is identical to the equality's left-hand side, so it equals 1 at *every* feasible point. Thus $z^* = 1$ and the optimum is not unique; the entire simplex is optimal.

Exercise 53. Linearise

$$\min\ \max(2x_1 + x_2,\ x_1 + 3x_2) \quad \text{s.t.} \quad x_1 + x_2 \leq 4, \quad x_1, x_2 \geq 0.$$

Solution. Introduce t as an upper bound on both affine expressions:

$$\begin{aligned} \text{minimize} \quad & t \\ \text{subject to} \quad & t \geq 2x_1 + x_2, \\ & t \geq x_1 + 3x_2, \\ & x_1 + x_2 \leq 4, \\ & x_1, x_2 \geq 0. \end{aligned}$$

For fixed (x_1, x_2), minimization pushes t down until it equals the larger of the two expressions. Thus this LP is exactly equivalent to the original min–max problem. No sign restriction on t is needed, although here the other constraints imply $t \geq 0$.

Exercise 54. A factory makes products A, B. Their machine-hour requirements on (M_1, M_2) are $(2,1)$ and $(1,3)$; capacities are $(120, 150)$; profits are €5 and €4. Formulate the LP, find every vertex, and solve it graphically.

Solution. Let $x_A, x_B \geq 0$ be production quantities. The model is

$$\begin{aligned} \text{maximize} \quad & 5x_A + 4x_B \\ \text{subject to} \quad & 2x_A + x_B \leq 120 \quad (M_1), \\ & x_A + 3x_B \leq 150 \quad (M_2), \\ & x_A, x_B \geq 0. \end{aligned}$$

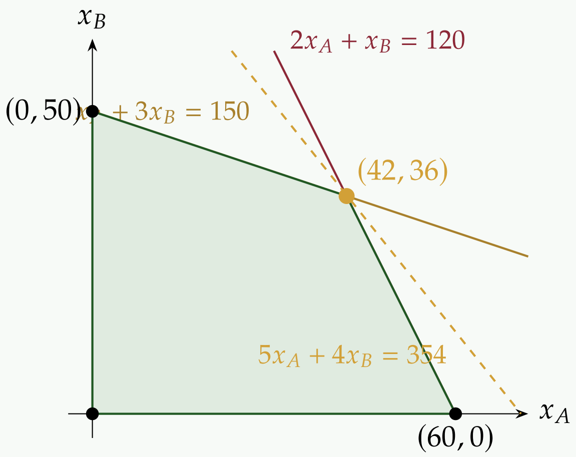
The axis vertices are $(60, 0)$ and $(0, 50)$. Solving the two binding capacity equations gives

$$2x_A + x_B = 120, \qquad x_A + 3x_B = 150 \quad \Longrightarrow \quad (x_A, x_B) = (42, 36).$$

Together with the origin, the vertices and profits are

(x_A, x_B)	$(0,0)$	$(60,0)$	$(42,36)$	$(0,50)$
$5x_A + 4x_B$	0	300	354	200.

Hence the optimum is $(42, 36)$ with profit €354.



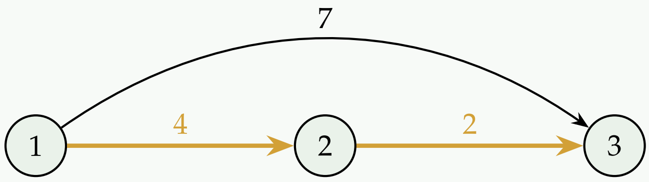
Exercise 55. For the directed graph with arcs $(1,2), (1,3), (2,3)$ and costs $c_{12} = 4, c_{13} = 7, c_{23} = 2$, formulate a shortest-path LP from node 1 to node 3.

Solution. Let $x_{ij} \geq 0$ be the amount of one unit of flow sent through arc (i, j). Send one unit from node 1 to node 3:

$$\begin{aligned} \text{minimize} \quad & 4x_{12} + 7x_{13} + 2x_{23} \\ \text{subject to} \quad & x_{12} + x_{13} = 1 \qquad \text{(source 1)}, \\ & x_{23} - x_{12} = 0 \qquad \text{(node 2)}, \\ & x_{13} + x_{23} = 1 \qquad \text{(sink 3)}, \\ & x_{12}, x_{13}, x_{23} \geq 0. \end{aligned}$$

The third balance equation is redundant once the first two hold, but it makes flow conservation explicit. The direct path costs 7, whereas $1 \to 2 \to 3$ costs $4 + 2 = 6$. Therefore

$$x_{12} = x_{23} = 1, \qquad x_{13} = 0, \qquad z^* = 6.$$



Integer Linear Programming

Exercise 1 (ILP vs. LP feasible region). Consider

$$\max\ 3x_1 + 2x_2 \quad \text{s.t.} \quad 2x_1 + x_2 \le 7, \quad x_1 + 2x_2 \le 7, \quad x_1, x_2 \in \mathbb{Z}_{\ge 0}.$$

Solve the LP relaxation, enumerate the integer feasible points, find the ILP optimum, and compute the integrality gap.

Solution. The two boundary lines meet at

$$2x_1 + x_2 = 7, \qquad x_1 + 2x_2 = 7 \quad \Longrightarrow \quad (x_1, x_2) = \left(\frac{7}{3}, \frac{7}{3}\right).$$

The LP vertices are $(0, 0)$, $(7/2, 0)$, $(7/3, 7/3)$, and $(0, 7/2)$. Their objective values are 0, $21/2$, $35/3$, and 7, so

$$x^*_{\text{LP}} = \left(\frac{7}{3}, \frac{7}{3}\right), \qquad z^*_{\text{LP}} = \frac{35}{3}.$$

The integer feasible points are

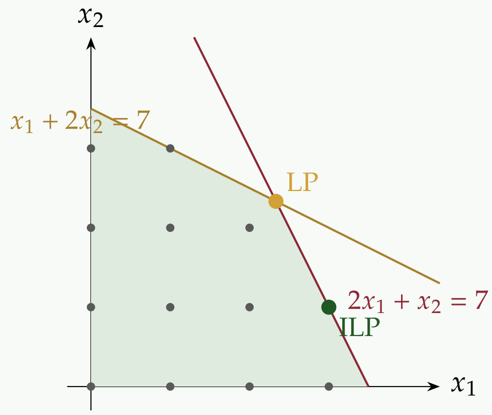
x_1	admissible x_2
0	$0, 1, 2, 3$
1	$0, 1, 2, 3$
2	$0, 1, 2$
3	$0, 1$

(13 points in total). Evaluating the objective, the best is $(3, 1)$, with

$$z^*_{\text{ILP}} = 3(3) + 2(1) = 11.$$

Therefore

$$\text{gap} = \frac{z^*_{\text{LP}}}{z^*_{\text{ILP}}} = \frac{35/3}{11} = \frac{35}{33} \approx 1.061.$$



Exercise 2 (Binary knapsack LP relaxation). A knapsack has capacity 10, weights $(6, 5, 4)$, and values $(9, 7, 5)$. Write the binary ILP, solve its LP relaxation by value density, find the ILP optimum, and compute the gap.

Solution. The ILP is

$$\max\ 9x_1 + 7x_2 + 5x_3 \quad \text{s.t.} \quad 6x_1 + 5x_2 + 4x_3 \le 10, \qquad x_i \in \{0, 1\}.$$

The densities are

$$\frac{9}{6} = 1.5, \qquad \frac{7}{5} = 1.4, \qquad \frac{5}{4} = 1.25.$$

The fractional solution takes item 1 completely and 4/5 of item 2:

$$x^{\text{LP}} = (1, 4/5, 0), \qquad z^*_{\text{LP}} = 9 + \frac{4}{5}7 = \frac{73}{5} = 14.6.$$

The feasible pair $\{1, 3\}$ fills the capacity exactly and has value 14; no other feasible selection is better. Hence

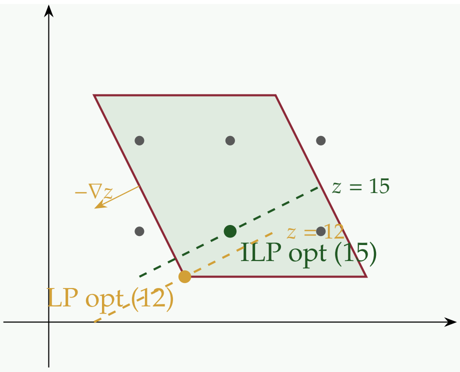
$$z^*_{\text{ILP}} = 14, \qquad \text{gap} = \frac{73/5}{14} = \frac{73}{70} \approx 1.043.$$

Exercise 3 (Integrality gap interpretation). For a minimisation ILP, $z^*_{\text{LP}} = 12$ and $z^*_{\text{ILP}} = 15$. Compute and interpret the gap and explain what a strong formulation means.

Solution. For minimisation the ratio is inverted:

$$\text{gap} = \frac{z^*_{\text{ILP}}}{z^*_{\text{LP}}} = \frac{15}{12} = \frac{5}{4} = 1.25.$$

The LP underestimates the integer optimum because it minimises over a larger feasible set. Thus the colleague's statement is correct for a feasible minimisation ILP with finite optima. A strong formulation has an LP bound close to the integer optimum, so its gap is close to 1.



Exercise 4 (Bounding lemma for maximisation ILP). Let $X = \{x \in \mathbb{Z}^n_{\geq 0} : Ax \leq b\}$ and $P = \{x \in \mathbb{R}^n_{\geq 0} : Ax \leq b\}$. Prove $X \subseteq P$ and the LP upper-bound property for maximisation.

Solution. Every $x \in X$ satisfies $Ax \leq b$ and $x \geq 0$; forgetting that its coordinates are integer leaves all those inequalities unchanged. Therefore $x \in P$ and $X \subseteq P$.

For objective $c^\top x$,

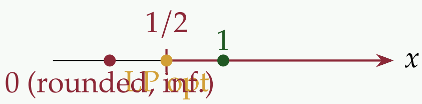
$$\max_{x \in X} c^\top x \leq \max_{x \in P} c^\top x$$

because the right-hand side maximises over a superset. Hence $z^*_{\mathrm{ILP}} \leq z^*_{\mathrm{LP}}$.

Exercise 5 (True or false: LP rounding). Assess: componentwise rounding down always gives (a) an integer-feasible point, (b) an integer optimum; (c) every binary relaxation vertex is integer; (d) LP infeasibility implies ILP infeasibility.

Solution.

(a) **False.** In $\min\{x : x \geq 1/2,\ x \in \mathbb{Z}_{\geq 0}\}$, the LP optimum $1/2$ rounds down to 0, which is infeasible.



(b) **False.** In Exercise 2, rounding $(1, 4/5, 0)$ down gives value 9, while the ILP optimum has value 14.

(c) **False.** For $2x_1 + 2x_2 \leq 3$, $0 \leq x_i \leq 1$, points such as $(1, 1/2)$ are fractional vertices.

(d) **True.** Since $X \subseteq P$, an empty relaxation P forces X to be empty.

Exercise 6 (Convex hull and ideal formulation). Let

$$X = \{(x_1, x_2) \in \mathbb{Z}^2_{\geq 0} : x_1 + x_2 \leq 3,\ x_1 \leq 2,\ x_2 \leq 2\}.$$

Enumerate X, describe $\mathrm{conv}(X)$, and explain ideality.

Solution. The integer points are

$$X = \{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1)\}.$$

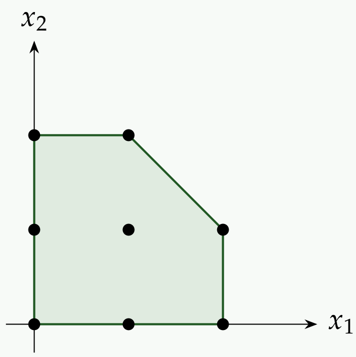
Their convex hull has vertices

$$(0,0),\ (2,0),\ (2,1),\ (1,2),\ (0,2)$$

and the minimal description

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_1 \leq 2, \quad x_2 \leq 2, \quad x_1 + x_2 \leq 3.$$

Every vertex is integer, so a linear objective reaches an optimum at an integer vertex. The relaxation over this hull therefore solves every ILP with feasible set X exactly.



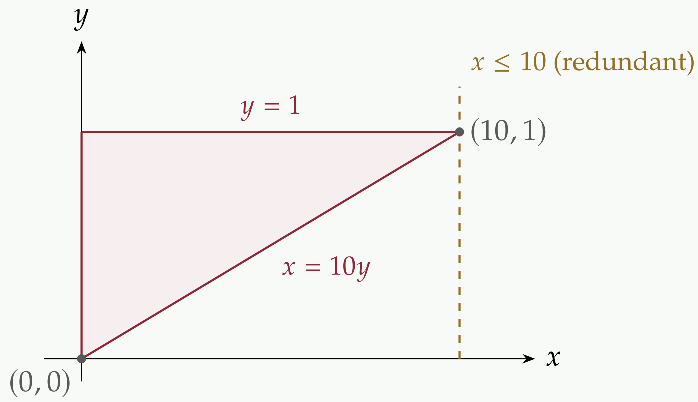
Exercise 7 (Strong vs. weak formulation). For $y \in \{0,1\}$ and $x \geq 0$, compare A: $x \leq 10y$ and B: $x \leq 10$, $x \leq 10y$, including their LP relaxations under $\max x - 5y$.

Solution. For binary y, both give $x = 0$ when $y = 0$ and $0 \leq x \leq 10$ when $y = 1$. In the relaxation, however, $0 \leq y \leq 1$ already implies

$$x \leq 10y \leq 10.$$

Thus the extra inequality $x \leq 10$ in B is redundant: the two relaxation polyhedra are identical. For fixed x, the objective prefers the smallest admissible $y = x/10$, giving $x - 5y = x/2$, maximised at $(x, y) = (10, 1)$ with value 5.

Consequently neither formulation is stronger. The premise that B should be tighter is incorrect for the formulations as written.



Exercise 8 (Comparing two ILP formulations). Compare

$$\text{I: } y_1 + y_2 + y_3 \leq 1 \quad \text{and} \quad \text{II: } y_i + y_j \leq 1 \text{ for every pair,}$$

first over binary variables and then over $[0,1]^3$.

Solution. Over $\{0,1\}^3$, both say that no two variables can equal one, hence they are equivalent. Over $[0,1]^3$, formulation I implies every pairwise inequality, but not conversely:

$$\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$$

satisfies all three pairwise inequalities and violates $y_1 + y_2 + y_3 \leq 1$. Therefore

$$P_{\text{I}} \subsetneq P_{\text{II}},$$

so formulation I is stronger.

Exercise 9 (Binary selection constraints). For six project indicators y_j, model: at most three; at least two; $3 \Rightarrow 1$; 4 and 5 incompatible; exactly one of 2 and 6.

Solution.

$$\sum_{j=1}^{6} y_j \leq 3, \qquad \sum_{j=1}^{6} y_j \geq 2,$$
$$y_3 \leq y_1, \quad y_4 + y_5 \leq 1,$$
$$y_2 + y_6 = 1, \qquad y_j \in \{0,1\}.$$

Each inequality is a direct algebraic translation of the corresponding logical rule.

Exercise 10 (Logical constraints with binary variables). Give linear formulations for AND, OR, NOT, implication, and implication to the negation of a binary variable.

Solution. With all variables binary:

condition	linear formulation
$z = y_1$ AND y_2	$z \le y_1,\ z \le y_2,\ z \ge y_1 + y_2 - 1$
$z = y_1$ OR y_2	$z \ge y_1,\ z \ge y_2,\ z \le y_1 + y_2$
$z =$ NOT y_1	$z + y_1 = 1$
$y_1 \Rightarrow y_2$	$y_1 \le y_2$
$y_1 \Rightarrow \neg y_2$	$y_1 + y_2 \le 1$

Exercise 11 (If–then constraint via Big-M). Model: if continuous $x_1 \ge 1$, then $x_2 \le 0$, where $0 \le x_i \le M_i$.

Solution. There is a subtle topological issue: the exact feasible set contains $x_1 < 1$ with arbitrary x_2, but at $x_1 = 1$ suddenly requires $x_2 = 0$. This set is not closed, so it cannot be represented exactly by a finite MILP with continuous x_1.

If x_1 has resolution $\varepsilon > 0$ (or is integer with $\varepsilon = 1$), use $y \in \{0, 1\}$ and

$$\begin{aligned} x_1 &\ge y, \\ x_1 &\le (1-\varepsilon)(1-y) + M_1 y, \\ x_2 &\le M_2(1-y). \end{aligned}$$

Then $y = 0$ permits only $x_1 \le 1 - \varepsilon$; if $x_1 \ge 1$, necessarily $y = 1$, and the last inequality forces $x_2 = 0$. Tight M_i values reduce the fractional region and strengthen the relaxation.

Exercise 12 (Either–or constraints). Enforce at least one of $3x_1 + 2x_2 \le 12$ and $x_1 + 4x_2 \le 10$ using Big-M.

Solution. With $y \in \{0, 1\}$:

$$\begin{aligned} 3x_1 + 2x_2 &\le 12 + My, \\ x_1 + 4x_2 &\le 10 + M(1-y). \end{aligned}$$

If $y = 0$, the first constraint is active; if $y = 1$, the second is active. The relaxed constraint may also happen to hold, so "or both" is allowed. In practice the two constraints should use separate, tight constants based on bounds for x_1, x_2.

Exercise 13 (Big-M conditional constraint). Link warehouse throughput q_i to opening variable z_i, with capacity C_i.

Solution. The exact linking inequality is

$$0 \le q_i \le C_i z_i, \qquad z_i \in \{0, 1\}.$$

Thus the tightest valid Big-M is $M = C_i$. Replacing it by $10C_i$ does not change binary-feasible solutions, but permits much larger q_i for fractional z_i and therefore gives a weaker LP relaxation.

Exercise 14 (At-most-k binary constraint). For eight sensor indicators, allow at most three active sensors and model the conditional rule concerning sensor 2.

Solution. The global rule is

$$\sum_{i=1}^{8} y_i \leq 3.$$

If sensor 2 is active, at most two *other* sensors may be active:

$$\sum_{i \neq 2} y_i \leq 2 + 5(1 - y_2).$$

The constant 5 is sufficient because at most seven other sensors exist and the active bound is 2. This conditional inequality is actually redundant in the presence of the global rule: when $y_2 = 1$, the global rule already gives $\sum_{i \neq 2} y_i \leq 2$. Both expressions are linear.

Exercise 15 (Linearising a product of binary variables). Linearise $w = y_1 y_2$ and generalise to a product of k binaries.

Solution. Use

$$w \leq y_1, \qquad w \leq y_2, \qquad w \geq y_1 + y_2 - 1, \qquad w \in \{0, 1\}.$$

If either input is zero, an upper bound forces $w = 0$; if both are one, the lower bound forces $w = 1$. Thus all four truth-table rows are exact.

For $w = \prod_{i=1}^{k} y_i$, one auxiliary binary and $k + 1$ constraints suffice:

$$w \leq y_i \quad (i = 1, \ldots, k), \qquad w \geq \sum_{i=1}^{k} y_i - (k - 1).$$

Exercise 16 (Fixed startup cost). A machine has fixed cost 200, variable cost $5x$, and capacity 100. Model the cost and activation, sketch the feasible set, and discuss $(x, y) = (0, 1)$.

Solution. With $y = 1$ iff the machine is on,

$$\text{minimize } 200y + 5x, \qquad 0 \leq x \leq 100y, \qquad y \in \{0, 1\}.$$

The integer-feasible set is the isolated point $(0, 0)$ together with the segment $\{(x, 1) : 0 \leq x \leq 100\}$. Its LP relaxation is the triangle $0 \leq y \leq 1$, $0 \leq x \leq 100y$.

If the only objective is to minimise this positive cost, $y = 1, x = 0$ is dominated by $y = 0, x = 0$ and is never optimal. Other model constraints could nevertheless require the machine to be switched on.

Exercise 17 (Piecewise linear objective with binary variables). For

$$c(x) = \begin{cases} 2x & 0 \le x \le 20, \\ 40 + 3(x - 20) & 20 < x \le 40, \\ 100 + 5(x - 40) & 40 < x \le 60, \end{cases}$$

formulate $\min c(x)$ subject to $x \ge 45$ using one binary per segment.

Solution. Introduce segment-specific quantities q_s and indicators d_s:

$$\begin{array}{ll} \text{minimize} & 2q_1 + (3q_2 - 20d_2) + (5q_3 - 100d_3) \\ \text{subject to} & x = q_1 + q_2 + q_3, \qquad x \ge 45, \\ & 0 \le q_1 \le 20d_1, \\ & 20d_2 \le q_2 \le 40d_2, \\ & 40d_3 \le q_3 \le 60d_3, \\ & d_1 + d_2 + d_3 = 1, \qquad d_s \in \{0, 1\}. \end{array}$$

Only one q_s can be non-zero, and its bounds place x in the corresponding interval. Since $x \ge 45$, necessarily $d_3 = 1$; the optimum is $x = 45$ with cost $100 + 5(45 - 40) = 125$.

Exercise 18 (Uncapacitated facility location). Formulate the problem with opening costs f_i, assignment costs c_{ij}, opening indicators y_i, and assignment indicators x_{ij}.

Solution.

$$\begin{array}{lll} \text{minimize} & \sum_{i=1}^{m} f_i y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} & \\ \text{subject to} & \sum_{i=1}^{m} x_{ij} = 1 & j = 1, \ldots, n, \\ & x_{ij} \le y_i & i = 1, \ldots, m, \ j = 1, \ldots, n, \\ & x_{ij}, y_i \in \{0, 1\}. & \end{array}$$

The linking constraints $x_{ij} \le y_i$ are precisely those that forbid assigning a customer to a closed facility. An equivalent aggregated link is $\sum_j x_{ij} \le n y_i$, but the disaggregated inequalities above usually give a stronger relaxation.

Exercise 19 (Assignment problem ILP). Formulate the n-worker, n-job maximum-profit assignment problem. Count variables and constraints, then instantiate $P = \begin{pmatrix} 3 & 1 & 4 \\ 2 & 5 & 2 \\ 4 & 3 & 6 \end{pmatrix}$.

Solution. Let $x_{ij} = 1$ iff worker i receives job j:

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{n} x_{ij} = 1 \qquad (i = 1, \dots, n), \\ & \sum_{i=1}^{n} x_{ij} = 1 \qquad (j = 1, \dots, n), \\ & x_{ij} \in \{0, 1\}. \end{aligned}$$

There are n^2 variables and $2n$ written equality constraints (one is linearly redundant).

For $n = 3$ the objective is

$$3x_{11} + x_{12} + 4x_{13} + 2x_{21} + 5x_{22} + 2x_{23} + 4x_{31} + 3x_{32} + 6x_{33},$$

with row equations $x_{i1} + x_{i2} + x_{i3} = 1$ for $i = 1, 2, 3$ and column equations $x_{1j} + x_{2j} + x_{3j} = 1$ for $j = 1, 2, 3$. Checking the six assignments, the identity assignment has profit $3 + 5 + 6 = 14$, the maximum.

Exercise 20 (TSP with subtour elimination). Write the directed TSP degree constraints, all SECs for $n = 4$, their general count, and explain why they are needed.

Solution. For $V = \{1, \dots, n\}$ and $i \ne j$:

$$\sum_{j \ne i} x_{ij} = 1, \qquad \sum_{j \ne i} x_{ji} = 1 \quad (i \in V), \qquad x_{ij} \in \{0, 1\}.$$

For every $S \subsetneq V$ with $2 \le |S| \le n - 1$:

$$\sum_{\substack{i,j \in S \\ i \ne j}} x_{ij} \le |S| - 1.$$

For $n = 4$, apply this inequality to the six pairs

$$\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}$$

and the four triples

$$\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}.$$

Thus the number written in this subset formulation is

$$\sum_{s=2}^{n-1} \binom{n}{s} = 2^n - n - 2.$$

Without SECs, $x_{12} = x_{21} = x_{34} = x_{43} = 1$ satisfies every in/out-degree equation but consists of two disconnected 2-cycles.

Exercise 21 (Graph coloring ILP). For the graph with edges $12, 13, 23, 34, 45$ and at most three colours, write the coloring ILP and determine whether three colours suffice.

Solution. For $v = 1, \ldots, 5$ and $c = 1, 2, 3$:

$$\begin{array}{lll} \text{minimize} & w_1 + w_2 + w_3 & \\ \text{subject to} & \sum_{c=1}^{3} x_{vc} = 1 & v = 1, \ldots, 5, \\ & x_{uc} + x_{vc} \le 1 & uv \in E,\ c = 1, 2, 3, \\ & x_{vc} \le w_c & v = 1, \ldots, 5,\ c = 1, 2, 3, \\ & x_{vc}, w_c \in \{0, 1\}. & \end{array}$$

Vertices $1, 2, 3$ form a triangle, so at least three colours are needed. They also suffice: colour $1, 2, 3$ differently, give vertex 4 the colour of vertex 1, and vertex 5 the colour of vertex 2. Hence $\chi(G) = 3$.

Exercise 22 (Maximum weighted independent set). For the weighted 5-cycle with weights $(3, 5, 2, 4, 1)$, formulate and solve the ILP and its edge-relaxation.

Solution.

$$\begin{array}{rl} \text{maximize} & 3x_1 + 5x_2 + 2x_3 + 4x_4 + x_5 \\ \text{subject to} & x_1 + x_2 \le 1,\ x_2 + x_3 \le 1,\ x_3 + x_4 \le 1, \\ & x_4 + x_5 \le 1,\ x_5 + x_1 \le 1, \qquad x_v \in \{0, 1\}. \end{array}$$

The point $x_v = 1/2$ is LP-feasible, but its value is only 7.5; the hint does not identify the optimum because the weights are not symmetric. The extreme points of the edge relaxation of an odd cycle are its integer stable sets plus the all-half point. The best integer stable set is $\{2, 4\}$ with weight $5 + 4 = 9$, exceeding 7.5. Therefore

$$z^*_{\text{LP}} = z^*_{\text{ILP}} = 9, \qquad \text{gap} = 1.$$

Exercise 23 (Bin packing ILP). Formulate bin packing with item sizes s_i, capacity C, assignment variables x_{ik}, and bin indicators y_k. Pack $(4, 3, 3, 2)$ for $C = 6$ using three and then two bins.

Solution. Using at most n candidate bins:

$$\begin{aligned}
\text{minimize} \quad & \sum_{k=1}^{n} y_k \\
\text{subject to} \quad & \sum_{k=1}^{n} x_{ik} = 1 && i = 1, \dots, n, \\
& \sum_{i=1}^{n} s_i x_{ik} \le C y_k && k = 1, \dots, n, \\
& x_{ik} \le y_k && i, k = 1, \dots, n, \\
& x_{ik}, y_k \in \{0, 1\}.
\end{aligned}$$

A 3-bin packing is $(4, 2)$, (3), (3). A 2-bin packing is $(4, 2)$ and $(3, 3)$, both exactly at capacity. Thus two bins are optimal because total size is $12 = 2C$.

Exercise 24 (Set covering model). For the given 8×6 coverage matrix, formulate set covering, give its relaxation, and test $y = (1, 0, 1, 1, 0, 0)$.

Solution. The compact formulation is

$$\min \sum_{i=1}^{6} y_i, \qquad \sum_{i=1}^{6} A_{ji} y_i \ge 1 \quad (j = 1, \dots, 8), \qquad y_i \in \{0, 1\}.$$

The LP relaxation replaces the last condition by $0 \le y_i \le 1$.

For the proposed vector, rows 1–6 and 8 are covered, but row 7 gives

$$y_2 + y_5 = 0 < 1.$$

Therefore the vector is **not feasible**.

Exercise 25 (Set packing ILP). For $S_1 = \{1, 2\}$, $S_2 = \{2, 3, 4\}$, $S_3 = \{1, 3\}$, $S_4 = \{4, 5\}$, $S_5 = \{1, 5\}$ with weights $(3, 5, 2, 4, 3)$, formulate and solve set packing.

Solution. With $y_j = 1$ iff S_j is selected:

$$\begin{aligned}
\max \quad & 3y_1 + 5y_2 + 2y_3 + 4y_4 + 3y_5 \\
\text{s.t.} \quad & y_1 + y_3 + y_5 \le 1 && (1), \\
& y_1 + y_2 \le 1 && (2), \\
& y_2 + y_3 \le 1 && (3), \\
& y_2 + y_4 \le 1 && (4), \\
& y_4 + y_5 \le 1 && (5), \\
& y_j \in \{0, 1\}.
\end{aligned}$$

Besides the empty set and five singleton packings, the only feasible pairs are

$$\{S_1, S_4\}\ (7), \qquad \{S_2, S_5\}\ (8), \qquad \{S_3, S_4\}\ (6).$$

No triple is feasible, so the optimum is $\{S_2, S_5\}$ with value 8. Covering uses ≥ 1; packing flips the direction to ≤ 1.

Exercise 26 (Multi-index scheduling model). For $x_{ijk} = 1$ iff nurse i works shift j on day k, model coverage, at most one shift per nurse per day, and the overnight-rest rule.

Solution. For nurses $i = 1, 2, 3$, shifts $j = 1, \ldots, 4$, days $k = 1, 2$:

$$\begin{aligned}
\sum_{i=1}^{3} x_{ijk} &\geq 1 && j = 1, \ldots, 4,\ k = 1, 2, \\
\sum_{j=1}^{4} x_{ijk} &\leq 1 && i = 1, 2, 3,\ k = 1, 2, \\
x_{i,4,1} + x_{i,1,2} &\leq 1 && i = 1, 2, 3, \\
x_{ijk} &\in \{0, 1\}.
\end{aligned}$$

The last inequality directly excludes the two-shift combination and needs no Big-M.

Exercise 27 (Bounded integer knapsack). For capacity 15, weights $(4, 3, 5, 2)$, values $(7, 5, 9, 3)$, and upper bounds $(2, 3, 1, 4)$, formulate the bounded ILP, give a binary expansion, and compute the LP bound.

Solution.

$$\begin{aligned}
\text{maximize} \quad & 7x_1 + 5x_2 + 9x_3 + 3x_4 \\
\text{subject to} \quad & 4x_1 + 3x_2 + 5x_3 + 2x_4 \leq 15, \\
& 0 \leq x_i \leq u_i, \qquad x_i \in \mathbb{Z}.
\end{aligned}$$

A binary expansion is

$$\begin{aligned}
x_1 &= b_{11} + 2b_{12} \leq 2, & x_2 &= b_{21} + 2b_{22} \leq 3, \\
x_3 &= b_{31}, & x_4 &= b_{41} + 2b_{42} + 4b_{43} \leq 4,
\end{aligned}$$

with all b binary. The explicit upper inequalities remove binary codes above u_i.

The densities are $9/5, 7/4, 5/3$, and $3/2$. Fractionally take one copy of item 3, two of item 1, and $2/3$ of item 2:

$$z^*_{\text{LP}} = 9 + 14 + \frac{2}{3}5 = \frac{79}{3} \approx 26.33.$$

Exercise 28 (SAT encoded as ILP). Encode $(y_1 \vee \neg y_2), (\neg y_1 \vee y_2 \vee y_3), (\neg y_2 \vee \neg y_3)$ as an ILP and test satisfiability.

Solution. The clause inequalities are

$$\begin{aligned}
y_1 + (1 - y_2) \geq 1 &\iff & y_1 - y_2 &\geq 0, \\
(1 - y_1) + y_2 + y_3 \geq 1 &\iff & -y_1 + y_2 + y_3 &\geq 0, \\
(1 - y_2) + (1 - y_3) \geq 1 &\iff & -y_2 - y_3 &\geq -1.
\end{aligned}$$

Together with $y_i \in \{0,1\}$ and objective $\max 0$, they form the feasibility ILP. The assignment $(y_1, y_2, y_3) = (0,0,0)$ satisfies all three clauses, so the formula is satisfiable.

Exercise 29 (Modelling checklist application). A factory can produce products A and B; A uses machines M1–M2, B uses M2–M3, capacities are limited, and each product has a fixed startup cost. Give a generic complete MILP.

Solution. Let $x_A, x_B \geq 0$ be production, $y_A, y_B \in \{0,1\}$ activation, p_A, p_B unit profits, F_A, F_B fixed costs, a_{mP} processing requirements, C_m capacities, and U_P tight production bounds:

$$\begin{aligned} \text{maximize} \quad & p_A x_A + p_B x_B - F_A y_A - F_B y_B \\ \text{subject to} \quad & a_{1A} x_A \leq C_1, \\ & a_{2A} x_A + a_{2B} x_B \leq C_2, \\ & a_{3B} x_B \leq C_3, \\ & 0 \leq x_A \leq U_A y_A, \qquad 0 \leq x_B \leq U_B y_B, \\ & y_A, y_B \in \{0,1\}. \end{aligned}$$

The capacity constraints describe resources; the last two inequalities are the coupling constraints linking continuous production to binary startup decisions.

Exercise 30 (Maximum satisfiability as ILP). For the preceding three clauses, formulate MAX-SAT with indicators $s_k = 1$ when clause k is counted as satisfied.

Solution. Let the sums of literals be

$$L_1 = y_1 + 1 - y_2, \quad L_2 = 1 - y_1 + y_2 + y_3, \quad L_3 = 2 - y_2 - y_3.$$

The correct maximisation formulation is

$$\max\ s_1 + s_2 + s_3, \qquad L_k \geq s_k\ (k = 1,2,3), \qquad y_i, s_k \in \{0,1\}.$$

Because the objective rewards $s_k = 1$, it is set to one whenever the clause has at least one true literal. If an exact "if and only if" encoding is required independently of the objective, also impose $L_1 \leq 2s_1$, $L_2 \leq 3s_2$, and $L_3 \leq 2s_3$.

The suggested form "$L_k + s_k \geq 1$" would make s_k indicate an *unsatisfied* clause, so it has the wrong polarity for the stated meaning.

Exercise 31 (Complement and mutual exclusion). Let y_1, y_2 be binary variables, and let z_i indicate whether event E_i occurs. Express complementarity, mutual exclusion, and exclusive choice linearly.

Solution. For binary variables, the three requirements are respectively

$$y_1 + y_2 = 1, \qquad z_1 + z_2 + z_3 \leq 1, \qquad z_1 + z_2 + z_3 = 1.$$

The inequality says "zero or one event", whereas the equality rules out the case in which no event occurs.

Exercise 32 (Linearising a product of binary and continuous). Let $y \in \{0, 1\}$, $0 \le x \le U$, and $w = yx$. Give and verify an exact linear formulation.

Solution. The convex-hull formulation is

$$0 \le w, \qquad w \le x, \qquad w \le Uy, \qquad w \ge x - U(1 - y).$$

If $y = 0$, the third inequality and nonnegativity give $w = 0$. If $y = 1$, the second and fourth give $w \le x$ and $w \ge x$, hence $w = x$.

The constant U must be valid, but should be as small as possible. An unnecessarily large U admits more fractional points when $0 < y < 1$, weakening the LP relaxation.

Exercise 33 (Lower bound on chromatic number via cliques). Prove the clique lower bound and apply it to the graph with edges $12, 13, 23, 34, 45$.

Solution. All vertices of a clique are pairwise adjacent, so no two of them may receive the same color. A clique of size r therefore needs r distinct colors, proving

$$\chi(G) \ge \omega(G).$$

Here $\{1, 2, 3\}$ is a triangle, hence $\omega(G) \ge 3$. There is no larger clique: vertex 4 is adjacent to 3 and 5 but not to 1 or 2, and vertex 5 is adjacent only to 4. Thus $\omega(G) = 3$.

A matching upper bound is obtained, for example, with

$$c(1) = A, \quad c(2) = B, \quad c(3) = C, \quad c(4) = A, \quad c(5) = B.$$

Consequently $\chi(G) = 3$.

Exercise 34 (Stepwise fixed cost model). A line has a low-speed mode with capacity 50 and cost $100 + 3x$, and a high-speed mode with capacity 120 and cost $250 + 2x$. Exactly one mode is selected.

Solution. Using the prescribed binary d and mode-specific production variables x_1, x_2, an entirely linear model is

$$\begin{aligned} & x = x_1 + x_2, \\ & 0 \le x_1 \le 50(1 - d), \qquad 0 \le x_2 \le 120d, \\ & d \in \{0, 1\}, \\ & \text{minimize} \quad 100(1 - d) + 3x_1 + 250d + 2x_2. \end{aligned}$$

Thus $d = 0$ activates only mode 1 and $d = 1$ only mode 2.

Algebraically, the two cost lines intersect when

$$100 + 3x = 250 + 2x, \qquad \text{so } x^* = 150.$$

This point is outside the capacity of mode 1. Therefore, wherever both modes are feasible ($0 \le x \le 50$), mode 1 is cheaper. At demand $x = 80$, mode 1 is infeasible and mode 2 is the only choice, with cost $250 + 2(80) = 410$.

Exercise 35 (Multi-dimensional knapsack). Formulate the five-item knapsack with weight and volume capacities, solve its weight-only LP relaxation, and compare the two LP bounds.

Solution. With $x_j = 1$ when item j is selected, the ILP is

$$\begin{aligned} \text{maximize} \quad & 5x_1 + 4x_2 + 3x_3 + 7x_4 + 6x_5 \\ \text{subject to} \quad & 3x_1 + 4x_2 + 2x_3 + 5x_4 + 3x_5 \le 10, \\ & 2x_1 + 3x_2 + 4x_3 + 3x_4 + 5x_5 \le 8, \\ & x_j \in \{0,1\} \quad (j = 1, \dots, 5). \end{aligned}$$

For the weight-only LP, order items by value per unit of weight:

$$5,\ 1,\ 3,\ 4,\ 2 \quad \text{with ratios} \quad 2,\ \frac{5}{3},\ \frac{3}{2},\ \frac{7}{5},\ 1.$$

Taking items $5, 1, 3$ and $2/5$ of item 4 fills the capacity:

$$x = (1, 0, 1, \tfrac{2}{5}, 1), \qquad z^*_{\text{LP,W}} = 6 + 5 + 3 + \frac{2}{5}7 = \frac{84}{5} = 16.8.$$

Adding the volume constraint can only remove feasible LP points, so

$$z^*_{\text{LP,WV}} \le z^*_{\text{LP,W}}.$$

No re-optimisation is needed to establish this comparison.

Exercise 36 (Integrality gap of two formulations). An ILP has value 10, while two LP relaxations have values 14 and 11. Compare the formulations.

Solution. For this maximisation problem,

$$\text{gap}(F_1) = \frac{14}{10} = 1.4, \qquad \text{gap}(F_2) = \frac{11}{10} = 1.1.$$

Formulation F_2 supplies the tighter upper bound and is therefore the stronger formulation in the objective direction. It is normally the better starting point for branch-and-bound: the remaining uncertainty is only 10% rather than 40%, so fewer nodes are typically needed to prove optimality.

Exercise 37 (LP relaxation of a small ILP). Solve the LP relaxation of

$$\max\{5x_1 + 8x_2 : x_1 + x_2 \le 6,\ 5x_1 + 9x_2 \le 45,\ x \ge 0,\ x \in \mathbb{Z}^2\},$$

then compare rounding and the integer optimum.

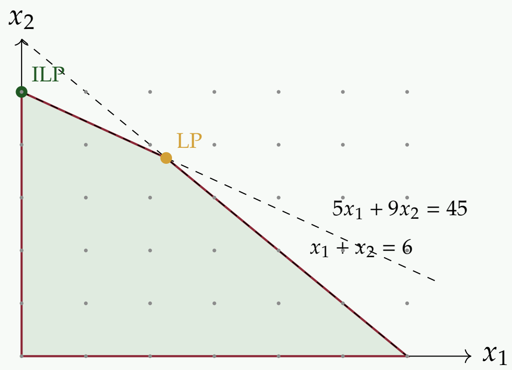
Solution. The two boundary lines intersect at

$$x_1 + x_2 = 6, \qquad 5x_1 + 9x_2 = 45,$$

which gives $(x_1, x_2) = (9/4, 15/4)$. Its value is

$$z^*_{\text{LP}} = 5\frac{9}{4} + 8\frac{15}{4} = \frac{165}{4} = 41.25.$$

It dominates the axis vertices $(6, 0)$, of value 30, and $(0, 5)$, of value 40.



Rounding each coordinate to the nearest integer gives $(2, 4)$, but $5(2) + 9(4) = 46 > 45$, so naive rounding is infeasible. Rounding down gives the feasible point $(2, 3)$ of value 34, but it is not optimal. Checking integer values of $x_2 = 0, \dots, 5$ gives the optimum

$$(x_1, x_2) = (0, 5), \qquad z^*_{\text{ILP}} = 40.$$

The integrality gap is

$$\frac{z^*_{\text{LP}}}{z^*_{\text{ILP}}} = \frac{41.25}{40} = \frac{33}{32}.$$

Exercise 38 (Capital budgeting with logical dependencies). Formulate and solve the five-project capital-budgeting instance, including the dependency of project 5 and the exclusion between projects 1 and 3.

Solution. Let y_j indicate whether project j is selected. The complete model is

$$\begin{aligned} \text{maximize} \quad & 12y_1 + 18y_2 + 9y_3 + 14y_4 + 20y_5 \\ \text{subject to} \quad & 4y_1 + 6y_2 + 3y_3 + 5y_4 + 8y_5 \le 15, \\ & y_5 \le y_2, \qquad y_5 \le y_4, \\ & y_1 + y_3 \le 1, \\ & y_j \in \{0, 1\} \quad (j = 1, \dots, 5). \end{aligned}$$

If project 5 were selected, projects 2 and 4 would also be required, using $8 + 6 + 5 = 19 > 15$ units of capital. Hence $y_5 = 0$ in every feasible solution. Among the first four projects, the best feasible choice is

$$\{1, 2, 4\}, \qquad \text{capital } 4 + 6 + 5 = 15, \qquad \text{NPV } 12 + 18 + 14 = 44.$$

The main competitor $\{2, 3, 4\}$ has NPV 41, so the optimum is 44.

Exercise 39 (Nurse rostering (small instance)). Formulate the four-nurse, three-shift assignment problem with nurse 4 unavailable at night, and solve it.

Solution. Let $x_{ij} = 1$ when nurse i covers shift j, where $j = 1, 2, 3$ denotes morning, afternoon, and night. Exclude x_{43}:

$$\begin{aligned}
\text{minimize}\quad & 2x_{11} + 3x_{12} + 5x_{13} + 4x_{21} + x_{22} + 3x_{23} \\
& + 3x_{31} + 4x_{32} + 2x_{33} + x_{41} + 2x_{42} \\
\text{subject to}\quad & \sum_{i:(i,j)\neq(4,3)} x_{ij} = 1 && (j = 1, 2, 3), \\
& \sum_{j:(i,j)\neq(4,3)} x_{ij} \leq 1 && (i = 1, \dots, 4), \\
& x_{ij} \in \{0, 1\}.
\end{aligned}$$

Omitting x_{43} is itself the unavailability constraint; equivalently one could include it and impose $x_{43} = 0$.

The independent minimum costs of the three shifts are $1, 1, 2$, and they are attained by three different nurses: nurse 4 in the morning, nurse 2 in the afternoon, and nurse 3 at night. This feasible roster has cost 4, equal to the obvious lower bound, so it is optimal.

Exercise 40 (Capacitated vehicle routing: formulation setup). Give the degree and load constraints for one capacitated route from a depot through four customers.

Solution. Let $N = \{1, \dots, 4\}$ and node 0 be the depot. For every customer,

$$\sum_{\substack{j\in N\cup\{0\}\\ j\neq i}} x_{ij} = 1, \qquad \sum_{\substack{j\in N\cup\{0\}\\ j\neq i}} x_{ji} = 1 \quad (i \in N).$$

For one route, the depot also has one departure and one return:

$$\sum_{j\in N} x_{0j} = 1, \qquad \sum_{i\in N} x_{i0} = 1.$$

Define u_i as the cumulative delivered load after customer i. The capacitated MTZ constraints are

$$\begin{aligned}
d_i \leq u_i \leq Q \qquad & (i \in N), \\
u_j \geq u_i + d_j - Q(1 - x_{ij}) \quad & (i, j \in N,\ i \neq j).
\end{aligned}$$

When $x_{ij} = 1$, the load must increase by d_j; when the arc is not used, the Big-M term deactivates the relation. Positive demand makes a customer-only cycle impossible, while $u_i \leq Q$ simultaneously enforces vehicle capacity. A one-vehicle solution of course requires $\sum_i d_i \leq Q$.

Exercise 41 (Concave piecewise linear minimisation). Linearise the three-segment production cost on $[0, 90]$ using binary segment selection.

Solution. Let $d_s = 1$ when segment s is active, and let q_s equal the actual output on that segment. An exact disjunctive formulation is

$$\begin{aligned}
&d_1 + d_2 + d_3 = 1, \qquad d_s \in \{0,1\},\\
&0 \le q_1 \le 30d_1,\\
&30d_2 \le q_2 \le 60d_2,\\
&60d_3 \le q_3 \le 90d_3,\\
&x = q_1 + q_2 + q_3,\\
&c = 4q_1 + (3q_2 + 30d_2) + (2q_3 + 90d_3).
\end{aligned}$$

For example, if $d_2 = 1$, then $x = q_2 \in [30, 60]$ and $c = 3x+30 = 120+3(x-30)$. The other two cases follow identically.

Exercise 42 (Strengthening the independent set formulation). For a triangle, compare the edge inequalities with a clique inequality.

Solution. The three edge constraints are

$$x_u + x_v \le 1, \qquad x_u + x_w \le 1, \qquad x_v + x_w \le 1.$$

Because an independent set contains at most one vertex of a clique, the stronger valid inequality is

$$x_u + x_v + x_w \le 1.$$

The fractional point $(1/2, 1/2, 1/2)$ satisfies every edge inequality at equality, but its sum is $3/2 > 1$. The clique inequality therefore removes a fractional point without removing any integer-feasible point.

Exercise 43 (Semi-continuous variable modelling). Model a variable that is either zero or lies in $[L, U]$, where $0 < L < U$.

Solution. With $y = 1$ meaning "on", use

$$Ly \le x \le Uy, \qquad y \in \{0,1\}.$$

For $y = 0$, both bounds force $x = 0$. For $y = 1$, they reduce to $L \le x \le U$.

No system containing only continuous linear variables can project exactly to $\{0\} \cup [L, U]$: every continuous LP feasible region and every linear projection of it is convex, whereas this set is nonconvex. A discrete device such as a binary variable is essential.

Exercise 44 (TSP vs. Hamiltonian path ILP). Explain the degree and subtour changes needed to turn a directed TSP model into a Hamiltonian-path model, count the relevant SECs for four nodes, and write the TSP degree constraints.

Solution. If path endpoints s and t are fixed, replace their TSP degrees by

	indegree	outdegree
s	0	1
t	1	0

and retain indegree = outdegree = 1 at every other vertex. The objective then contains $n-1$ selected arcs rather than n. A customer-only directed cycle can involve only internal vertices, so SECs are needed for

$$S \subseteq V \setminus \{s,t\}, \qquad |S| \geq 2.$$

For $n = 4$, the TSP's standard family has $2^4 - 4 - 2 = 10$ inequalities (six subsets of size 2 and four of size 3). With fixed path endpoints, only the subset consisting of the two internal vertices remains, so one SEC suffices. If endpoints are *not* fixed and are chosen by binary variables, the safe standard formulation retains all ten SECs; thus the requested count depends on this modelling convention.

For the given four-node directed TSP, the outgoing-degree equations are

$$x_{12} + x_{13} + x_{14} = 1, \quad x_{21} + x_{23} + x_{24} = 1,$$
$$x_{31} + x_{32} + x_{34} = 1, \quad x_{41} + x_{42} + x_{43} = 1,$$

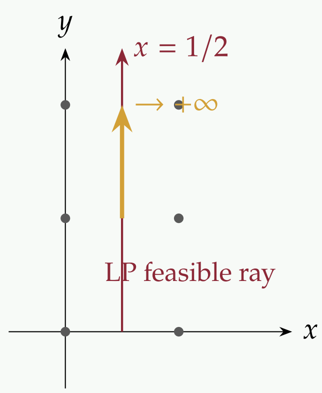
and the incoming-degree equations are

$$x_{21} + x_{31} + x_{41} = 1, \quad x_{12} + x_{32} + x_{42} = 1,$$
$$x_{13} + x_{23} + x_{43} = 1, \quad x_{14} + x_{24} + x_{34} = 1.$$

Exercise 45 (True or false: ILP theory). Classify five statements about relaxations, formulations, convex hulls, and valid inequalities.

Solution.

1. **False.** An LP relaxation may be infeasible or unbounded, so it need not have an optimal solution.
2. **False as stated.** For example, maximise y subject to $2x = 1$, $y \geq 0$, and $x, y \in \mathbb{Z}$. Its LP relaxation is feasible and unbounded, but the ILP is infeasible because no integer x satisfies $2x = 1$. Extra assumptions, including integer feasibility, are needed for the usual unboundedness implication.



3. **False.** Strong formulations often become stronger by adding valid inequalities, so they may have more constraints.
4. **False.** The convex hull of $\{0,1\}^n$ is simply $[0,1]^n$, which has only $2n$ bound inequalities.
5. **True.** By definition, a valid inequality is satisfied by every integer-feasible solution; adding it can remove only points from the relaxation,

not valid integer solutions.

Exercise 46 (LP relaxation bound for graph coloring). Analyse the standard assignment formulation on the triangle K_3 and explain what its LP relaxation says about formulation strength.

Solution. The numerical conclusion stated in the question is inconsistent with the standard formulation. With two colors, the fractional assignment

$$x_{v1} = x_{v2} = \frac{1}{2} \qquad (v = 1, 2, 3)$$

satisfies every assignment equation and every edge inequality. The linking constraints $x_{vk} \le w_k$, however, imply $w_1, w_2 \ge 1/2$. Therefore its objective is

$$w_1 + w_2 = 1,$$

not zero. This is also the LP optimum, because for any vertex $1 = x_{v1} + x_{v2} \le w_1 + w_2$.

The integer model with only $k = 2$ colors is *infeasible*; it cannot have integer optimum 3 because a third color variable is absent. If instead $k = 3$, the integer optimum is 3, while $x_{vk} = w_k = 1/3$ gives LP value 1. This large discrepancy illustrates the weakness and color symmetry of the standard assignment relaxation: fractional vertices can spread themselves evenly over all colors.

Exercise 47 (Radio tower coverage). Formulate the weighted set-covering model for the seven zones and five tower locations, relax it, and test $y = (1, 1, 0, 1, 0)$.

Solution. Let $y_i = 1$ when tower i is installed. The ILP is

$$\begin{aligned}
\text{minimize} \quad & 3y_1 + 2y_2 + 4y_3 + 2y_4 + 3y_5 \\
\text{subject to} \quad & y_1 + y_3 + y_5 \ge 1, \\
& y_1 + y_2 \ge 1, \\
& y_2 + y_3 \ge 1, \\
& y_3 + y_4 \ge 1, \\
& y_4 + y_5 \ge 1, \\
& y_1 + y_5 \ge 1, \\
& y_2 + y_4 \ge 1, \\
& y_i \in \{0, 1\} \quad (i = 1, \ldots, 5).
\end{aligned}$$

The LP relaxation replaces the last line by $0 \le y_i \le 1$.

For the proposed vector, the seven left-hand sides are

$$1,\ 2,\ 1,\ 1,\ 1,\ 1,\ 2.$$

All are at least one, so the vector is feasible. It installs towers $1, 2, 4$ at total cost $3 + 2 + 2 = 7$.

Exercise 48 (k-median facility location). Write the complete k-median formulation and compare it with uncapacitated facility location.

Solution. Let $y_i = 1$ if candidate i is opened and $x_{ij} = 1$ if client j is assigned to it. The model is

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{4}\sum_{j=1}^{5} c_{ij}x_{ij} \\ \text{subject to} \quad & \sum_{i=1}^{4} x_{ij} = 1 && (j = 1,\dots,5), \\ & x_{ij} \le y_i && (i = 1,\dots,4,\ j = 1,\dots,5), \\ & \sum_{i=1}^{4} y_i = k, \\ & x_{ij}, y_i \in \{0,1\}. \end{aligned}$$

The linking constraints prevent assignment to a closed facility. Unlike uncapacitated facility location, standard k-median has no fixed opening costs in the objective and fixes the number of open facilities exactly to k. Facility location instead lets the opening costs determine how many facilities are worthwhile.

Exercise 49 (Stronger formulation implies smaller integrality gap). Let $P_2 \subseteq P_1$ be two relaxations of the same maximisation ILP. Compare their bounds and gaps, and give an example where the inclusion is strict but the bounds coincide.

Solution. Since maximisation over a subset cannot improve the objective,

$$z_1 = \max_{x \in P_1} c^\top x \ge z_2 = \max_{x \in P_2} c^\top x.$$

Both sets contain every integer-feasible point, hence

$$z_1 \ge z_2 \ge z^*_{\text{ILP}}.$$

Assuming $z^*_{\text{ILP}} > 0$, division by it gives

$$\frac{z_2}{z^*_{\text{ILP}}} \le \frac{z_1}{z^*_{\text{ILP}}},$$

so formulation 2 has no larger integrality gap.

Strict containment need not improve a particular objective. For

$$P_1 = [0,1] \times [0, \tfrac{3}{2}], \qquad P_2 = [0,1] \times [0,1],$$

with both variables required to be integer, the integer-feasible set is $\{0,1\}^2$ in both formulations. For objective $\max x_1$, both LP bounds equal 1 even though $P_2 \subsetneq P_1$.

Exercise 50 (Bin packing LP relaxation lower bound). Formulate and solve the LP relaxation for item sizes $(0.4, 0.5, 0.6, 0.7)$ and exhibit an optimal integer packing.

Solution. Using at most four bins, let $x_{ik} = 1$ if item i is placed in bin k, and let $y_k = 1$ if bin k is used:

$$\begin{aligned} \text{minimize} \quad & \sum_{k=1}^{4} y_k \\ \text{subject to} \quad & \sum_{k=1}^{4} x_{ik} = 1 && (i = 1, \dots, 4), \\ & 0.4x_{1k} + 0.5x_{2k} \\ & \qquad + 0.6x_{3k} + 0.7x_{4k} \le y_k && (k = 1, \dots, 4), \\ & x_{ik} \le y_k && (i = 1, \dots, 4;\ k = 1, \dots, 4), \\ & x_{ik} \in \{0, 1\} && (i = 1, \dots, 4;\ k = 1, \dots, 4), \\ & y_k \in \{0, 1\} && (k = 1, \dots, 4). \end{aligned}$$

Summing the capacity constraints shows that every LP solution satisfies

$$\sum_k y_k \ge 0.4 + 0.5 + 0.6 + 0.7 = 2.2.$$

This bound is attained. Set

$$(y_1, y_2, y_3, y_4) = (1, 1, \tfrac{1}{5}, 0)$$

and, for every item i, set

$$(x_{i1}, x_{i2}, x_{i3}, x_{i4}) = (\tfrac{5}{11}, \tfrac{5}{11}, \tfrac{1}{11}, 0).$$

Every item is fully assigned, and each capacity constraint is tight. Thus $z^*_{\text{LP}} = 2.2$. Since the integer objective is integral, this proves the lower bound $\lceil 2.2 \rceil = 3$ bins.

Three bins are sufficient:

$$\{0.4, 0.6\}, \qquad \{0.5\}, \qquad \{0.7\}.$$

Hence the integer optimum is 3.

Exercise 51 (At-least-one and threshold constraints). Express four logical rules involving seven binary task variables.

Solution. The first three rules are

$$y_2 + y_4 + y_6 \ge 1, \qquad y_3 + y_5 \ge y_1, \qquad y_6 = y_7.$$

For the final rule, introduce $z_0, z_1, z_2 \in \{0, 1\}$ and impose

$$z_0 + z_1 + z_2 = 1, \qquad \sum_{i=1}^{7} y_i = 0z_0 + 3z_1 + 6z_2.$$

Exactly one indicator is active, so the total is forced to be one of $0, 3, 6$.

Exercise 52 (Bounding variables for valid Big-M). Find tight constants for a basic activation constraint and for a bounded either–or disjunction.

Solution. If $0 \le x \le 50$, the tightest constant in $x \le My$ is $M = 50$. Any smaller value wrongly excludes feasible production when $y = 1$; any larger value merely weakens the relaxation.

If x has no finite upper bound, no finite M can preserve every possible value when $y = 1$. One must derive a genuine bound from the application, use an indicator constraint supported by the solver, or reformulate the model.

For the first disjunct, the largest violation over $[0, 10]^2$ is

$$\max(2x_1 - x_2 - 4) = 2(10) - 0 - 4 = 16.$$

For the second it is

$$\max(x_1 + 3x_2 - 9) = 10 + 3(10) - 9 = 31.$$

Thus a tight Big-M formulation is

$$\begin{aligned} 2x_1 - x_2 &\le 4 + 16y, \\ x_1 + 3x_2 &\le 9 + 31(1 - y), \\ y &\in \{0, 1\}. \end{aligned}$$

When $y = 0$ the first inequality is active; when $y = 1$ the second is active.

Exercise 53 (MILP: production with setup). Model production and setup time for three products on two machines, including activation links and costs.

Solution. For each machine $i = 1, 2$, capacity must cover both processing and setup:

$$\sum_{j=1}^{3} \big(a_{ij}x_{ij} + s_{ij}y_{ij}\big) \le C_i.$$

A valid product-specific link is

$$0 \le x_{ij} \le \frac{C_i}{a_{ij}}\, y_{ij} \qquad (i = 1, 2,\ j = 1, 2, 3).$$

The complete cost model from the data provided is therefore

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{2}\sum_{j=1}^{3} \big(c_{ij}x_{ij} + f_{ij}y_{ij}\big) \\ \text{subject to} \quad & \sum_{j=1}^{3} \big(a_{ij}x_{ij} + s_{ij}y_{ij}\big) \le C_i \quad (i = 1, 2), \\ & 0 \le x_{ij} \le (C_i/a_{ij})y_{ij} \qquad (i = 1, 2,\ j = 1, 2, 3), \\ & y_{ij} \in \{0, 1\}. \end{aligned}$$

In an operational model, demand or production-target constraints must also be supplied; without them, minimising nonnegative costs naturally selects zero production.

Exercise 54 (Clique inequalities as valid inequalities). Explain how clique inequalities strengthen the independent-set relaxation and whether they make it integral for every graph.

Solution. For a clique of size two, $Q = \{u, v\}$, the clique inequality is exactly the edge constraint $x_u + x_v \leq 1$.

For a triangle, the three edge inequalities admit

$$x_u = x_v = x_w = \frac{1}{2},$$

because every pair sums to one. The clique inequality $x_u + x_v + x_w \leq 1$ rejects this point since its left-hand side is 3/2.

Even all maximal-clique inequalities do not describe the independent-set polytope of every graph. In a five-cycle C_5, all maximal cliques are just edges, and the all-half point satisfies them. Its total value is 5/2, whereas every integer independent set has size at most 2. Additional inequalities, such as odd-cycle inequalities, are still needed.

Exercise 55 (TSP on four cities: full model and LP relaxation). Write the symmetric four-city TSP, enumerate the tours, and examine the fractional point proposed in the question.

Solution. Use one binary variable $x_{ij} = x_{ji}$ for each undirected edge $1 \leq i < j \leq 4$. Let $\mathcal{S} = \{S \subsetneq \{1,2,3,4\} : 2 \leq |S| \leq 3\}$. The complete model is

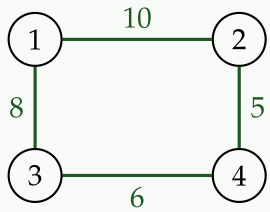
$$\begin{aligned}
\text{minimize} \quad & 10x_{12} + 8x_{13} + 9x_{14} + 7x_{23} + 5x_{24} + 6x_{34} \\
\text{subject to} \quad & x_{12} + x_{13} + x_{14} = 2, \\
& x_{12} + x_{23} + x_{24} = 2, \\
& x_{13} + x_{23} + x_{34} = 2, \\
& x_{14} + x_{24} + x_{34} = 2, \\
& \sum_{\substack{i<j \\ i,j \in S}} x_{ij} \leq |S| - 1 && (S \in \mathcal{S}), \\
& x_{ij} \in \{0, 1\} && (i < j).
\end{aligned}$$

There are six SECs for pairs and four for triples.

The trace's count $4!/2 = 12$ identifies an ordering only with its reverse, but not with its rotations. Those twelve representatives are

tour	cost	tour	cost
1234	32	1243	29
1324	29	1342	29
1423	29	1432	32
2134	29	2143	32
2314	29	2413	29
3124	29	3214	32

where every sequence returns to its first city. Many rows are rotations of the same cycle. Up to both rotation and reversal there are only $(4-1)!/2 = 3$ distinct undirected tours, with costs $32, 29, 29$. Thus the optimum is 29, for example the tour $1-2-4-3-1$.



The proposed fractional point sets all six undirected edges to $1/2$. Each city is then incident to only $3/2$, so it does *not* satisfy the degree equation requiring degree 2. Moreover, for an undirected four-city complete graph, degree equations and bounds already imply every size-2 and size-3 SEC; the requested counterexample cannot exist.

Subtours can still be illustrated with the directed formulation:

$$x_{12} = x_{21} = x_{34} = x_{43} = 1$$

gives two directed 2-cycles and satisfies all in/out degree equations, but violates the SEC for $S = \{1, 2\}$ because $x_{12} + x_{21} = 2 > |S| - 1 = 1$. A fractional version is obtained by setting $x_{12} = x_{21} = x_{34} = x_{43} = 3/4$ and $x_{13} = x_{31} = x_{24} = x_{42} = 1/4$; it also satisfies every directed degree equation and violates the same SEC.

The Simplex Method

Sign convention. The chapter defines reduced costs as $\hat{c}_j = c_j - c_B^\top B^{-1} A_j$, so a maximisation tableau is optimal when $\hat{c}_j \le 0$. Several exercise tableaux instead display $z_j - c_j = -\hat{c}_j$ and explicitly request the most-negative entry. For those tableaux we follow the displayed convention: a negative bottom-row entry identifies an improving variable, and optimality means that every bottom-row entry is nonnegative.

Exercise 1 (Converting to standard form). Convert the three-constraint maximisation problem to standard form, identify an initial basis, and give its BFS.

Solution. Add one slack variable to each inequality:

$$\begin{aligned} \text{maximize} \quad & 3x_1 + 5x_2 \\ \text{subject to} \quad & x_1 + s_1 = 4, \\ & 2x_2 + s_2 = 12, \\ & 3x_1 + 5x_2 + s_3 = 25, \\ & x_1, x_2, s_1, s_2, s_3 \ge 0. \end{aligned}$$

The slack columns form the identity matrix, so the natural basis is $B = \{s_1, s_2, s_3\}$. Setting the nonbasic variables $x_1 = x_2 = 0$ gives

$$(x_1, x_2, s_1, s_2, s_3) = (0, 0, 4, 12, 25), \qquad z = 0.$$

All basic variables are nonnegative, hence this is a BFS.

Exercise 2 (Standard form with equality and $\ge$ constraints). Convert the minimisation problem with one equality and one greater-than constraint to simplex-ready standard form.

Solution. Negate the objective, subtract a surplus variable from the $\ge$

constraint, and introduce artificials to obtain an initial identity basis:

$$\begin{aligned} \text{maximize} \quad & -4x_1 - x_2 \\ \text{subject to} \quad & x_1 + x_2 + a_1 = 5, \\ & 2x_1 - x_2 - s_2 + a_2 = 3, \\ & x_1, x_2, s_2, a_1, a_2 \geq 0. \end{aligned}$$

There are five variables: two original variables, one surplus variable s_2, and two artificial variables a_1, a_2. Strictly speaking, the algebraic standard form before constructing a starting basis has only x_1, x_2, s_2; the artificials are auxiliary simplex variables and must eventually be driven to zero.

Exercise 3 (Identifying a basis). For a 3×6 equality system, test the proposed basis $\{x_1, x_3, x_5\}$, diagnose degeneracy, and bound the number of BFSs.

Solution.

1. The three corresponding columns must be linearly independent: the 3×3 basis matrix must satisfy $\det B \neq 0$.
2. The basic solution is $(x_1, x_3, x_5) = (2, 0, 7)$. It is feasible but degenerate because the basic variable x_3 equals zero.
3. There are at most

$$\binom{6}{3} = 20$$

bases, hence at most 20 BFSs. The actual number may be smaller because some submatrices are singular, some basic solutions are infeasible, and different degenerate bases may represent the same BFS.

Exercise 4 (Unique determination of basic variables). Prove that a basis and the right-hand side uniquely determine the basic variables.

Solution. Partition the equality system according to a basis:

$$Bx_B + Nx_N = b.$$

At the associated basic solution, $x_N = 0$, so

$$Bx_B = b.$$

Because a basis matrix is invertible, multiplication by B^{-1} gives

$$x_B = B^{-1}b.$$

This vector is unique: if both u and v satisfied $Bu = Bv = b$, then $B(u - v) = 0$, and invertibility would imply $u - v = 0$. Feasibility is the additional condition $B^{-1}b \geq 0$.

Exercise 5 (Reading a BFS from a tableau). Read the basis, BFS, objective value, and optimality status from the given five-variable tableau.

Solution. The row labels identify

$$B = \{s_1, x_1, s_3\}.$$

Setting the nonbasic variables $x_2 = s_2 = 0$ and reading the right-hand side gives

$$(x_1, x_2, s_1, s_2, s_3) = (8, 0, 6, 0, 2).$$

The bottom-right entry gives the current objective value $z = 28$.

This exercise's bottom row uses the $z_j - c_j$ convention. Since the nonbasic variable x_2 has entry $-3 < 0$, the tableau is not optimal; x_2 is an improving entering candidate. Under the chapter's $c_j - z_j$ convention the same reduced cost would be $+3$.

Exercise 6 (Optimality check). Check two tableaux using the most-negative bottom-row rule.

Solution. The displayed rows use $z_j - c_j$, so optimality requires every nonbasic entry to be nonnegative.

(a) The entries -5 and -4 are negative, so the BFS is not optimal. The most-negative rule selects x_1.

(b) Every bottom-row entry is nonnegative. The current BFS is optimal, with $(x_1, x_2) = (4, 6)$ and objective value 36.

Exercise 7 (Ratio test — entering column given). With x_2 entering, apply the ratio test and identify the new basis.

Solution. Only positive entries in the entering column constrain the step:

$$\frac{12}{3} = 4, \qquad \frac{8}{1} = 8, \qquad \frac{10}{2} = 5.$$

The minimum is 4, so s_1 leaves. The pivot element is the 3 in row s_1, column x_2, and the updated basis is

$$\{x_2, s_2, s_3\}.$$

Geometrically, x_2 increases from zero to four; at that instant the first slack reaches zero before either of the others.

Exercise 8 (Full simplex pivot). Perform the pivot from the preceding tableau and read the new BFS.

Solution. Divide the pivot row by three:

$$R_1 \leftarrow \frac{1}{3}R_1 = (\tfrac{1}{3}, 1, \tfrac{1}{3}, 0, 0 \mid 4).$$

Then use $R_2 \leftarrow R_2 - R_1$, $R_3 \leftarrow R_3 - 2R_1$, and $R_z \leftarrow R_z + 6R_1$. The result is

	x_1	x_2	s_1	s_2	s_3	rhs
x_2	$\frac{1}{3}$	1	$\frac{1}{3}$	0	0	4
s_2	$\frac{5}{3}$	0	$-\frac{1}{3}$	1	0	4
s_3	$-\frac{2}{3}$	0	$-\frac{2}{3}$	0	1	2
$z - c$	0	0	2	0	0	24

Thus

$$(x_1, x_2, s_1, s_2, s_3) = (0, 4, 0, 4, 2), \qquad z = 24.$$

All bottom-row entries are now nonnegative, so for the LP represented by this tableau the new BFS is already optimal.

Exercise 9 (Two simplex iterations — small LP). Apply two pivots to the two-variable LP and decide whether the result is optimal.

Solution. After adding slacks, the initial tableau is

	x_1	x_2	s_1	s_2	rhs
s_1	1	1	1	0	4
s_2	1	3	0	1	6
$z - c$	−2	−3	0	0	0

The most-negative entry selects x_2. The ratios are 4 and 2, so s_2 leaves:

	x_1	x_2	s_1	s_2	rhs
s_1	$\frac{2}{3}$	0	1	$-\frac{1}{3}$	2
x_2	$\frac{1}{3}$	1	0	$\frac{1}{3}$	2
$z - c$	−1	0	0	1	6

Now x_1 enters. The ratios are 3 and 6, so s_1 leaves:

	x_1	x_2	s_1	s_2	rhs
x_1	1	0	$\frac{3}{2}$	$-\frac{1}{2}$	3
x_2	0	1	$-\frac{1}{2}$	$\frac{1}{2}$	1
$z - c$	0	0	$\frac{3}{2}$	$\frac{1}{2}$	9

All bottom-row entries are nonnegative. After exactly two pivots the solution is optimal:

$$(x_1, x_2) = (3, 1), \qquad z^* = 9.$$

Exercise 10 (Three-variable LP — simplex from scratch). Solve the three-variable, three-resource LP by complete simplex iterations.

Solution. With slacks s_1, s_2, s_3, the initial tableau is

	x_1	x_2	x_3	s_1	s_2	s_3	rhs
s_1	6	4	2	1	0	0	240
s_2	3	2	5	0	1	0	270
s_3	5	6	5	0	0	1	420
$z-c$	-5	-4	-3	0	0	0	0

First x_1 enters and s_1 leaves:

	x_1	x_2	x_3	s_1	s_2	s_3	rhs
x_1	1	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	0	0	40
s_2	0	0	4	$-\frac{1}{2}$	1	0	150
s_3	0	$\frac{8}{3}$	$\frac{10}{3}$	$-\frac{5}{6}$	0	1	220
$z-c$	0	$-\frac{2}{3}$	$-\frac{4}{3}$	$\frac{5}{6}$	0	0	200

Next x_3 enters and s_2 leaves:

	x_1	x_2	x_3	s_1	s_2	s_3	rhs
x_1	1	$\frac{2}{3}$	0	$\frac{5}{24}$	$-\frac{1}{12}$	0	$\frac{55}{2}$
x_3	0	0	1	$-\frac{1}{8}$	$\frac{1}{4}$	0	$\frac{75}{2}$
s_3	0	$\frac{8}{3}$	0	$-\frac{5}{12}$	$-\frac{5}{6}$	1	95
$z-c$	0	$-\frac{2}{3}$	0	$\frac{2}{3}$	$\frac{1}{3}$	0	250

Finally x_2 enters and s_3 leaves:

	x_1	x_2	x_3	s_1	s_2	s_3	rhs
x_1	1	0	0	$\frac{5}{16}$	$\frac{1}{8}$	$-\frac{1}{4}$	$\frac{15}{4}$
x_3	0	0	1	$-\frac{1}{8}$	$\frac{1}{4}$	0	$\frac{75}{2}$
x_2	0	1	0	$-\frac{5}{32}$	$-\frac{5}{16}$	$\frac{3}{8}$	$\frac{285}{8}$
$z-c$	0	0	0	$\frac{9}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1095}{4}$

The last row is nonnegative, hence

$$(x_1, x_2, x_3) = \left(\frac{15}{4}, \frac{285}{8}, \frac{75}{2}\right), \qquad z^* = \frac{1095}{4} = 273.75.$$

Exercise 11 (Pivot arithmetic check). Derive the generic tableau update and verify entries from Exercise 8.

Solution. After dividing pivot row r by a_{rs}, its entry in column j is a_{rj}/a_{rs}. To eliminate the entering-column coefficient a_{is} from another row, subtract a_{is} times the normalised pivot row:

$$\bar{a}_{ij} = a_{ij} - a_{is}\frac{a_{rj}}{a_{rs}} = a_{ij} - \frac{a_{is}}{a_{rs}}a_{rj}.$$

For the pivot $a_{12} = 3$ in Exercise 8:

$$\bar{a}_{21} = 2 - \frac{1}{3}(1) = \frac{5}{3},$$
$$\bar{a}_{31} = 0 - \frac{2}{3}(1) = -\frac{2}{3},$$
$$\bar{a}_{33} = 0 - \frac{2}{3}(1) = -\frac{2}{3},$$
$$\bar{c}_{s_1} = 0 - \frac{-6}{3}(1) = 2.$$

These agree with the full tableau obtained there.

Exercise 12 (Detecting an unbounded LP). State the tableau criterion for unboundedness and apply it to the given tableau.

Solution. In the displayed $z - c$ convention, choose a column with a negative bottom-row entry. If every constraint-row entry in that column is nonpositive, no basic variable decreases as the entering variable increases; the ratio test has no leaving row, so the objective is unbounded above.

The most-negative entry is -5 in column s_2. Its constraint column is $(-2, -3)^\top \le 0$, so the LP is unbounded. The row labels contain a typo: the two identity columns show that the current basic variables are x_1, x_2. Setting $s_1 = 0$ and $s_2 = t \ge 0$ gives

$$(x_1, x_2, s_1, s_2) = (3, 5, 0, 0) + t(2, 3, 0, 1).$$

This ray remains nonnegative and feasible, while the tableau objective is

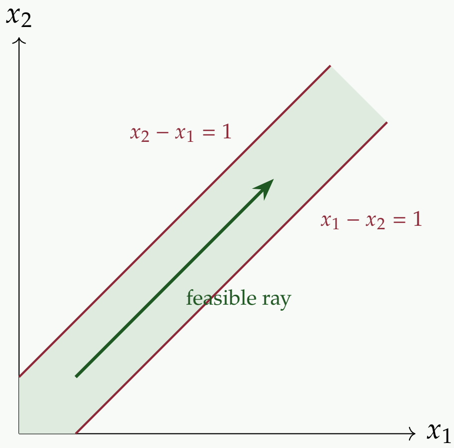
$$z(t) = 15 + 5t \longrightarrow +\infty.$$

Exercise 13 (Constructing an unbounded instance). Construct, draw, and verify a feasible two-variable LP that is unbounded.

Solution. One example is

$$\begin{aligned} \text{maximize} \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1 - x_2 \le 1, \\ & -x_1 + x_2 \le 1, \\ & x_1, x_2 \ge 0. \end{aligned}$$

The feasible region is the diagonal strip $|x_1 - x_2| \le 1$ in the first quadrant.



For every $t \geq 0$, $(x_1, x_2) = (t, t)$ is feasible and has objective $2t$. Since $2t \longrightarrow +\infty$, no finite optimum exists.

Exercise 14 (Degenerate BFS). Read the BFS, perform the ratio test with x_2 entering, and explain the connection between degeneracy and cycling.

Solution. The initial basis is $\{s_1, s_2, s_3\}$ and the BFS is

$$(x_1, x_2, s_1, s_2, s_3) = (0, 0, 0, 4, 6).$$

It is degenerate because the basic variable s_1 equals zero.

For x_2 entering, the ratios are

$$\frac{0}{1} = 0, \qquad \frac{4}{2} = 2, \qquad \frac{6}{1} = 6.$$

Thus s_1 leaves and the step length is zero. The basis changes, but x_2 enters with value zero and the geometric point and objective do not move.

At a degenerate vertex several bases can represent the same point. If a pivot rule repeatedly exchanges zero-valued basic variables, it may return to an earlier basis without objective improvement. That is cycling; an anti-cycling rule such as Bland's prevents it.

Exercise 15 (True/false: degeneracy and reduced costs). Classify four statements concerning zero reduced costs, degeneracy, cycling, and Bland's rule.

Solution.

1. **False.** A zero reduced cost concerns the objective, not the values of basic variables. For $\max\{x_1 + x_2 : x_1 + x_2 \leq 1,\ x \geq 0\}$, the nonbasic variable x_2 has zero reduced cost at $(1, 0)$, while the basic value is positive.
2. **False.** Degeneracy only means that *some* possible pivot may have zero step. For example, at the origin of $x_1 \leq 0,\ x_2 \leq 1$, the slack of the first constraint is zero, but letting x_2 enter moves to $(0, 1)$ and improves $\max x_2$.
3. **True**, if "non-degenerate LP" means every BFS is non-degenerate. Every improving pivot then has positive step length and strictly improves the objective, so an earlier basis cannot recur.

4. **True.** Bland's smallest-index rules for both entering and leaving variables guarantee finite termination.

Exercise 16 (Bland's rule — avoiding cycling). State Bland's rule and apply it to Beale's cycling instance.

Solution. Number the variables

$$x_1, x_2, x_3, x_4, s_1, s_2, s_3.$$

Bland's rule selects the smallest-index nonbasic variable with an improving reduced cost. In a ratio-test tie it removes the smallest-index basic variable among the tied rows.

Starting from the slack basis, exact pivot arithmetic gives

pivot	enter	leave	new basis	z
0			$\{s_1, s_2, s_3\}$	0
1	x_1	s_1	$\{x_1, s_2, s_3\}$	0
2	x_3	s_2	$\{x_1, x_3, s_3\}$	0
3	s_1	s_3	$\{x_1, x_3, s_1\}$	77/100

The first two pivots are degenerate. The third produces

$$x_1 = 1, \qquad x_3 = 1, \qquad s_1 = \frac{3}{4},$$

with all other variables zero. Its objective is $3/4 + 1/50 = 77/100$, and the reduced costs satisfy the optimality condition. Thus Bland's rule terminates in three pivots, within the claimed bound of four.

Exercise 17 (Bland's rule — small degenerate example). Construct a 2×4 tableau with a degenerate pivot and compare Dantzig's and Bland's entering choices.

Solution. Consider

	x_1	x_2	s_1	s_2	rhs
s_1	1	1	1	0	0
s_2	1	2	0	1	2
$z - c$	−2	−3	0	0	0

The BFS $(x_1, x_2, s_1, s_2) = (0, 0, 0, 2)$ is degenerate. Dantzig's most-negative rule chooses x_2; the ratios are

$$0/1 = 0, \qquad 2/2 = 1,$$

so s_1 leaves with step zero. The pivot changes the basis but not the point or objective.

At the same starting tableau, Bland's rule chooses the smallest-index improving variable, namely x_1. This pivot is also degenerate because the first ratio is zero, but the deterministic choice prevents an arbitrary sequence of exchanges and is part of the anti-cycling guarantee.

Exercise 18 (Vertices and basic feasible solutions). List the vertices of the bounded two-variable region, identify bases, and find the first simplex move for objective $x_1 + 2x_2$.

Solution. Introduce slacks in the order

$$x_1 + x_2 + s_1 = 6, \qquad x_1 + s_2 = 4, \qquad x_2 + s_3 = 4.$$

The vertices and one corresponding basis at each are

(x_1, x_2)	tight geometric constraints	basis
$(0,0)$	$x_1 = 0,\ x_2 = 0$	$\{s_1, s_2, s_3\}$
$(4,0)$	$x_1 = 4,\ x_2 = 0$	$\{x_1, s_1, s_3\}$
$(4,2)$	$x_1 = 4,\ x_1 + x_2 = 6$	$\{x_1, x_2, s_3\}$
$(2,4)$	$x_2 = 4,\ x_1 + x_2 = 6$	$\{x_1, x_2, s_2\}$
$(0,4)$	$x_1 = 0,\ x_2 = 4$	$\{x_2, s_1, s_2\}$.

At the origin the objective coefficients favour x_2 because $2 > 1$. Increasing x_2, the bounds are $x_2 \le 6$ and $x_2 \le 4$, so s_3 leaves and the first visited vertex is $(0, 4)$.

Exercise 19 (Tracing the simplex path). Sketch the quadrilateral, solve the LP graphically, and list the simplex path from the origin.

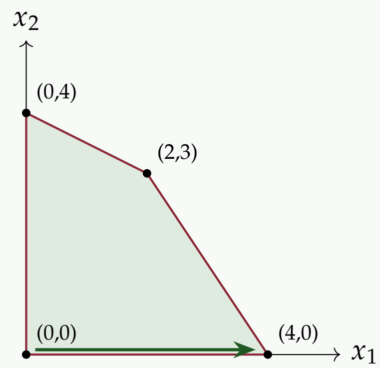
Solution. The two lines meet at

$$x_1 + 2x_2 = 8, \qquad 3x_1 + 2x_2 = 12 \quad \Longrightarrow \quad (x_1, x_2) = (2, 3).$$

The vertices and objective values for $3x_1 + x_2$ are

(x_1, x_2)	$(0,0)$	$(4,0)$	$(2,3)$	$(0,4)$
z	0	12	9	4.

Hence $(4, 0)$ is optimal.



With slacks s_1, s_2, the path is

basis	(x_1, x_2)	z
$\{s_1, s_2\}$	$(0,0)$	0
$\{s_1, x_1\}$	$(4,0)$	12.

Indeed, x_1 has the most favourable initial reduced cost; the second constraint gives the minimum ratio $12/3 = 4$. After that pivot all reduced costs satisfy optimality, so the method stops after one move.

Exercise 20 (Adjacent bases and edges). Explain the basis interpretation of adjacent vertices and illustrate it on the preceding LP.

Solution. At a nondegenerate BFS, fixing the nonbasic variables to zero identifies the active boundaries at a vertex. Replacing one basic column by one nonbasic column releases one active boundary and activates another. Exactly one degree of freedom is available during that transition, so the feasible points swept out form an edge joining adjacent vertices.

In Exercise 19, the origin has basis $\{s_1, s_2\}$ and $(4, 0)$ has basis $\{s_1, x_1\}$. The bases differ only by exchanging s_2 and x_1, and the corresponding geometric edge is the segment $\{(t, 0) : 0 \le t \le 4\}$. At a degenerate vertex, two adjacent bases may represent the same geometric point, producing a zero-length move.

Exercise 21 (Reading the optimal tableau). Read the primal optimum and identify the tight constraints from the given final tableau.

Solution. The basic variables are x_2, s_2, x_1, with right-hand sides $3, 2, 5$. Thus

$$(x_1^*, x_2^*) = (5, 3), \qquad (s_1^*, s_2^*, s_3^*) = (0, 2, 0), \qquad z^* = 40.$$

A zero slack means the corresponding original inequality is tight. Therefore constraints 1 and 3 are tight, while constraint 2 has two units of unused capacity.

Exercise 22 (Dual variables from the optimal tableau). Read the dual multipliers from the final tableau and test the proposed strong-duality identity.

Solution. For a $\le$ maximisation problem,

$$y^\top = c_B^\top B^{-1}.$$

If a final tableau displays $z_j - c_j$, the entries under the original slack columns equal y_j because each slack column is e_j. The given row therefore suggests

$$y = (2, 0, 1).$$

The numerical verification requested in the exercise is impossible:

$$y^\top b = 2(8) + 0(4) + 1(10) = 26 \neq 40.$$

There is a second contradiction. The stated primal vector and objective coefficients give

$$c^\top x^* = 3(5) + 4(3) = 27 \neq 40.$$

Thus the tableau, b, c, and stated objective value cannot all belong to the same LP. The formula and tableau-reading rule are correct, but the supplied numerical data are inconsistent.

Exercise 23 (Reconstructing the original LP). Use the slack columns of the preceding tableau to reconstruct the constraint matrix and test whether a consistent objective exists.

Solution. The basis order is (x_2, s_2, x_1). The three slack columns give

$$B^{-1} = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{4} \\ -1 & 1 & \frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{3}{4} \end{pmatrix}, \qquad B = \begin{pmatrix} 3 & 0 & 1 \\ 2 & 1 & 0 \\ 2 & 0 & 2 \end{pmatrix}.$$

Because the columns of B are respectively A_2, e_2, A_1,

$$A_1 = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 3 \\ 2 \\ 2 \end{pmatrix}.$$

Moreover,

$$b = B \begin{pmatrix} 3 \\ 2 \\ 5 \end{pmatrix} = \begin{pmatrix} 14 \\ 8 \\ 16 \end{pmatrix}.$$

Hence the uniquely reconstructed constraints are

$$\begin{aligned} x_1 + 3x_2 &\leq 14, \\ 2x_2 &\leq 8, \\ 2x_1 + 2x_2 &\leq 16, \qquad x \geq 0. \end{aligned}$$

The bottom row, interpreted as $z - c$, gives $y = (2, 0, 1)$ and therefore would imply

$$c_1 = y^\top A_1 = 4, \qquad c_2 = y^\top A_2 = 8.$$

But at $(5, 3)$ this objective equals 44, not the displayed 40. Consequently no full original LP is consistent with every entry of the given tableau. Ignoring the incorrect bottom-right value, the objective reconstructed from the other entries is $\max 4x_1 + 8x_2$.

Exercise 24 (Phase I formulation). Set up Phase I for one equality and one less-than constraint, and state the feasibility test.

Solution. The equality needs an artificial variable, while the inequality has a natural slack:

$$\begin{aligned} x_1 + x_2 + a_1 &= 4, \\ x_1 - x_2 + s_2 &= 2, \\ x_1, x_2, s_2, a_1 &\geq 0. \end{aligned}$$

Phase I solves

$$\min a_1 \qquad \text{or equivalently} \qquad \max(-a_1),$$

starting from basis $\{a_1, s_2\}$ and BFS $(x_1, x_2, s_2, a_1) = (0, 0, 2, 4)$.

If the Phase I optimum is zero, the artificial can be removed and the remaining basis initializes Phase II. If the optimum is strictly positive, no point with $a_1 = 0$ exists, so the original LP is infeasible.

Exercise 25 (Phase I — full execution). Execute Phase I for the two equality constraints and then optimise the original objective.

Solution. Introduce a_1, a_2 and maximise $-a_1 - a_2$. Using $z - c$ in the last row, the canonical initial tableau is

	x_1	x_2	a_1	a_2	rhs
a_1	1	2	1	0	6
a_2	2	1	0	1	8
$z-c$	-3	-3	0	0	-14

Choose x_1 first; the ratios are 6 and 4, so a_2 leaves:

	x_1	x_2	a_1	a_2	rhs
a_1	0	$\frac{3}{2}$	1	$-\frac{1}{2}$	2
x_1	1	$\frac{1}{2}$	0	$\frac{1}{2}$	4
$z-c$	0	$-\frac{3}{2}$	0	$\frac{3}{2}$	-2

Now x_2 enters and a_1 leaves:

	x_1	x_2	a_1	a_2	rhs
x_2	0	1	$\frac{2}{3}$	$-\frac{1}{3}$	$\frac{4}{3}$
x_1	1	0	$-\frac{1}{3}$	$\frac{2}{3}$	$\frac{10}{3}$
$z-c$	0	0	1	1	0

The Phase I optimum is zero and both artificials are nonbasic. After deleting their columns, Phase II has no nonbasic original variable: the two equalities uniquely determine

$$x_1 = \frac{10}{3}, \qquad x_2 = \frac{4}{3}.$$

Therefore this feasible point is automatically optimal for the original objective, with

$$z^* = 3\frac{10}{3} + 2\frac{4}{3} = \frac{38}{3}.$$

Exercise 26 (Phase I detects infeasibility). Apply the auxiliary objective to two contradictory equality constraints.

Solution. With $t = x_1 + x_2$, the artificial equations are

$$t + a_1 = 5, \qquad t + a_2 = 7, \qquad t, a_1, a_2 \geq 0.$$

Thus $0 \leq t \leq 5$ and

$$a_1 + a_2 = (5 - t) + (7 - t) = 12 - 2t.$$

This is minimised at $t = 5$, giving

$$\min(a_1 + a_2) = 2 > 0.$$

At least two units of artificial violation remain because the same quantity cannot equal both 5 and 7. Hence the original LP is infeasible.

Exercise 27 (Artificial variable remains in basis). Explain how to remove a zero-valued artificial variable or recognize a redundant row.

Solution. Suppose a basic artificial has value zero after Phase I.

1. If its row contains a nonzero coefficient in a non-artificial column, pivot on that coefficient. This is a degenerate pivot: the artificial leaves, while the BFS remains unchanged.
2. If every non-artificial coefficient in the row is zero, the row has become $a_i = 0$. It contains no information about the original variables and may be deleted as a redundant equality.

For the first case, consider

	x_1	x_2	a_2	rhs
x_1	1	1	0	1
a_2	0	1	1	0

with basis $\{x_1, a_2\}$. The artificial is basic at zero. Pivoting on the second row's x_2 coefficient gives basis $\{x_1, x_2\}$ and removes a_2 without changing $(x_1, x_2) = (1, 0)$.

Exercise 28 (Big-M formulation). Write the Big-M model and its initial tableau for the Phase I setup of Exercise 24.

Solution. Penalise the artificial in the original objective:

$$\begin{aligned} \text{maximize} \quad & 2x_1 + x_2 - Ma_1 \\ \text{subject to} \quad & x_1 + x_2 + a_1 = 4, \\ & x_1 - x_2 + s_2 = 2, \\ & x_1, x_2, s_2, a_1 \geq 0. \end{aligned}$$

With basis $\{a_1, s_2\}$, eliminate the basic artificial's coefficient from the objective row. In $z - c$ convention:

	x_1	x_2	s_2	a_1	rhs
a_1	1	1	0	1	4
s_2	1	−1	1	0	2
$z - c$	$-(M+2)$	$-(M+1)$	0	0	$-4M$

For a sufficiently large symbolic M, an artificial variable remaining positive at the optimum signals infeasibility.

Exercise 29 (Big-M — full solve). Solve the equality-constrained LP with one additional inequality using the symbolic Big-M method.

Solution. Add artificial a_1 and slack s_2:

$$x_1 + x_2 + a_1 = 4, \qquad x_1 + 2x_2 + s_2 = 6.$$

The canonical initial tableau is

	x_1	x_2	s_2	a_1	rhs
a_1	1	1	0	1	4
s_2	1	2	1	0	6
$z-c$	$-(M+2)$	$-(M+3)$	0	0	$-4M$

For large M, x_2 enters. The second row wins the ratio test:

	x_1	x_2	s_2	a_1	rhs
a_1	$\frac{1}{2}$	0	$-\frac{1}{2}$	1	1
x_2	$\frac{1}{2}$	1	$\frac{1}{2}$	0	3
$z-c$	$-\frac{M+1}{2}$	0	$\frac{M+3}{2}$	0	$-M+9$

Then x_1 enters and a_1 leaves:

	x_1	x_2	s_2	a_1	rhs
x_1	1	0	-1	2	2
x_2	0	1	1	-1	2
$z-c$	0	0	1	$M+1$	10

The artificial is nonbasic at zero and all ordinary reduced-cost entries are nonnegative. Therefore

$$(x_1, x_2) = (2, 2), \qquad z^* = 2(2) + 3(2) = 10.$$

Exercise 30 (Comparing Phase I and Big-M). Contrast the two methods theoretically and computationally.

Solution. Theoretical differences:

1. Two-Phase separates feasibility from optimisation; Big-M combines them in one penalised objective.
2. Phase I has an exact, scale-free feasibility certificate $\min \sum a_i = 0$. Big-M requires a penalty known to dominate every possible gain in the original objective.

Practical differences:

1. Two-Phase requires replacing and recanonicalising the objective between phases; Big-M keeps one tableau.
2. Symbolic M is cumbersome, while a large numerical M creates poor scaling, cancellation, and unreliable reduced-cost comparisons.

In floating-point arithmetic, an excessively large M can make small but meaningful coefficients disappear relative to M, whereas an insufficiently large M may make retaining an artificial variable look profitable. Both can lead to a wrong basis or an incorrect feasibility conclusion. Two-Phase is therefore the safer default.

Exercise 31 (Furniture workshop — Phase II continuation). Reproduce the two pivots for the workshop LP, then add $x_1 + x_2 \leq 4$ and re-optimise.

Solution. For the original two-constraint LP, the initial tableau is

	x_1	x_2	s_1	s_2	rhs
s_1	6	4	1	0	24
s_2	1	2	0	1	6
$z-c$	-5	-4	0	0	0

First x_1 enters and s_1 leaves:

	x_1	x_2	s_1	s_2	rhs
x_1	1	$\frac{2}{3}$	$\frac{1}{6}$	0	4
s_2	0	$\frac{4}{3}$	$-\frac{1}{6}$	1	2
$z-c$	0	$-\frac{2}{3}$	$\frac{5}{6}$	0	20

Then x_2 enters and s_2 leaves:

	x_1	x_2	s_1	s_2	rhs
x_1	1	0	$\frac{1}{4}$	$-\frac{1}{2}$	3
x_2	0	1	$-\frac{1}{8}$	$\frac{3}{4}$	$\frac{3}{2}$
$z-c$	0	0	$\frac{3}{4}$	$\frac{1}{2}$	21

Thus $(3, 3/2)$ is optimal with value 21.

After adding $x_1 + x_2 + s_3 = 4$, start with the three slack variables. Dantzig's rule again lets x_1 enter. The first and third constraints tie at ratio 4; choosing s_1 to leave gives

	x_1	x_2	s_1	s_2	s_3	rhs
x_1	1	$\frac{2}{3}$	$\frac{1}{6}$	0	0	4
s_2	0	$\frac{4}{3}$	$-\frac{1}{6}$	1	0	2
s_3	0	$\frac{1}{3}$	$-\frac{1}{6}$	0	1	0
$z-c$	0	$-\frac{2}{3}$	$\frac{5}{6}$	0	0	20

x_2 is still improving, but the third row gives ratio zero. The pivot is degenerate: x_2 replaces s_3 without moving from $(4, 0)$. The resulting bottom row is nonnegative, so

$$(x_1, x_2) = (4, 0), \qquad z^* = 20.$$

The original optimum is cut off because $3 + 3/2 > 4$.

Exercise 32 (Furniture workshop — sensitivity). Find the allowable increase of the x_1 profit and the allowable decrease of the first right-hand side.

Solution. At the optimal basis (x_1, x_2),

$$B = \begin{pmatrix} 6 & 4 \\ 1 & 2 \end{pmatrix}, \qquad B^{-1} = \begin{pmatrix} \frac{1}{4} & -\frac{1}{2} \\ -\frac{1}{8} & \frac{3}{4} \end{pmatrix}.$$

Replace $c_1 = 5$ by $5 + \Delta$. The dual multipliers become

$$y^\top = (5 + \Delta, 4)B^{-1} = \left(\frac{3}{4} + \frac{\Delta}{4}, \frac{1}{2} - \frac{\Delta}{2}\right).$$

The slack reduced costs are $-y_1, -y_2$, so optimality requires $y \geq 0$. Hence

$$-3 \leq \Delta \leq 1.$$

In particular, the coefficient of x_1 may increase by at most 1, from 5 to 6.

If b_1 decreases by $d \geq 0$, then

$$x_B = \begin{pmatrix} 3 \\ 3/2 \end{pmatrix} - d \begin{pmatrix} 1/4 \\ -1/8 \end{pmatrix} = \begin{pmatrix} 3 - d/4 \\ 3/2 + d/8 \end{pmatrix}.$$

Feasibility requires $d \leq 12$. Thus b_1 may fall from 24 to 12 before the current basis ceases to be feasible.

Exercise 33 (Reduced cost formula derivation). Derive the reduced costs, reduced objective, and simplex optimality condition from a feasible basis.

Solution. Partition $A = [B\ N]$ and $x = (x_B, x_N)$. From

$$Bx_B + Nx_N = b$$

we obtain

$$x_B = B^{-1}b - B^{-1}Nx_N.$$

Substituting into the objective gives

$$\begin{aligned} z &= c_B^\top x_B + c_N^\top x_N \\ &= c_B^\top B^{-1}b + \left(c_N^\top - c_B^\top B^{-1}N\right) x_N. \end{aligned}$$

Therefore

$$\bar{c}_N = c_N - N^\top B^{-\top} c_B \quad \text{or rowwise} \quad \bar{c}_N^\top = c_N^\top - c_B^\top B^{-1}N,$$

and

$$z = c_B^\top B^{-1}b + \bar{c}_N^\top x_N.$$

For every feasible point $x_N \geq 0$. If $\bar{c}_N \leq 0$ componentwise, then $\bar{c}_N^\top x_N \leq 0$, so no feasible solution exceeds the current BFS value $c_B^\top B^{-1}b$. This proves sufficiency.

Exercise 34 (Optimality condition — sufficiency and necessity). Prove the reduced-cost test and show how a zero reduced cost can coexist with a unique optimum.

Solution. Sufficiency follows directly from

$$z = \bar{z} + \sum_{j \in N} \bar{c}_j x_j.$$

If $x_j \geq 0$ and all $\bar{c}_j \leq 0$, then $z \leq \bar{z}$ for every feasible point.

For necessity at a nondegenerate optimal BFS, suppose instead that a nonbasic $\bar{c}_s > 0$. If the entering column has no positive entry, the LP is unbounded in direction s, contradicting optimality. Otherwise the ratio test defines a leaving variable. Nondegeneracy makes every basic value positive, so the step $\theta^* > 0$ and

$$z_{\text{new}} - \bar{z} = \bar{c}_s \theta^* > 0,$$

again contradicting optimality. Hence every nonbasic reduced cost must be nonpositive.

Zero reduced cost does not always yield another geometric optimum when the BFS is degenerate. Consider

$$\max x_1 \quad \text{s.t.} \quad x_1 \le 1, \quad x_1 + x_2 \le 1, \quad x \ge 0.$$

The unique optimum is $(1, 0)$. In basis $\{x_1, s_2\}$, the basic slack $s_2 = 0$, and nonbasic x_2 has reduced cost zero. Attempting to enter x_2 has step zero, so it changes only the basis and produces no second point.

Exercise 35 (Multiple optimal solutions). Pivot on the zero-reduced-cost variable and describe the complete set of optima.

Solution. The current BFS is

$$(x_1, x_2, s_1, s_2) = (6, 0, 0, 2), \qquad z = 18.$$

The nonbasic variable x_2 has zero reduced cost. Its ratio test gives

$$6/2 = 3, \qquad 2/1 = 2,$$

so s_2 leaves. Contrary to the wording in the question, this pivot is *not* degenerate: the step is 2. The new tableau is

	x_1	x_2	s_1	s_2	rhs
x_1	1	0	3	−2	2
x_2	0	1	−1	1	2
$z - c$	0	0	3	0	18

and gives the second optimal BFS $(x_1, x_2) = (2, 2)$.

Every point on the joining edge is optimal:

$$(x_1, x_2) = (1 - \lambda)(6, 0) + \lambda(2, 2) = (6 - 4\lambda, 2\lambda), \qquad 0 \le \lambda \le 1.$$

Exercise 36 (Worst-case complexity of simplex). Write the three-dimensional Klee–Minty instance and interpret its exponential path.

Solution. The version used in the chapter is

$$\begin{aligned} \text{maximize} \quad & 4x_1 + 2x_2 + x_3 \\ \text{subject to} \quad & x_1 \le 5, \\ & 4x_1 + x_2 \le 25, \\ & 8x_1 + 4x_2 + x_3 \le 125, \\ & x_1, x_2, x_3 \ge 0. \end{aligned}$$

It is a skewed three-dimensional cube and therefore has $2^3 = 8$ vertices. With Dantzig's rule, the simplex path visits all eight, requiring $2^3 - 1 = 7$

pivots.

This is a deliberately adversarial construction, not a typical instance. Real LPs are often sparse and structured, and empirical or smoothed analyses show far shorter paths. An exponential worst-case bound therefore coexists with excellent average practical performance.

Exercise 37 (Polynomial vs. exponential algorithms). Compare the complexity and practical behaviour of simplex, ellipsoid, and interior-point methods.

Solution.

1. Simplex has exponential worst-case pivot complexity ($2^n - 1$ on Klee–Minty families). The ellipsoid method and interior-point methods have polynomial bit-complexity bounds.
2. Simplex is often preferred because it exploits sparsity well, supports warm starts, returns a useful optimal basis, and is extremely fast on ordinary instances despite the worst case. Its basis also makes sensitivity analysis and economic interpretation immediate.
3. LP feasibility and optimisation are in the complexity class **P**. Leonid Khachiyan first proved polynomial-time solvability in 1979 using the ellipsoid method.

Exercise 38 (Diet problem — simplex setup). Formulate the diet LP, convert it to maximisation standard form, and construct the Phase I tableau.

Solution. Let $A, B \geq 0$ be daily food units. The model is

$$\begin{aligned} \text{minimize} \quad & 2A + \frac{3}{2}B \\ \text{subject to} \quad & 400A + 300B \geq 2000, \\ & 20A + 10B \geq 50. \end{aligned}$$

Negating the objective and introducing surplus and artificial variables gives

$$\begin{aligned} \text{maximize} \quad & -2A - \frac{3}{2}B \\ \text{subject to} \quad & 400A + 300B - s_1 + a_1 = 2000, \\ & 20A + 10B - s_2 + a_2 = 50, \\ & A, B, s_1, s_2, a_1, a_2 \geq 0. \end{aligned}$$

Phase I maximises $-a_1 - a_2$. With basis $\{a_1, a_2\}$, its canonical $z - c$ tableau is

	A	B	s_1	s_2	a_1	a_2	rhs
a_1	400	300	−1	0	1	0	2000
a_2	20	10	0	−1	0	1	50
$z - c$	−420	−310	1	1	0	0	−2050

Phase I drives the artificials to zero; then their columns are deleted and the original negated cost row starts Phase II.

Exercise 39 (Transport LP — one simplex pivot). Choose the entering and leaving variables, pivot once, and report the new objective.

Solution. The most-negative bottom-row entry is −3, so x_{12} enters. Its column is $(1,0)^\top$: only the first row participates in the ratio test, and s_1 leaves at step 50. The pivot element is already one. Eliminating the objective coefficient gives

	x_{11}	x_{12}	x_{21}	x_{22}	s_1	rhs
x_{12}	1	1	0	0	1	50
x_{21}	0	0	1	1	0	30
$z-c$	1	0	0	−1	3	240

The new objective value is 240. The remaining negative entry under x_{22} shows that another improving pivot is possible.

Exercise 40 (Production planning — three products). Set up the initial tableau, perform the first pivot, and identify the next entering variable.

Solution. Using x_1, x_2, x_3 for P, Q, R:

	x_1	x_2	x_3	s_1	s_2	s_3	rhs
s_1	2	1	1	1	0	0	14
s_2	1	2	1	0	1	0	14
s_3	1	1	2	0	0	1	14
$z-c$	−8	−5	−4	0	0	0	0

x_1 enters. The ratios are $7, 14, 14$, so s_1 leaves:

	x_1	x_2	x_3	s_1	s_2	s_3	rhs
x_1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	7
s_2	0	$\frac{3}{2}$	$\frac{1}{2}$	$-\frac{1}{2}$	1	0	7
s_3	0	$\frac{1}{2}$	$\frac{3}{2}$	$-\frac{1}{2}$	0	1	7
$z-c$	0	−1	0	4	0	0	56

The solution is not yet optimal because x_2 has bottom-row entry −1. Thus x_2 is the next entering variable.

Exercise 41 (Maximisation with four constraints). Solve the stated three-constraint production LP and list the bases visited.

Solution. The title says four constraints, but the displayed LP has three. Adding slacks s_1, s_2, s_3 and applying the most-negative rule gives

iteration	basis	(x_1, x_2, x_3)	z
0	$\{s_1, s_2, s_3\}$	$(0,0,0)$	0
1	$\{s_1, s_2, x_2\}$	$(0,30,0)$	60
2	$\{x_1, s_2, x_2\}$	$(10,30,0)$	70.

In the first pivot, x_2 enters and s_3 leaves because $30/1 < 40/1$. The resulting BFS has $s_1 = 10, s_2 = 60$. Then x_1 enters and s_1 leaves at step 10. At

$$(x_1, x_2, x_3) = (10, 30, 0)$$

the slacks are $(0, 40, 0)$ and all reduced costs satisfy optimality. Hence $z^* = 10 + 2(30) = 70$.

Exercise 42 (Entering variable tie-breaking). Compare Bland's and largest-coefficient rules on the tied initial tableau.

Solution. Both x_1 and x_2 have bottom-row entry -4.

1. Bland's rule selects the smallest index, x_1.
2. The largest-objective-coefficient rule is also tied, and the prescribed smallest-index tie-break again selects x_1.
3. Since the two rules make the same choice, their entire subsequent path is identical. First x_1 enters and s_1 leaves; then x_2 enters and s_2 leaves. The optimum is

$$(x_1, x_2) = \left(\frac{8}{5}, \frac{18}{5}\right), \qquad z^* = \frac{104}{5},$$

after two pivots.

Exercise 43 (Ratio test tie — perturbation method). Break the tied ratio test lexicographically when x_1 enters.

Solution. Without perturbation, both ratios equal

$$\frac{4}{2} = 2.$$

Adding the *same* scalar ϵ to both right-hand sides would not break the tie. Lexicographic perturbation assigns distinct infinitesimals, for example

$$b_1 = 4 + \epsilon, \qquad b_2 = 4 + \epsilon^2, \qquad 0 < \epsilon \ll 1.$$

Then

$$\frac{4+\epsilon}{2} = 2 + \frac{\epsilon}{2}, \qquad \frac{4+\epsilon^2}{2} = 2 + \frac{\epsilon^2}{2}.$$

Since $\epsilon^2 < \epsilon$, row 2 has the smaller perturbed ratio, so s_2 leaves. Formal implementations compare coefficient vectors lexicographically rather than assigning a numerical tiny value.

Exercise 44 (Feasibility and the simplex method). Interpret a zero Phase I optimum with a basic zero artificial and analyse the requested examples.

Solution.

1. Yes. Phase I value zero means every artificial variable is zero, whether basic or nonbasic. Removing the artificial coordinates leaves an $x \geq 0$ satisfying the original equalities.
2. For

$$x_1 + x_2 = 1, \qquad 2x_1 + 2x_2 = 2,$$

the second equality is redundant. A zero artificial may remain basic in its transformed zero row; that row can be dropped.
3. The requested counterexample cannot exist as stated. If the artificial's

zero row contains a nonzero original-variable coefficient, it can be pivoted out. If it contains none, the transformed row is $0 = 0$, proving row dependence and hence redundancy. An inconsistent rank-deficient subsystem would instead force a *positive* Phase I optimum, contradicting the premise that the optimum is zero.

Exercise 45 (Sensitivity: objective coefficient ranging). Interpret the stated reduced costs and derive the generic effect of changing a basic objective coefficient.

Solution. Under the chapter convention $\bar{c}_j = c_j - c_B^\top B^{-1} A_j$, a maximisation basis is optimal when $\bar{c}_j \le 0$. Therefore

$$\bar{c}_2 = c_2 - 3 \le 0 \quad \Longleftrightarrow \quad c_2 \le 3.$$

There is no lower bound from this reduced cost alone.

However, the simultaneous statement $\bar{c}_{s_1} = 2$ contradicts the claim that the tableau is optimal. If the exercise intended the opposite $z - c$ convention, the inequalities reverse: $c_2 \ge 3$, and the entry 2 is compatible with optimality. Thus the numerical data are sign-ambiguous.

Generically, if x_1 is the rth basic variable and its coefficient increases by Δ, then every nonbasic reduced cost becomes

$$\bar{c}_j(\Delta) = \bar{c}_j - \Delta\,(B^{-1}A_j)_r.$$

The basis remains optimal exactly while

$$\bar{c}_j - \Delta\,(B^{-1}A_j)_r \le 0 \qquad \text{for every nonbasic } j$$

under the chapter convention.

Exercise 46 (Sensitivity: RHS ranging). Derive the allowable range for a change in one right-hand side and apply it to the supplied final tableau.

Solution. If b_1 changes by δ, then

$$x_B(\delta) = B^{-1}(b + \delta e_1) = x_B + \delta\, B^{-1} e_1.$$

The current basis remains feasible exactly when every component of this vector is nonnegative:

$$(x_B)_i + \delta (B^{-1})_{i1} \ge 0 \qquad (i = 1, \ldots, m).$$

The reduced costs do not depend on b, so a basis that remains feasible also remains optimal.

In Exercise 21, the basic order is (x_2, s_2, x_1) and the first slack column gives

$$B^{-1} e_1 = \begin{pmatrix} 1/2 \\ -1 \\ -1/2 \end{pmatrix}.$$

Thus

$$\begin{pmatrix} x_2 \\ s_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 5 \end{pmatrix} + \delta \begin{pmatrix} 1/2 \\ -1 \\ -1/2 \end{pmatrix}.$$

Feasibility requires

$$3 + \delta/2 \geq 0, \quad 2 - \delta \geq 0, \quad 5 - \delta/2 \geq 0,$$

so the same basis is feasible for

$$-6 \leq \delta \leq 2.$$

Exercise 47 (Cycling: explicit three-pivot cycle). Give an explicit cycling tableau, list the repeated bases, and explain why Bland's rule prevents the repetition.

Solution. The dimensions in the question are inconsistent with the standard six-pivot construction: Beale's example has three constraints, four structural variables, and three slacks, hence seven variables. Its initial tableau is

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	rhs
s_1	$\frac{1}{2}$	$-\frac{11}{2}$	$-\frac{5}{2}$	9	1	0	0	0
s_2	$\frac{1}{2}$	$-\frac{3}{2}$	$-\frac{1}{2}$	1	0	1	0	0
s_3	1	0	0	0	0	0	1	1
$z - c$	-10	57	9	24	0	0	0	0

Use Dantzig's most-negative entering rule and the ordinary first minimum-ratio row in ties. Every one of the following pivots has step zero:

pivot	enter	leave	basis
0			$\{s_1, s_2, s_3\}$
1	x_1	s_1	$\{x_1, s_2, s_3\}$
2	x_2	s_2	$\{x_1, x_2, s_3\}$
3	x_3	x_1	$\{x_3, x_2, s_3\}$
4	x_4	x_2	$\{x_3, x_4, s_3\}$
5	s_1	x_3	$\{s_1, x_4, s_3\}$
6	s_2	x_4	$\{s_1, s_2, s_3\}$.

The initial basis and objective value zero have returned. Repeating the same six choices completes a second identical round, so the algorithm cycles forever.

The trace's phrase "two rounds of three pivots" does not match this classical six-pivot cycle. More importantly, merely specifying the matrix dimensions cannot determine a cycle: the entering and tie-breaking rules are part of the construction. Bland's rule replaces the most-negative choice by the smallest-index improving variable and uses smallest-index leaving ties; its finite-termination theorem precludes the repeated sequence above.

Exercise 48 (Recap: simplex algorithm correctness). State the correctness ingredients, explain the cycling caveat, and prove strict improvement for a nondegenerate pivot.

Solution. The core facts are:

1. an improving pivot never decreases the objective;
2. only finitely many bases exist;
3. if all nonbasic reduced costs satisfy the optimality condition, the current BFS is globally optimal.

These facts alone are slightly incomplete as a termination proof. During cycling, none of them is false: the objective remains constant, the set of bases is finite, and no visited basis is optimal. What fails is the unstated claim that a basis cannot repeat. Bland's rule restores that missing property by guaranteeing that no cycle of bases can occur.

For a nondegenerate pivot with entering variable x_s, write the objective along the feasible edge as

$$z(\theta) = \bar{z} + \bar{c}_s \theta.$$

If $\bar{c}_s > 0$, the direction is improving. Nondegeneracy makes the minimum ratio $\theta^* > 0$, hence

$$z_{\text{new}} - \bar{z} = \bar{c}_s \theta^* > 0.$$

Thus every nondegenerate improving pivot strictly raises the objective.

Exercise 49 (Simplex on a minimisation LP). Negate the objective and solve the nonnegative-cost minimisation problem.

Solution. Negating the objective gives

$$\max \ -x_1 - 2x_2 - 3x_3$$

with the same three $\leq$ constraints. After adding slacks, the initial tableau is

	x_1	x_2	x_3	s_1	s_2	s_3	rhs
s_1	1	1	0	1	0	0	10
s_2	0	1	1	0	1	0	8
s_3	1	0	1	0	0	1	7
$z - c$	1	2	3	0	0	0	0

All $z - c$ entries are already nonnegative, so no pivot is performed. This is also obvious from the original problem: all variables and all cost coefficients are nonnegative, and $x = 0$ is feasible. Therefore

$$(x_1, x_2, x_3) = (0, 0, 0), \qquad \min(x_1 + 2x_2 + 3x_3) = 0.$$

The initial tableau is also the final tableau; there are no intermediate iterations.

Exercise 50 (Characterising all optimal bases). Identify zero reduced costs and parameterise the optimal face of the given final tableau.

Solution. The basis is $\{x_1, s_2, x_2\}$, so the nonbasic variables are s_1, s_3. In the displayed $z - c$ convention,

$$(z - c)_{s_1} = 0, \qquad (z - c)_{s_3} = 2.$$

Thus s_1 can enter without changing the objective, while moving s_3 would worsen it. The optimum is not unique.

Set $s_3 = 0$ and let $s_1 = t$. The tableau rows give

$$x_1 = 4 - 2t, \qquad s_2 = 3 + t, \qquad x_2 = 5 - t.$$

Nonnegativity requires $0 \le t \le 2$. Hence the optimal face in the original variable plane is the segment

$$(x_1, x_2) = (4 - 2t, 5 - t), \qquad 0 \le t \le 2,$$

joining the optimal BFSs $(4, 5)$ and $(0, 3)$.

Exercise 51 (Eta-factorisation and the basis update). Describe the eta matrix, verify the pivot of Exercise 8, and explain its computational role.

Solution. Let $d = B^{-1}A_s$ be the entering column and let row r leave. The elementary matrix E equals the identity except in column r, where

$$E_{rr} = \frac{1}{d_r}, \qquad E_{ir} = -\frac{d_i}{d_r} \quad (i \neq r).$$

Left multiplication performs exactly the pivot row operations, so $B'^{-1} = EB^{-1}$.

In Exercise 8 the initial basis is the identity, $d = (3, 1, 2)^\top$, and row 1 leaves. Therefore

$$E = \begin{pmatrix} \frac{1}{3} & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{2}{3} & 0 & 1 \end{pmatrix}, \qquad B'^{-1} = E.$$

Multiplying the original columns and b by E produces the pivoted rows, including the new right-hand side $(4, 4, 2)^\top$.

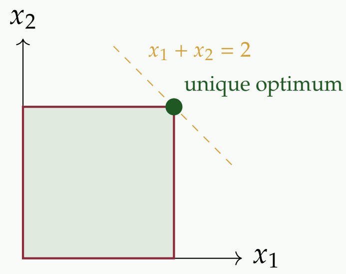
Large implementations avoid recomputing and reinverting B' from scratch. They store a sequence of sparse eta matrices and apply them to vectors, periodically refactorising for numerical stability. This substantially reduces time and memory.

Exercise 52 (Degenerate optimum). Construct and analyse a two-variable LP with a unique degenerate optimal BFS.

Solution. Consider

$$\begin{aligned} \text{maximize} \quad & x_1 + x_2 \\ \text{subject to} \quad & x_1 \le 1, \\ & x_2 \le 1, \\ & x_1 + x_2 \le 2, \\ & x_1, x_2 \ge 0. \end{aligned}$$

The unique optimum is $(1, 1)$. All three inequalities are tight there, although in two dimensions only two independent boundaries are needed; the third inequality is redundant but tight. This is the precise meaning needed here: a slack equals zero *if and only if* its constraint is binding, so the wording "zero slack but not binding" is literally impossible.



Choose basis $\{x_1, x_2, s_3\}$. Its tableau is

	x_1	x_2	s_1	s_2	s_3	rhs
x_1	1	0	1	0	0	1
x_2	0	1	0	1	0	1
s_3	0	0	−1	−1	1	0
$\hat{c}$	0	0	−1	−1	0	2

The basic variable $s_3 = 0$, so the BFS is degenerate. Under the chapter's $\hat{c} = c - z$ convention, all nonbasic reduced costs are nonpositive, confirming optimality.

Exercise 53 (BFSs and extreme points). Define extreme points and prove the correspondence with basic feasible solutions.

Solution. A point $x \in P$ is extreme if

$$x = \lambda u + (1-\lambda)v, \quad u, v \in P, \quad 0 < \lambda < 1$$

implies $u = v = x$.

Let x be a BFS with basis B and nonbasic set N. Suppose $x = \lambda u + (1-\lambda)v$ for feasible u, v. Since $x_N = 0$ and $u_N, v_N \geq 0$, every nonbasic coordinate must satisfy $u_N = v_N = 0$. Feasibility then gives

$$Bu_B = b = Bv_B.$$

Invertibility of B yields $u_B = v_B = B^{-1}b = x_B$, hence $u = v = x$. Therefore every BFS is extreme.

For a concrete converse example, take

$$x_1 + x_3 = 1, \qquad x_2 + x_4 = 1, \qquad x \geq 0.$$

Its feasible set is a square. The four extreme points are obtained by choosing one variable from each equation, for example

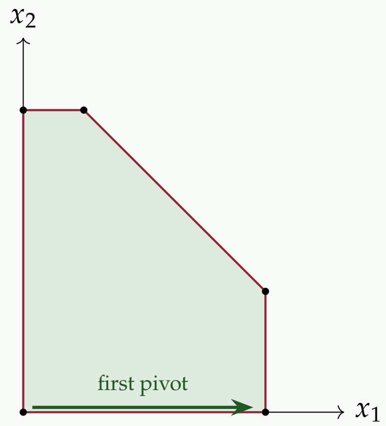
$$(1, 1, 0, 0)$$

with basis $\{x_1, x_2\}$. Likewise each of the other three vertices has at least one basis. In general, the standard rank argument extends this converse to every extreme point.

Exercise 54 (Geometric interpretation of a pivot). Draw the feasible region, perform the first pivot, and interpret the move geometrically.

Solution. The trace says six vertices, but the region actually has five:

$$(0,0), \quad (4,0), \quad (4,2), \quad (1,5), \quad (0,5).$$



At the origin, x_1 has objective coefficient 2 and x_2 coefficient 1, so x_1 enters. The ratio test gives

$$6/1 = 6 \quad \text{from } x_1 + x_2 \le 6, \qquad 4/1 = 4 \quad \text{from } x_1 \le 4.$$

Thus the slack of $x_1 \le 4$ leaves and the new vertex is $(4,0)$. During the pivot all other nonbasic variables remain zero and one variable increases continuously, so the feasible trajectory has one degree of freedom: it is precisely the polytope edge from $(0,0)$ to $(4,0)$.

Exercise 55 (Choosing the Big-M constant). Explain what a valid penalty must dominate, whether the supplied data bounds determine one, and give a failure caused by a small penalty.

Solution. M must make every positive artificial variable more expensive than any objective improvement obtainable by violating the original equalities. Conceptually, if the original objective can vary by at most $U - L$ and every nonzero artificial violation is at least $\varepsilon > 0$, then

$$M > \frac{U - L}{\varepsilon}$$

is sufficient.

No safe numerical bound follows from $|A_{ij}| \le K$, $|b_i| \le R$, and the dimensions alone. The objective coefficients are not bounded in the question, and with arbitrary real data the smallest relevant ε can be arbitrarily close to zero. Additional rational encoding-size bounds and a bound on c are necessary; the requested formula in terms of only K, R, m, n is therefore under-specified.

For a concrete small-M failure, consider the feasible problem

$$\min x \qquad \text{s.t.} \qquad x \ge 10, \quad x \ge 0.$$

Write $x - s + a = 10$ and maximise the penalised objective $-x - Ma$. The valid original solution $x = 10, a = 0$ has value -10. If $M = 1/2$, the artificial solution $x = s = 0, a = 10$ has penalised value -5 and is incorrectly preferred. A solver that treats a positive artificial at the penalised optimum as an infeasibility certificate would falsely declare this feasible LP infeasible.

For a genuinely infeasible LP, no value of M can force all artificials to zero; explicitly inspecting their final values still detects infeasibility. The main danger of too small an M is therefore the opposite false conclusion on a feasible problem, which corrects the ambiguous wording of the question.

CHAPTER 5

Duality & Valid Inequalities

Exercise 1 (Dual of a standard max LP). Write the dual and explain its dimensions.

Solution. The dual is

$$\begin{aligned} \text{minimize} \quad & 4u_1 + 6u_2 \\ \text{subject to} \quad & u_1 + u_2 \geq 3, \\ & u_1 + 3u_2 \geq 5, \\ & u_1, u_2 \geq 0. \end{aligned}$$

A max problem with $\leq$ constraints becomes a min problem. Each of the two primal constraints creates one dual variable, while each of the two primal variables creates one dual constraint. In general an $m \times n$ primal produces an $n \times m$ dual.

Exercise 2 (Dual of a min LP with $\geq$ constraints). Dualise the three-variable covering LP.

Solution.

$$\begin{aligned} \text{maximize} \quad & 5u_1 + 3u_2 \\ \text{subject to} \quad & u_1 + 2u_2 \leq 2, \\ & 2u_1 + u_2 \leq 4, \\ & u_1 \leq 1, \\ & u_1, u_2 \geq 0. \end{aligned}$$

The two primal constraints give two nonnegative dual variables; the three nonnegative primal variables give three $\leq$ dual constraints.

Exercise 3 (Dual with equality constraints). Dualise a max problem containing one equality constraint.

Solution. Let u_1 correspond to the equality:

$$\begin{aligned} \text{minimize} \quad & 10u_1 + 14u_2 + 12u_3 \\ \text{subject to} \quad & u_1 + 2u_2 + u_3 \geq 4, \\ & u_1 + u_2 \geq 6, \\ & u_1 + 3u_3 \geq 2, \\ & u_1 \text{ free}, \qquad u_2, u_3 \geq 0. \end{aligned}$$

An equality may be multiplied by a multiplier of either sign, hence its dual variable is unrestricted. A primal $\leq$ row still produces a nonnegative dual variable.

Exercise 4 (Dual with a free variable). Write the dual and explain the equality generated by x_3.

Solution.

$$\begin{aligned} \text{minimize} \quad & 6u_1 + 2u_2 \\ \text{subject to} \quad & u_1 + u_2 \geq 1, \\ & u_1 - u_2 \geq -1, \\ & u_1 = 2, \\ & u_1, u_2 \geq 0. \end{aligned}$$

The x_3 column is $(1, 0)^\top$, so its dual expression is u_1. Because a free variable may move positively or negatively, both dual inequality directions must hold; together they force equality.

Exercise 5 (Dual with non-positive variable). Dualise the problem with $x_2 \leq 0$.

Solution.

$$\begin{aligned} \text{minimize} \quad & 8u_1 + 7u_2 \\ \text{subject to} \quad & 2u_1 + u_2 \geq 5, \\ & u_1 + 2u_2 \leq 3, \\ & u_1, u_2 \geq 0. \end{aligned}$$

The primal rows are still $\leq$, hence $u \geq 0$. The non-positive variable reverses its corresponding dual inequality from $\geq$ to $\leq$.

Exercise 6 (Mixed constraint types). Dualise the minimisation problem with $\geq$, equality, and $\leq$ rows.

Solution.

$$\begin{aligned} \text{maximize} \quad & 4u_1 + 3u_2 + 5u_3 \\ \text{subject to} \quad & u_1 + u_3 \leq 1, \\ & u_1 + u_2 \leq 2, \\ & u_2 + u_3 \leq 3, \\ & u_1 \geq 0, \quad u_2 \text{ free}, \quad u_3 \leq 0. \end{aligned}$$

For a min primal, a $\geq$ row gives $u_i \geq 0$, an equality gives a free multiplier, and a $\leq$ row gives $u_i \leq 0$. There are three dual variables and three dual constraints.

Exercise 7 (Diet LP dual). Formulate the diet problem and interpret its dual prices.

Solution. With food quantities $x_1, x_2 \geq 0$:

$$\begin{aligned} \text{minimize} \quad & 3x_1 + 5x_2 \\ \text{subject to} \quad & 400x_1 + 200x_2 \geq 2000, \\ & 10x_1 + 15x_2 \geq 50, \\ & 5x_1 + 8x_2 \geq 30. \end{aligned}$$

Its dual is

$$\begin{aligned} \text{maximize} \quad & 2000u_1 + 50u_2 + 30u_3 \\ \text{subject to} \quad & 400u_1 + 10u_2 + 5u_3 \leq 3, \\ & 200u_1 + 15u_2 + 8u_3 \leq 5, \\ & u \geq 0. \end{aligned}$$

u_i is the marginal value of one extra unit of nutrient requirement. The dual maximises the imputed value of the requirements without pricing any food bundle above its market cost. By strong duality its optimum equals the minimum diet cost; here food 1 alone at $x_1 = 6$ costs 18 and is optimal.

Exercise 8 (Dual of the furniture workshop LP). Use complementary slackness to solve the workshop dual.

Solution.

$$\begin{aligned} \text{minimize} \quad & 40u_1 + 60u_2 + 50u_3 \\ \text{subject to} \quad & u_1 + 2u_2 + u_3 \geq 40, \\ & u_1 + u_2 + u_3 \geq 30, \\ & u \geq 0. \end{aligned}$$

At $(20, 20)$ the first two primal rows are tight and the third has slack 10, so $u_3 = 0$. Both primal variables are positive, hence both dual inequalities are equalities:

$$u_1 + 2u_2 = 40, \qquad u_1 + u_2 = 30.$$

Thus $u^* = (20, 10, 0)$ and $40(20) + 60(10) = 1400$, equal to the primal value.

Exercise 9 (Proving weak duality from first principles). Prove weak duality for the standard max–min pair.

Solution. For primal-feasible x and dual-feasible u,

$$c^\top x \leq (A^\top u)^\top x = u^\top Ax \leq u^\top b = b^\top u.$$

The first inequality uses $x \geq 0$ and $A^\top u \geq c$; the second uses $u \geq 0$ and $Ax \leq b$.

Exercise 10 (Verifying weak duality numerically). Check the proposed primal and dual points and their bounds.

Solution. At $x = (3, 1)$ the resource uses are $(22, 5) \leq (24, 6)$ and the objective is 19. At $u = (1/2, 2)$ the dual left sides are 5 and 6, so the point is dual feasible, with value

$$24(1/2) + 6(2) = 24.$$

Weak duality gives $19 \leq z^* \leq 24$. Neither point is optimal: the feasible point $(3, 3/2)$ and dual point $(3/4, 1/2)$ both have value 21, certifying the true optimum.

Exercise 11 (Using weak duality to bound without solving). Construct a strong dual upper bound for the three-variable LP.

Solution. The dual is

$$\min 10u_1 + 18u_2 \quad \text{s.t.} \quad u_1 + 2u_2 \geq 3, u_1 + u_2 \geq 2, u_1 + 3u_2 \geq 5, \; u \geq 0.$$

The intersection of the first and third relevant lower bounds gives $u = (1/2, 3/2)$, which is feasible and has value 32. It is in fact dual optimal. Weak duality guarantees

$$c^\top x \leq 32$$

for every primal-feasible x, and more generally every primal feasible value is no larger than every dual feasible value.

Exercise 12 (Weak duality and infeasibility/unboundedness). Classify five assertions about primal and dual status.

Solution.

1. **False.** If the primal is infeasible, the dual may be unbounded or infeasible.
2. **False.** A dual-infeasible primal may be unbounded or infeasible.
3. **True.** A feasible dual point would give a finite upper bound, contradicting primal unboundedness.
4. **False.** Both members of a primal–dual pair can be infeasible.
5. **True**, whenever the optimal values exist; it is the optimal-value form of weak duality.

Exercise 13 (Gap between primal and dual). Interpret primal value 120 and dual value 135.

Solution. Initially

$$120 \le z^* \le 135.$$

A new primal value 128 raises the lower bound: $128 \le z^* \le 135$, leaving an absolute gap of 7. A feasible primal and feasible dual solution with equal objective values certify optimality immediately; equivalently, feasibility plus complementary slackness is sufficient.

Exercise 14 (Strong duality: statement and conditions). State strong duality and the exceptional status cases.

Solution. If either member of an LP primal–dual pair has a finite optimum, then both have optimal solutions and

$$z_P^* = z_D^*.$$

Equivalently, if both are feasible, both are bounded in their respective directions and the optimal values coincide. If the primal is unbounded, the dual is infeasible. If the primal is infeasible while the dual is feasible, the dual must be unbounded; primal infeasibility alone also permits dual infeasibility.

Exercise 15 (CS to find dual optimal — two-variable LP). Use complementary slackness at $x^* = (2, 3)$.

Solution. The dual is

$$\min 9u_1 + 8u_2 \quad \text{s.t.} \quad 3u_1 + u_2 \ge 6, \quad u_1 + 2u_2 \ge 8, \quad u \ge 0.$$

Both primal constraints are tight and both variables are positive. Therefore both dual inequalities are tight:

$$3u_1 + u_2 = 6, \qquad u_1 + 2u_2 = 8.$$

Solving gives $u^* = (4/5, 18/5)$ and $9(4/5) + 8(18/5) = 36$.

Exercise 16 (CS to find dual optimal — three-constraint LP). Find a dual optimum corresponding to $x^* = (3, 1)$.

Solution.

$$\begin{aligned} \text{minimize} \quad & 4u_1 + 6u_2 + 7u_3 \\ \text{subject to} \quad & u_1 + u_2 + 2u_3 \ge 2, \\ & u_1 + 3u_2 + u_3 \ge 3, \qquad u \ge 0. \end{aligned}$$

All three primal constraints are tight, and $x_1, x_2 > 0$, so both dual constraints are equalities. The system has multiple solutions; one is

$$u^* = (3/2, 1/2, 0).$$

It satisfies both dual constraints at equality and has value $4(3/2)+6(1/2) = 9$.

Exercise 17 (CS to find dual optimal — three-variable LP). Find the dual optimum corresponding to $x^* = (0, 0, 6)$.

Solution.

$$\min 6u_1 + 8u_2 \quad \text{s.t.} \quad u_1 + 2u_2 \geq 1,\ u_1 \geq 2,\ u_1 + u_2 \geq 3,\ u \geq 0.$$

The first primal row is tight and the second has slack 2, hence $u_2 = 0$. Since $x_3 > 0$, its dual constraint is tight, so $u_1 + u_2 = 3$ and $u^* = (3, 0)$. The values agree: $3(6) = 18 = 6(3) + 8(0)$.

Exercise 18 (Strong duality verification — production LP). Check the claimed production optimum and give the correct certificate.

Solution. The dual is

$$\min 120u_1 + 80u_2 \quad \text{s.t.} \quad 2u_1 + u_2 \geq 10, 3u_1 + u_2 \geq 12, u_1 + 2u_2 \geq 8,\ u \geq 0.$$

The claimed $x = (0, 40, 0)$ has value 480, but its second resource has slack 40. CS would force $u_2 = 0$ and $x_2 > 0$ would force $3u_1 = 12$, giving $u_1 = 4$; then $2u_1 = 8 < 10$, so the dual is infeasible. The claim is therefore false.

The correct primal optimum is

$$x^* = (160/3, 0, 40/3), \qquad z^* = 640,$$

and the correct dual optimum is $u^* = (4, 2)$. The first and third dual constraints are tight, the second is slack, and $120(4) + 80(2) = 640$.

Exercise 19 (Degenerate primal optimum). Explain why degeneracy may leave the dual optimum undetermined by CS.

Solution. For standard max–min form, complementary slackness is

$$u_i(b_i - A_i x) = 0, \qquad x_j(A_j^\top u - c_j) = 0.$$

A degenerate basic variable may equal zero even though its column is in the basis. Its second CS equation then imposes no equality on the corresponding dual constraint. Consequently the equalities obtained from positive primal variables may be insufficient to determine all dual multipliers. The dual can have multiple optimal points, often an edge or higher-dimensional optimal face.

Exercise 20 (Dual of dual equals primal). Dualise the dual and interpret the symmetry.

Solution. Treating

$$\min\{b^\top u : A^\top u \geq c,\ u \geq 0\}$$

as a min problem with $\geq$ rows gives

$$\max\{c^\top x : Ax \leq b,\ x \geq 0\},$$

which is exactly the primal. Thus variables and constraints exchange roles symmetrically. The statement is **true** when sign and constraint conventions are dualised consistently.

Exercise 21 (Writing CS conditions). Write every complementary slackness equation for the displayed pair.

Solution. The primal-row conditions are

$$u_1(8 - 2x_1 - x_2) = 0, \qquad u_2(7 - x_1 - 2x_2) = 0.$$

The dual-row conditions are

$$x_1(2u_1 + u_2 - 4) = 0, \qquad x_2(u_1 + 2u_2 - 5) = 0.$$

Together with primal and dual feasibility, these equations are an optimality certificate.

Exercise 22 (Checking a primal-dual pair via CS). Test $x = (3, 2)$ and $u = (1, 2)$.

Solution. Both primal rows are tight: $2(3) + 2 = 8$ and $3 + 2(2) = 7$. Both dual rows are tight: $2(1) + 2 = 4$ and $1 + 2(2) = 5$. Hence all four CS products vanish. The points are feasible and their values agree:

$$4(3) + 5(2) = 22 = 8(1) + 7(2).$$

They are therefore optimal.

Exercise 23 (CS: given primal solution, find dual). Find a dual optimum corresponding to $(4, 6, 0)$.

Solution. The dual is

$$\min 10u_1 + 14u_2 \quad \text{s.t.} \quad u_1 + 2u_2 \geq 7, \quad u_1 + u_2 \geq 5, \quad u_1 \geq 3, \quad u \geq 0.$$

Both primal constraints are tight, so neither multiplier is forced to zero. Since $x_1, x_2 > 0$, the first two dual constraints are equalities:

$$u_1 + 2u_2 = 7, \qquad u_1 + u_2 = 5.$$

Thus $u^* = (3, 2)$ and both values equal $7(4) + 5(6) = 58 = 10(3) + 14(2)$.

Exercise 24 (CS: non-optimal pair detection). Check feasibility, CS, and the gap for the proposed pair.

Solution. $x = (2, 2)$ is primal feasible and makes both rows tight. $u = (2, 1)$ is dual feasible because its dual left sides are 3 and 4. However, $x_2 > 0$ while its dual constraint has positive slack $4 - 2 = 2$, so CS fails. The objective values are

$$z_P = 3(2) + 2(2) = 10, \qquad z_D = 4(2) + 6(1) = 14.$$

The duality gap is 4, and the pair is not optimal.

Exercise 25 (CS with equality constraints). Summarise complementary slackness for equalities and free variables.

Solution. An equality row has zero primal slack identically, so its multiplier is free and the row-side CS product imposes no additional restriction. A free primal variable corresponds to an equality dual constraint, so its dual slack is identically zero.

primal feature	dual feature	CS implication
$\leq$ row	$u_i \geq 0$	$u_i(b_i - A_i x) = 0$
$=$ row	u_i free	row product automatic
$x_j \geq 0$	$A_j^\top u \geq c_j$	$x_j(A_j^\top u - c_j) = 0$
x_j free	$A_j^\top u = c_j$	column product automatic

Exercise 26 (CS optimality certificate). Check the claimed primal and dual solutions for the three-resource LP.

Solution. $x = (30, 0, 0)$ is primal feasible, but all three rows have positive slack: $60, 180, 270$. The claimed dual $u = (5/6, 0, 0)$ satisfies the first dual constraint at equality but gives

$$4(5/6) = 10/3 < 4, \qquad 2(5/6) = 5/3 < 3,$$

so it is not dual feasible. CS also fails because $u_1 > 0$ while the first primal row is slack. Finally,

$$c^\top x = 150, \qquad b^\top u = 200.$$

The claimed pair is not optimal; the corrected optimum was computed in Chapter 4 as $1095/4$.

Exercise 27 (CS and reduced costs). Relate simplex reduced costs to dual feasibility.

Solution. Let $u^\top = c_B^\top B^{-1}$. Then

$$\bar{c}_j = c_j - u^\top A_j.$$

The condition $\bar{c}_j \leq 0$ is exactly $A_j^\top u \geq c_j$, the jth dual constraint. Thus all reduced costs nonpositive means that the basis multipliers form a dual-feasible solution. Since the primal BFS is feasible and $c_B^\top B^{-1} b = b^\top u$, primal and dual values coincide, proving optimality.

Exercise 28 (Farkas' Lemma: statement). State Farkas' alternative and explain its cone geometry.

Solution. Exactly one of the following systems has a solution:

$$\begin{array}{ll} \text{(i)} & Ax = b, \quad x \geq 0, \\ \text{(ii)} & A^\top y \leq 0, \quad b^\top y > 0. \end{array}$$

The vectors Ax with $x \geq 0$ form the cone generated by the columns of A. If b lies in this cone, (i) gives its nonnegative coefficients. If not, y defines a separating hyperplane: every cone vector has $y^\top Ax \leq 0$, while $y^\top b > 0$ places b strictly on the other side.

Exercise 29 (Farkas' Lemma: proving infeasibility). Find a certificate for the inconsistent three-equation system.

Solution. Take

$$y = (1, -1, 1).$$

For

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 1 & -1 \end{pmatrix}, \qquad b = (5, 5, 2)^\top,$$

we have

$$A^\top y = (0, 0)^\top \leq 0, \qquad b^\top y = 5 - 5 + 2 = 2 > 0.$$

If $Ax = b$ and $x \geq 0$ existed, then $b^\top y = x^\top A^\top y \leq 0$, contradicting $b^\top y = 2$.

Exercise 30 (Farkas alternative for inequalities). Certify infeasibility of $x_1 + x_2 \leq -1$, $x \geq 0$.

Solution. Use the scalar multiplier $y = 1$. Then $y \geq 0$,

$$A^\top y = (1, 1)^\top \geq 0, \qquad b^\top y = -1 < 0.$$

This is the stated Farkas certificate. Algebraically, a nonnegative sum $x_1 + x_2$ cannot be at most -1.

Exercise 31 (LP infeasibility certificate via Farkas). Show the two lower-bound inequalities are inconsistent and certify it.

Solution. The rows require $x_1 - x_2 \geq 3$ and $x_2 - x_1 \geq 1$; summing gives $0 \geq 4$, impossible. In $\leq$ form they are

$$-x_1 + x_2 \leq -3, \qquad x_1 - x_2 \leq -1.$$

With $y = (1, 1) \geq 0$,

$$A^\top y = (0, 0)^\top \geq 0, \qquad b^\top y = -4 < 0.$$

This satisfies the inequality-form Farkas alternative and certifies an empty feasible region.

Exercise 32 (Connecting Farkas to duality). Describe infeasibility certificates and their relation to the dual.

Solution. Primal feasibility asks for $x \geq 0$ satisfying $Ax \leq b$. In the inequality alternative, infeasibility is certified by a multiplier $y \geq 0$ with $A^\top y \geq 0$ and $b^\top y < 0$ (after fixing a consistent sign convention). Such a vector is a dual-type ray: it is not necessarily a feasible solution of the original objective-dependent dual, but it proves that a nonnegative combination of primal rows yields the contradiction $0 \leq b^\top y < 0$.

The final statement is **true**: if the primal is infeasible, its dual can be unbounded or infeasible. Farkas does not select which of those two status cases occurs.

Exercise 33 (Shadow prices — production LP). Interpret the supplied multipliers and check their consistency.

Solution. If the stated values were optimal and the basis remained unchanged, $u_1 = 30$ would mean one extra unit of raw material raises maximum profit by about \$30; $u_3 = 0$ would mean marginal labour has no value; two extra machine hours would raise profit by $2(10) = \$20$.

However, $(30, 10, 0)$ is dual feasible but not optimal: its dual value is 4500, while the primal optimum is

$$(x_1, x_2) = (60, 30), \qquad z^* = 4200,$$

with correct dual prices $(0, 20, 10)$. Thus the numerical interpretations above describe the intended exercise, not the actual optimum of the displayed LP.

Exercise 34 (Shadow prices — diet problem). Interpret nutrient prices and price a proposed food.

Solution. An extra 5 grams of required protein changes minimum cost by $5(0.10) = \$0.50$ within the sensitivity range. A zero iron price means the iron requirement has no marginal cost, usually because it is nonbinding or redundant locally.

The new food's imputed nutrient value is

$$300(0.04) + 12(0.10) = 13.20.$$

Its market cost is only \$2.50, so its minimisation reduced cost is $2.50 - 13.20 = -\$10.70$. It is attractive and the current basis cannot remain optimal.

Exercise 35 (Shadow prices — transportation LP). Interpret supply and demand prices and compute a route reduced cost.

Solution. In a cost-minimisation transportation model, an extra unit of supply can lower cost, hence a supply marginal value may be negative. The demand price $v_1 = 5$ means one more unit required at Store 1 raises minimum cost by approximately \$5.

For route $(1,1)$ the dual imputed cost is $u_1 + v_1 = -3 + 5 = 2$, so

$$\bar{c}_{11} = c_{11} - (u_1 + v_1) = 8 - 2 = 6.$$

A zero reduced cost would mean the route is exactly competitive with the shadow value of its endpoints and may be basic or support an alternative optimum.

Exercise 36 (Identifying binding constraints from shadow prices). Infer activity information from $u^* = (0, 12, 0, 7, 0)$.

Solution. Constraints 2 and 4 are definitely binding because their multipliers are positive. No row is definitely nonbinding from the prices alone: a zero multiplier can accompany either positive slack or degeneracy at a binding row. To improve a max objective locally, increase the RHS of rows 2 and 4; row 2 has the larger marginal return, 12 per unit versus 7.

Exercise 37 (Negative shadow prices). Interpret a positive price on a minimisation requirement.

Solution. For a min problem with $a_i^\top x \geq b_i$, a positive shadow price means tightening the requirement raises minimum cost. Increasing b_i by one changes the value by approximately $u_i^* > 0$.

A minimal picture is

$$\min x_1 + x_2 \quad \text{s.t.} \quad x_1 + x_2 \geq b, \quad x \geq 0.$$

The optimal face is the line segment $x_1 + x_2 = b$, its value is $z^*(b) = b$, and the slope (shadow price) is 1.

Exercise 38 (Shadow price of an equality constraint). Interpret a free multiplier $u^* = 15$.

Solution. Locally,

$$z^*(B + \Delta) \approx z^*(B) + 15\Delta.$$

Thus one extra budget unit raises the objective by 15 while the current basis remains valid. An equality multiplier is free because changing the prescribed level can help or hurt: unlike a one-sided resource bound, either direction may shrink the economically favourable set. Here the positive sign specifically implies that increasing B raises the optimal value by 15 locally.

Exercise 39 (RHS ranging — single constraint). Express the optimal basic variables as functions of b_1.

Solution. Solving

$$6x_1 + 4x_2 = b_1, \qquad x_1 + 2x_2 = 6$$

gives

$$x_1 = \frac{b_1}{4} - 3, \qquad x_2 = \frac{9}{2} - \frac{b_1}{8}.$$

At $b_1 = 26$, the solution is $(7/2, 5/4)$. The basis remains feasible when both expressions are nonnegative:

$$12 \le b_1 \le 36.$$

Reduced costs do not change with b, so feasibility preserves optimality throughout this interval.

Exercise 40 (RHS ranging — which constraint is most valuable?). Use prices $(8, 0, 5)$ to assess capacity investments.

Solution. Resource 1 has the greatest marginal value, 8 per unit. Four extra units of resource 3 are worth $4(5) = 20$, exceeding their total cost \$18, so the purchase yields a predicted net gain of \$2 within the valid range. Ten extra units of resource 2 have first-order value zero, so they do not change profit while the current basis remains optimal.

Exercise 41 (Objective coefficient ranging — basic variable). Find the range of c_1 preserving basis $\{x_1, x_2\}$.

Solution. For

$$B = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}, \qquad B^{-1} = \frac{1}{5}\begin{pmatrix} 3 & -1 \\ -1 & 2 \end{pmatrix},$$

the dual multipliers are

$$(c_1, 4)B^{-1} = \left(\frac{3c_1 - 4}{5}, \frac{8 - c_1}{5}\right).$$

The nonbasic slacks have nonpositive reduced costs exactly when these multipliers are nonnegative, hence

$$\frac{4}{3} \le c_1 \le 8.$$

Inside this interval the basis and solution $(3, 2)$ stay fixed; only the objective value changes. Above 8 the second multiplier becomes negative, so the current basis loses optimality and a different vertex must be selected.

Exercise 42 (Objective coefficient ranging — non-basic variable). Interpret reduced cost $\bar{c}_3 = -4$.

Solution. c_3 may increase by 4 before the reduced cost reaches zero. At that boundary the current basis remains optimal but an alternative optimum may appear. A larger increase makes $\bar{c}_3 > 0$, so x_3 becomes an improving entering candidate and re-optimisation is required. Any decrease makes the reduced cost more negative, so it cannot disturb the current max basis.

Exercise 43 (Adding a new variable — sensitivity perspective). Price a proposed product using $u^* = (3, 1)$.

Solution.

$$\bar{c}_3 = 7 - u^{*\top} A_3 = 7 - [3(2) + 1(1)] = 0.$$

The product is exactly break-even at shadow prices. Introducing it cannot improve the current objective, though it may create an alternative optimum. If its profit were 5, then $\bar{c}_3 = 5 - 7 = -2$, so it should remain at zero.

Exercise 44 (Sensitivity analysis: simultaneous RHS changes). State and apply the 100% rule.

Solution. For each RHS change, divide its magnitude by the allowable change in the same direction. If the sum of these fractions is at most one, the current basis is guaranteed to remain feasible and optimal; a larger sum is inconclusive, not a proof of failure.

Here

$$\theta_1 = \frac{38 - 30}{40 - 30} = 0.8, \qquad \theta_2 = \frac{22 - 15}{25 - 15} = 0.7.$$

Their sum is $1.5 > 1$, so the rule gives no guarantee. The new basic solution must be checked directly.

Exercise 45 (Interpretation of the dual problem as pricing). Explain the resource-pricing interpretation of the dual.

Solution. At prices u_i, selling all available resources earns $\sum_i b_i u_i = b^\top u$. The constraint $A_j^\top u \geq c_j$ says the resources consumed by one unit of product j cost at least its product profit; otherwise the factory would rather buy resources and produce that item. The entrepreneur minimises the total valuation while keeping every production opportunity covered. The factory maximises achievable profit, while the entrepreneur seeks the cheapest resource-price certificate dominating that profit; strong duality balances the two incentives.

Exercise 46 (True/False: duality fundamentals). Classify five duality statements.

Solution.

(a) **True**, after writing the primal and dual with consistent general sign conventions.
(b) **True** if a finite optimum exists, by strong duality.
(c) **False.** Primal variables correspond to dual constraints, not dual variables; the counts swap.
(d) **False.** Equal finite values require feasibility and boundedness; infeasible cases have no such equality.
(e) **True.** Feasibility of both sides plus weak duality rules out either being unbounded, and strong duality applies.

Exercise 47 (True/False: complementary slackness). Classify five CS statements.

Solution.

(a) **True.**
(b) **True.**
(c) **True**: primal feasibility, dual feasibility, and CS are necessary and sufficient.
(d) **False**; they are also sufficient when both points are feasible.
(e) **True** for the row-side products: equality rows have zero slack automatically. Their dual variables are free, although other dual constraints may still restrict them.

Exercise 48 (True/False: shadow prices and sensitivity). Classify five sensitivity statements.

Solution.

(a) **False.** Zero marginal value does not imply the row can be removed or that the optimizer remains unchanged.
(b) **False.** A shadow price is exact throughout its allowable RHS range, not merely infinitesimally.
(c) **True** for max problems with $\leq$ rows.
(d) **True**, by complementary slackness.
(e) **False** as a blanket statement: individual ranges are computed separately, but simultaneous changes can interact and alter the basis.

Exercise 49 (True/False: infeasibility and unboundedness). Classify five status statements.

Solution.

(a) **False**; the dual may also be infeasible.
(b) **True**.
(c) **True**.
(d) **True**; otherwise a feasible primal point would bound the dual by weak duality.
(e) **True**, with the signs adjusted to the chosen inequality form.

Exercise 50 (True/False: valid inequalities and Farkas). Classify five statements about aggregation and certificates.

Solution.

(a) **True**, provided all inequalities are first oriented in the same direction.
(b) **False**; it may be a derived supporting inequality.
(c) **True**: dual feasibility ensures the aggregated resource inequality dominates the primal objective coefficients.
(d) **True**; Farkas provides a finite linear multiplier certificate.
(e) **False**; derived inequalities can expose bounds, redundancy, and infeasibility in ordinary LPs as well.

Exercise 51 (Bounding the optimal value via duality). Produce matching primal and dual bounds by inspection.

Solution. The feasible point $x = (0, 8, 0)$ has value 32, so $z^* \geq 32$. The dual is

$$\begin{aligned} \text{minimize} \quad & 8u_1 + 10u_2 + 7u_3 \\ \text{subject to} \quad & u_1 + 2u_2 + u_3 \geq 3, \\ & u_1 + u_2 \geq 4, \\ & u_1 + 2u_3 \geq 2, \qquad u \geq 0. \end{aligned}$$

Taking $u = (4, 0, 0)$ is feasible and has value 32. Therefore

$$32 \leq z^* \leq 32,$$

so both points are optimal and $z^* = 32$. Coincident feasible bounds are a complete optimality certificate.

Exercise 52 (Primal-dual relationship: correspondence table). Complete the max-primal correspondence table.

Solution.

primal max feature	dual min feature
$\leq$ row i	$u_i \geq 0$
$\geq$ row i	$u_i \leq 0$
$=$ row i	u_i free
$x_j \geq 0$	constraint $j : \geq$
$x_j \leq 0$	constraint $j : \leq$
x_j free	constraint $j : =$

For example, $\max\{x : x \geq 1,\ x \leq 0\}$ assigns a nonpositive multiplier to the $\geq$ row, while the nonpositive variable creates a $\leq$ dual constraint, illustrating both reversed entries.

Exercise 53 (Degeneracy and the dual). Describe the CS information available at the degenerate BFS.

Solution. There are two dual variables, one per primal constraint. Dual-side CS gives an equality only for positive primal variables: $x_1 = 4$ supplies one equation, while $x_3 = 0$ and all nonbasic zero variables supply none. Primal-side CS gives u_i times each row slack equal to zero; only strictly slack rows force a multiplier to zero.

Thus CS may provide fewer than two independent equations and need not identify a unique dual point. Intersecting the dual feasible polyhedron with the optimal-value hyperplane produces the dual optimal set, typically a segment or higher-dimensional face.

Exercise 54 (Verifying dual feasibility). Check the claimed dual point against the primal optimum.

Solution. The dual is

$$\min 6u_1 + 8u_2 + 9u_3 \quad \text{s.t.} \quad u_1 + u_3 \geq 8, \quad 2u_2 + u_3 \geq 5, \quad u \geq 0.$$

The point $(5, 1, 3)$ is feasible, and its value is $30 + 8 + 27 = 65$. The primal point $(6, 3)$ is feasible and has value $48 + 15 = 63$, so the values do not agree. CS fails because the second primal row has slack 2 but $u_2 = 1 > 0$.

The correct dual optimum is $(3, 0, 5)$, with value $18 + 45 = 63$. Both dual constraints are tight and all CS conditions hold.

Exercise 55 (Constructing a valid inequality upper bound). Find the tightest nonnegative aggregation dominating the objective.

Solution. Choose $y \geq 0$ so that

$$2y_1 + y_2 + y_3 \geq 60, \qquad y_1 + y_2 \geq 50.$$

Then the nonnegative combination of primal rows yields

$$60x_1 + 50x_2 \leq 100y_1 + 60y_2 + 40y_3.$$

Minimising the right side is exactly the dual:

$$\min 100y_1 + 60y_2 + 40y_3 \quad \text{s.t. the two inequalities above,} \quad y \geq 0.$$

The optimum is $y = (10, 40, 0)$, giving the valid inequality

$$60x_1 + 50x_2 \leq 3400.$$

It is tight at the primal optimum $(40, 20)$, whose profit is 3400.

Solving ILPs

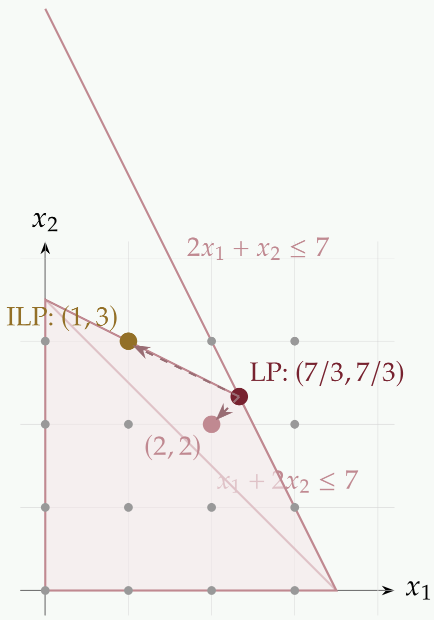
Exercise 1 (Why rounding is not enough). Solve the relaxation, compare two rounding rules, and find the integer optimum.

Solution. The two active constraints give

$$x_1 + 2x_2 = 7, \qquad 2x_1 + x_2 = 7,$$

hence $x^* = (7/3, 7/3)$ and $z^*_{\text{LP}} = 56/3$. Flooring and nearest-integer rounding both give $(2, 2)$, which is feasible and has value 16. Enumeration gives the better integer point $(1, 3)$, of value 18, and no other integer point has larger value. Thus

$$z^*_{\text{ILP}} = 18, \qquad z^*_{\text{LP}} - z^*_{\text{ILP}} = \frac{2}{3}.$$



Rounding is therefore a heuristic: it may lose value, and in other instances it may even destroy feasibility.

Exercise 2 (LP relaxation upper bound). Explain the LP bound and construct examples with gap one and zero.

Solution. The LP relaxation drops the integrality constraints, so its feasible set contains every integer-feasible point. Maximizing over a larger set cannot yield a smaller maximum: $z^*_{\text{LP}} \geq z^*_{\text{ILP}}$.

For an exact gap of 1, consider

$$\max\ 2x_1 + 2x_2 \quad \text{s.t.} \quad x_1 + x_2 \leq \tfrac{3}{2}, \quad x \in \mathbb{Z}^2_{\geq 0}.$$

The LP optimum is $x_1 = x_2 = 3/4$, giving $z^*_{\text{LP}} = 2 \cdot \frac{3}{4} + 2 \cdot \frac{3}{4} = 3$. The only integer points satisfying $x_1 + x_2 \leq 1.5$ are $(0,0), (1,0), (0,1)$; the best objective is 2. Hence $z^*_{\text{LP}} - z^*_{\text{ILP}} = 1$.

Replacing the RHS 3/2 by 2 gives a zero-gap example:

$$\max\ 2x_1 + 2x_2 \quad \text{s.t.} \quad x_1 + x_2 \leq 2, \quad x \in \mathbb{Z}^2_{\geq 0}.$$

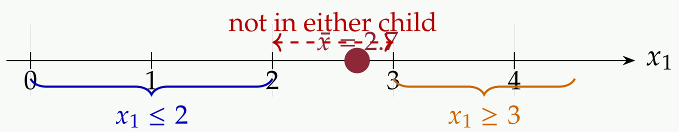
Now the LP optimum is $(1,1)$ with value 4 — already integer — so $z^*_{\text{LP}} = z^*_{\text{ILP}} = 4$. The gap vanishes because the constraint happens to have an integer optimal vertex.

Exercise 3 (Branching and the feasible set). Branch at $\bar{x} = (2.7, 1, 3.4)$ and explain the partition.

Solution. x_1 and x_3 are fractional. Branching on x_1 produces

$$Ax \leq b,\ x \geq 0,\ x_1 \leq 2 \qquad \text{and} \qquad Ax \leq b,\ x \geq 0,\ x_1 \geq 3.$$

Every integer value of x_1 satisfies exactly one branch, whereas an LP point with $2 < x_1 < 3$ satisfies neither. The *incumbent* is the best integer-feasible solution found so far.



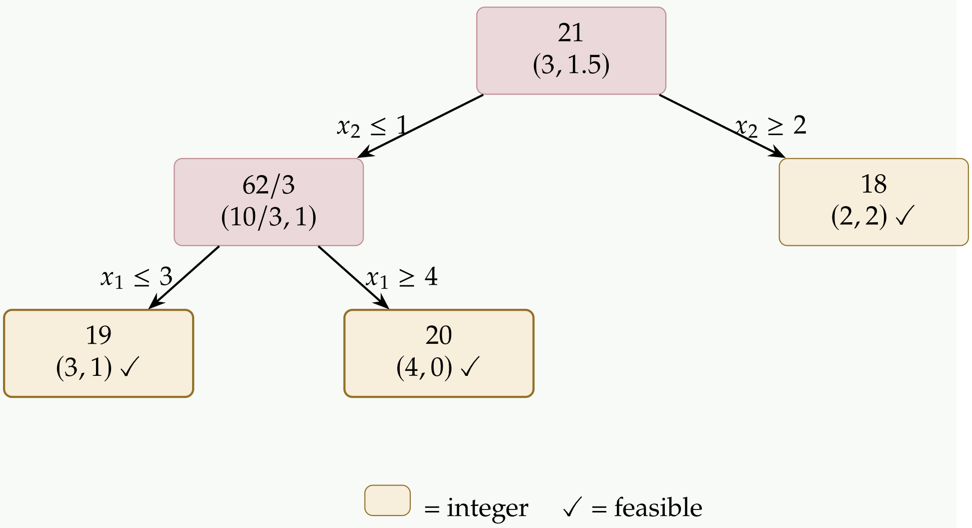
Exercise 4 (Fathoming rules). Classify three nodes by the reason for pruning.

Solution. A B&B node is pruned when its subproblem can be proven to contain no solution better than the current incumbent (or no solution at all).

1. The LP relaxation is infeasible, meaning the constraints are contradictory. Since the integer subproblem adds even more restrictions, it too is infeasible — prune by infeasibility, discarding the whole subtree.
2. The LP bound 18.3 is the best possible objective achievable at this node. The incumbent already has value 22, which is larger. Because we are maximising, no descendant can beat 22; prune by bound.
3. The LP optimum $(3, 0, 2)$ is integer and therefore feasible for the original ILP. Its value 15 exceeds the current incumbent 12, so it becomes the new incumbent. Once the node is solved, no further exploration is needed — prune by integer feasibility.

Exercise 5 (B&B trace — Instance I). Solve $\max 5x_1 + 4x_2$ by Branch and Bound.

Solution. The root optimum is $(3, 3/2)$, with bound 21. Branch on x_2. For $x_2 \geq 2$, the optimum is the integer point $(2, 2)$, value 18. For $x_2 \leq 1$, the LP optimum is $(10/3, 1)$, bound $62/3$; branching on x_1 gives the integer leaves $(3, 1)$, value 19, and $(4, 0)$, value 20. Hence the optimum is $(4, 0)$.



Exercise 6 (B&B trace — Instance II). Apply Branch and Bound to the second two-variable ILP.

Solution. Step 1 — root node. Solve the LP relaxation:

$$\max\ 3x_1 + 7x_2 \quad \text{s.t.} \quad x_1 + 4x_2 \leq 9,$$
$$3x_1 + 2x_2 \leq 10, \quad x_1, x_2 \geq 0.$$

From $x_1 + 4x_2 = 9$ we have $x_1 = 9 - 4x_2$. Substituting into $3x_1 + 2x_2 = 10$ gives $3(9 - 4x_2) + 2x_2 = 10$, so $-10x_2 = -17$ and $x_2 = 17/10$. Then $x_1 = 9 - 4(17/10) = 11/5$. Hence

$$x_{\text{root}} = (11/5,\ 17/10), \qquad z_{\text{LP}} = 3(11/5) + 7(17/10) = 37/2 = 18.5.$$

Both variables are fractional; pick x_2 (fractional part 0.7).

Step 2 — branch $x_2 \leq 1$. With $x_2 \leq 1$, the second constraint $3x_1 + 2x_2 \leq 10$ becomes $3x_1 \leq 8$, so $x_1 \leq 8/3$. The first constraint $x_1 + 4x_2 \leq 9$ becomes $x_1 \leq 5$, which is looser. Thus the LP optimum is $x_1 = 8/3,\ x_2 = 1$:

$$z = 3(8/3) + 7(1) = 8 + 7 = 15.$$

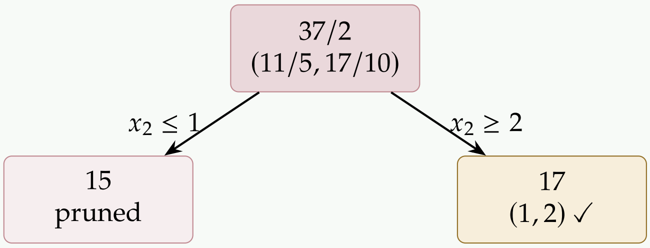
The solution is fractional $(8/3 = 2.66\ldots)$, so this node remains open for now.

Step 3 — branch $x_2 \geq 2$. From $x_1 + 4x_2 \leq 9$ we get $x_1 \leq 9 - 8 = 1$; from $3x_1 + 2x_2 \leq 10$ we get $3x_1 \leq 6$, i.e. $x_1 \leq 2$. Hence the tightest bound is $x_1 \leq 1$. The best feasible point is $(1, 2)$, which is integer:

$$z = 3(1) + 7(2) = 17.$$

This becomes the incumbent (best integer solution found so far).

Step 4 — prune. The open node $x_2 \le 1$ has LP bound 15, which is *worse* than the incumbent value 17 (maximisation). No descendant can exceed 15, so the node is pruned by bound. All nodes are closed; the incumbent $(1, 2)$ with value 17 is globally optimal.



Exercise 7 (B&B trace — Instance III). Trace Branch and Bound for the third instance.

Solution. Step 1 — root node. The ILP is

$$\max\ 2x_1 + 3x_2 \quad \text{s.t.} \quad 4x_1 + 6x_2 \le 13,\ 2x_1 + x_2 \le 5,\ x_1, x_2 \ge 0,\ x_1, x_2 \in \mathbb{Z}.$$

Dividing the first constraint by 2 gives $2x_1 + 3x_2 \le 13/2$, so the objective is bounded above by $13/2$ regardless of the second constraint. The LP optimum occurs at a vertex. Setting $x_1 = 0$ makes the first constraint active: $6x_2 = 13$, so $x_2 = 13/6$. This point also satisfies the second constraint ($2 \cdot 0 + 13/6 = 2.16\ldots \le 5$). Hence an optimal BFS is

$$x_{\text{root}} = (0, 13/6), \qquad z_{\text{LP}} = 3(13/6) = 13/2.$$

Only x_2 is fractional; branch on it.

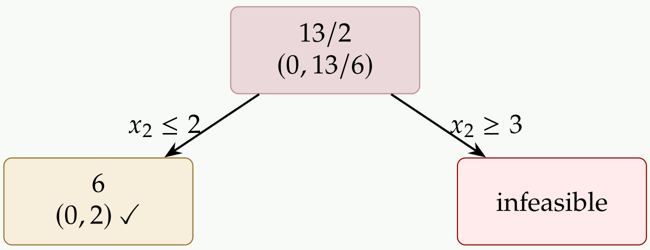
Step 2 — branch $x_2 \le 2$. With $x_2 = 2$ the first constraint gives $4x_1 + 12 \le 13$, so $x_1 \le 1/4$; the second gives $2x_1 + 2 \le 5$, so $x_1 \le 3/2$. The tightest is $x_1 \le 1/4$, forcing $x_1 = 0$ (integrality). The point $(0, 2)$ is integer:

$$z = 2(0) + 3(2) = 6.$$

This becomes the incumbent.

Step 3 — branch $x_2 \ge 3$. From the first constraint, $4x_1 + 18 \le 13$ implies $4x_1 \le -5$, which is impossible for $x_1 \ge 0$. Hence the LP relaxation is infeasible; prune by infeasibility.

The only leaf with a feasible integer point is $(0, 2)$ with value 6; it is optimal.



Exercise 8 (B&B trace — Instance IV). Solve the minimization instance, remembering that LP values are lower bounds.

Solution. Step 1 — root node. The ILP is

$$\min\ 4x_1 + 3x_2 \quad \text{s.t.} \quad 2x_1 + x_2 \geq 5,\ x_1 + 2x_2 \geq 5,\ x_1, x_2 \geq 0,\ x_1, x_2 \in \mathbb{Z}.$$

Solving the two equalities gives $x_1 = x_2 = 5/3$:

$$x_{\text{root}} = (5/3, 5/3), \qquad z_{\text{LP}} = 4(5/3) + 3(5/3) = 35/3 \approx 11.67.$$

This is a *lower* bound (minimisation). Both variables are fractional; branch on x_1.

Step 2 — branch $x_1 \leq 1$. Setting $x_1 = 1$, the constraints become $2 + x_2 \geq 5$ and $1 + 2x_2 \geq 5$, so $x_2 \geq 3$ and $x_2 \geq 2$. Hence $x_2 = 3$ is forced. The point $(1, 3)$ is integer:

$$z = 4(1) + 3(3) = 13.$$

This becomes the incumbent.

Step 3 — branch $x_1 \geq 2$. The LP relaxation minimises over $x_1 \geq 2$, $x_1 + 2x_2 \geq 5$, $2x_1 + x_2 \geq 5$. At equality of the last two constraints we get $x_2 = 3/2$, which satisfies $x_1 \geq 2$:

$$x = (2, 3/2), \qquad z = 4(2) + 3(3/2) = 25/2 = 12.5.$$

The bound $12.5 < 13$ (below the incumbent), so the node stays open. Branch on x_2.

Step 4a — branch $x_1 \geq 2$, $x_2 \leq 1$. From $x_1 + 2x_2 \geq 5$ and $x_2 \leq 1$ we get $x_1 \geq 3$. The LP optimum is $(3, 1)$:

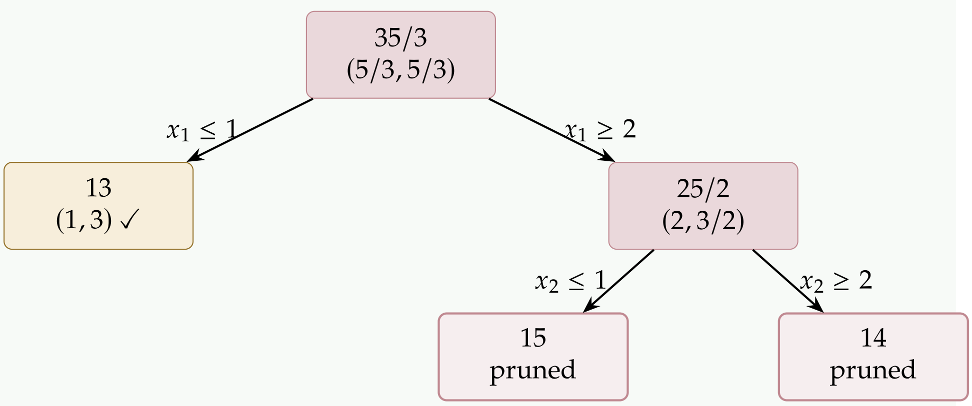
$$z = 4(3) + 3(1) = 15.$$

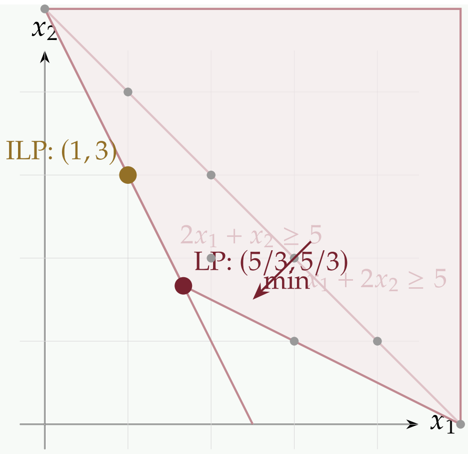
Since $15 \geq 13$ (incumbent), the node cannot beat the incumbent: prune by bound.

Step 4b — branch $x_1 \geq 2$, $x_2 \geq 2$. The point $(2, 2)$ satisfies both constraints and is the LP optimum:

$$z = 4(2) + 3(2) = 14.$$

Again $14 \geq 13$, so prune by bound. All nodes are closed; $(1, 3)$ with value 13 is optimal.





Exercise 9 (Classify nodes in a given B&B tree). Classify $N_1, \ldots, N_4$ with incumbent 14.

Solution. The incumbent is 14 (maximisation). Each node is examined against the three pruning rules:

1. N_1 (bound 17) has a fractional solution $(3.0, 1.75)$. Since $17 > 14$, the node could still contain a better integer point — it *remains open* and must be branched further.
2. N_2 is infeasible — prune by infeasibility. The constraint $x_1 \geq 4$ added at this node contradicts the original bounds, so the subproblem has no solution at all.
3. N_3 yields the integer point $(3, 1)$ with value 14.8. This is better than the current incumbent 14, so it *updates the incumbent* to 14.8 and the node is pruned by integer feasibility.
4. N_4 has bound 13.5, which is *below* the original incumbent 14. Even though its solution $(2, 2)$ is integer, the bound was already too low when the incumbent was still 14; it is pruned by bound.

After closing N_3 and N_4, the only remaining open node is N_1. If its two children are processed and both are pruned (e.g. by bound or infeasibility), no open node has a bound exceeding the incumbent, and the search terminates. In particular, an integer optimum of value 15 is impossible because every open-node bound is at most 17 and no node with bound 15 or higher remains unexplored after the children of N_1 are closed.

Exercise 10 (Fathoming by bound vs. integer feasibility). Identify the correct rule in four maximization scenarios.

Solution. Recall the pruning rules for a *maximisation* B&B tree:

- **Prune by bound:** the LP value $\leq$ incumbent — no descendant can improve.
- **Prune by integer feasibility:** the LP solution is integer and feasible — it becomes the new incumbent.
- **Do not prune:** the LP solution is fractional and its bound exceeds the incumbent, so a better integer point may exist deeper in the subtree.

1. LP value 22, incumbent 23. Since $22 < 23$, the node cannot contain a better solution regardless of integrality — prune by bound.
2. LP value 22.5, incumbent 23. Same logic: $22.5 < 23$ prunes by bound,

even though the value is closer.

3. LP value 25 with an integer solution. The value exceeds the current incumbent (say, below 25), so the node becomes the new incumbent (25), then prune by integer feasibility — the subproblem is solved to optimality.
4. LP value above 23 (say, 24.8) with a fractional solution. The node is promising: its bound is larger than the incumbent, so a better integer point might be found deeper. Do not prune; branch or add cuts.

The first three nodes are terminal — their subtrees require no further work. The fourth is active and must be explored.

Exercise 11 (Most-fractional vs. nearest-to-integer branching). Compare two branching rules at $(1.1, 2.5, 3.9, 0.5)$.

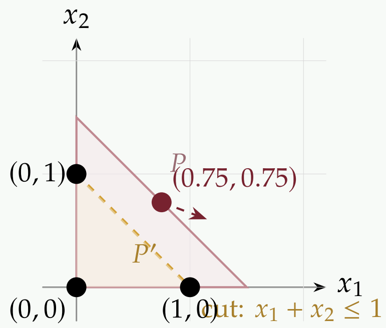
Solution. Most-fractional chooses x_2 or x_4, both with fractional part 0.5. Nearest-to-integer chooses x_1 or x_3, at distance 0.1 from an integer. A split at 0.5 tends to divide the relaxation more evenly; a split near an integer may make one child very small or infeasible. For example, $(1.1, 2.5)$ makes the two rules choose x_2 and x_1, respectively.

Exercise 12 (Best-first vs. depth-first search). Choose the next open node and compare memory and incumbent behavior.

Solution. Best-first explores the node of bound 22.7. Depth-first explores the deepest node, here the one of bound 19.8; it is not pruned because $19.8 > 18$. Depth-first normally stores fewer nodes and often finds a complete integer solution quickly. Best-first improves the global bound aggressively, but need not find an incumbent sooner.

Exercise 13 (Valid cuts do not remove integer points). Check $x_1 + x_2 \le 1$ and define separation.

Solution. The nonnegative integer points satisfying $x_1 + x_2 \le 1.5$ are $(0, 0), (1, 0), (0, 1)$; all satisfy $x_1 + x_2 \le 1$. Every fractional point with sum 1.5, including $(0.75, 0.75)$, is removed. The *separation problem* asks whether a given point violates an inequality from a chosen valid family and, if so, returns one. Validity is essential because a cut that removes any integer-feasible point might remove the true optimum.



Exercise 14 (Chvátal–Gomory cut — Example I). Apply multiplier 1/2 to

$2x_1 + 3x_2 \le 7$.

Solution. Scaling gives $x_1 + \frac{3}{2}x_2 \le \frac{7}{2}$. For nonnegative integer variables, flooring yields

$$x_1 + x_2 \le 3.$$

The point $(0, 2)$ satisfies it. The final request in the exercise is incorrect: $(0, 7/3)$ also satisfies $x_1 + x_2 \le 3$, so this particular cut does *not* remove that LP optimum. It does remove, for example, $(7/2, 0)$.

Exercise 15 (Chvátal–Gomory cut — Example II). Combine the constraints with multipliers 2/3 and 1/3.

Solution. The combination is

$$\frac{4}{3}x_1 + \frac{1}{3}x_2 \le \frac{14}{3}.$$

Flooring gives the valid, though weak, inequality $x_1 \le 4$. The point $(3, 2)$ satisfies both original constraints and also $3 \le 4$.

Exercise 16 (Chvátal–Gomory cut — Example III). Derive and compare the two specified cuts.

Solution. The first multiplier gives

$$x_1 + \frac{5}{3}x_2 \le \frac{11}{3} \quad \Longrightarrow \quad x_1 + x_2 \le 3.$$

The second gives $\frac{2}{3}x_1 + \frac{2}{3}x_2 \le 2$, whose coefficient-wise flooring is the trivial inequality $0 \le 2$. Hence the first cut is strictly tighter.

Exercise 17 (Chvátal–Gomory cut — Example IV). Apply the requested multiplier to the binary knapsack constraint.

Solution. Multiplication by 1/7 gives

$$x_1 + \frac{4}{7}x_2 + \frac{3}{7}x_3 \le \frac{10}{7},$$

and flooring gives only $x_1 \le 1$, already implied by binarity. It is valid for every feasible binary point but is not a cover inequality. A genuine cover is $x_1 + x_2 \le 1$, because weights $7 + 4$ exceed capacity 10. Thus the multiplier and the "cover" conclusion stated in the exercise are not mutually consistent.

Exercise 18 (Gomory cut from simplex tableau — Instance I). Derive the cut from the row for x_1.

Solution.

$$f_0 = \frac{3}{4}, \qquad \left\{\frac{3}{4}\right\} = \frac{3}{4}, \qquad \left\{-\frac{1}{4}\right\} = \frac{3}{4}.$$

The cut is

$$\frac{3}{4}s_1 + \frac{3}{4}s_2 \geq \frac{3}{4}, \quad \text{or} \quad s_1 + s_2 \geq 1.$$

The current BFS has left side zero and is cut off. The proposed point $x_1 = 2, s_1 = s_2 = 0$ neither satisfies the cut nor the tableau row $2 = 7/4$; consequently the last requested verification is impossible.

Exercise 19 (Gomory cut from simplex tableau — Instance II). Derive cuts from both fractional rows.

Solution. Both $x_1 = 5/2$ and $x_2 = 7/4$ are fractional. Their cuts are

$$\frac{1}{2}s_1 + \frac{3}{4}s_2 \geq \frac{1}{2}, \qquad \frac{1}{4}s_1 + \frac{3}{4}s_2 \geq \frac{3}{4}.$$

Writing the second as

$$-\frac{1}{4}s_1 - \frac{3}{4}s_2 + s_4 = -\frac{3}{4}$$

shows why the newly basic slack has RHS $-3/4$: the extended basis is primal infeasible and is repaired by dual simplex.

Exercise 20 (Gomory cut from simplex tableau — Instance III). Derive and insert the cut.

Solution. All relevant fractional parts are $2/3$, so

$$\frac{2}{3}x_1 + \frac{2}{3}s_1 \geq \frac{2}{3} \quad \Longleftrightarrow \quad x_1 + s_1 \geq 1.$$

In tableau form one may add

$$-\frac{2}{3}x_1 - \frac{2}{3}s_1 + s_{\text{new}} = -\frac{2}{3}.$$

The old reduced costs remain dual feasible, but $s_{\text{new}} = -2/3 < 0$; dual simplex is therefore the natural reoptimization method.

Exercise 21 (Gomory cut from simplex tableau — Instance IV). Compute the cut and the dual ratio test.

Solution. The coefficient fractional parts are $1/3, 2/3, 1/3$, and the RHS fractional part is $1/3$. Hence

$$\frac{1}{3}s_1 + \frac{2}{3}s_2 + \frac{1}{3}s_3 \geq \frac{1}{3}.$$

The dual ratios are

$$\frac{0.5}{1/3} = 1.5, \qquad \frac{1.2}{2/3} = 1.8, \qquad \frac{0.8}{1/3} = 2.4.$$

Thus s_1 enters.

Exercise 22 (One iteration of the cutting plane algorithm). Perform one cut for the stated ILP.

Solution. Because the first constraint is twice the objective, the LP bound is 9/2. Choose the optimal BFS (9/4, 0). Dividing its active row by 4 gives

$$x_1 + \frac{1}{2}x_2 + \frac{1}{4}s_1 = \frac{9}{4}.$$

The Gomory cut is $\frac{1}{2}x_2 + \frac{1}{4}s_1 \geq \frac{1}{4}$, which, after substituting $s_1 = 9 - 4x_1 - 2x_2$, simplifies to $x_1 \leq 2$. The new LP has an optimum $(2, 1/2)$, still of value 9/2, so it is not integer. Another fractional row must be cut (or one may branch).

Exercise 23 (Dual simplex pivot after adding a Gomory cut). Insert the supplied cut and perform one dual pivot.

Solution. Add

$$s_3 - \frac{3}{4}s_1 - \frac{1}{4}s_2 = -\frac{3}{4}.$$

The basis is primal infeasible but retains dual feasibility. Thus s_3 leaves. Both eligible ratios equal 1; choosing s_1 to enter and solving the pivot row gives

$$s_1 - \frac{4}{3}s_3 + \frac{1}{3}s_2 = 1.$$

Substitution in the other rows yields

	s_1	s_2	s_3	RHS
x_1	0	−5/12	2/3	2
x_2	0	5/6	−1/3	1
s_1	1	1/3	−4/3	1

and the objective row is updated by the same row operation. The new BFS is $x_1 = 2, x_2 = 1, s_1 = 1$. (The displayed source tableau labels its objective row $-z$ while using the opposite RHS convention; the pivot on the constraint rows is unaffected.)

Exercise 24 (Convergence of the Gomory cutting plane algorithm). State the finite-convergence idea and its practical limitation.

Solution. For a bounded pure integer program with rational data, Gomory's algorithm terminates after finitely many cuts. A cut from a fractional basic row is valid for all integer points, while the current BFS has all non-basic variables zero and violates the positive fractional RHS. In practice, many cuts can make LPs large, ill-conditioned, and slow; modern solvers therefore combine selective cuts with branching.

Exercise 25 (Branch and Cut on a small ILP). Solve the instance by adding one root cut.

Solution. The root intersection is $(7/3, 7/3)$, with value 35/3. Adding the two constraints and dividing by 3 gives

$$x_1 + x_2 \le \frac{14}{3},$$

so integrality yields the valid CG cut $x_1 + x_2 \le 4$. Reoptimization gives $(3, 1)$, value 11, which is integer and optimal. No branching is needed after the cut; without it, at least the root and its two children must be processed.

Exercise 26 (How Branch and Cut uses cuts at each node). Describe when cuts are generated and where they remain valid.

Solution. Cuts are separated after solving a node relaxation and before deciding to branch; several solve–separate rounds may occur. Local cuts exploit constraints introduced along a branch and can tighten that entire subtree. A globally valid cut may be used everywhere, whereas a cut valid only under node assumptions must remain local. Common families include Gomory mixed-integer cuts, cover cuts, clique cuts, and flow-cover cuts.

Exercise 27 (Comparing LP relaxation strength). Choose the stronger formulation and give an example.

Solution. For maximization, formulation 2 is stronger because $13 < 14.5$ while both still bound the integer optimum 12. Tighter node bounds fall below the incumbent earlier, causing more pruning. For example, the ILP

$$\max x_1 + x_2, \quad x_1 + x_2 \le \tfrac{3}{2}, \quad x \in \mathbb{Z}^2_{\ge 0}$$

has LP value $3/2$. Adding the integer-valid inequality $x_1 + x_2 \le 1$ leaves the same integer points but lowers the relaxation value to 1.

Exercise 28 (Tightening a formulation with variable bounds). Explain the effect of an added cover inequality.

Solution. The repeated constraint $x_1 \le 1$ adds nothing if the bound is already present. The improvement comes from $x_1 + x_2 \le 1$, which removes LP points satisfying the original relaxation but not the cover. It is redundant exactly when the other LP constraints already imply it. A smaller relaxation gives stronger node bounds and therefore earlier fathoming.

Exercise 29 (Effect of Big-M on the B&B tree). Compare $M = 100$ and $M = 10\,000$.

Solution. Without an objective, the numerical root bound cannot be computed. Qualitatively, the relaxation permits $y = x/M$: supporting $x = 5$ requires $y = 0.05$ for $M = 100$, but only 0.0005 for $M = 10\,000$. If y has a cost, the latter gives a much weaker bound. Use the smallest valid M, here $M = 5$ for $x \le My$, or use an indicator constraint. For two variables bounded by 5, the disaggregated constraints $x_1 \le 5y,\ x_2 \le 5y$ are stronger than merely $x_1 + x_2 \le 10y$.

Exercise 30 (True or False: Gomory cuts and integer feasibility). Assess four statements about Gomory cuts.

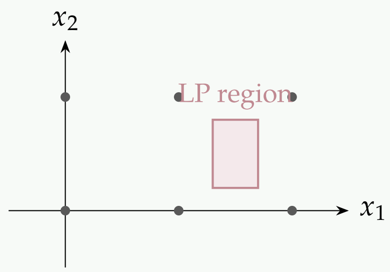
Solution.

1. False: a correctly derived Gomory cut is valid for every integer-feasible point.
2. True when "Gomory cut" means one derived from a row with fractional RHS: the current BFS sets all nonbasic variables to zero.
3. True for the pure fractional-cut construction: the basic variable and the variables used in the integrality argument must be integer restricted. Mixed-integer variants handle continuous variables differently.
4. False: several dual pivots may be required.

Exercise 31 (True or False: B&B completeness and correctness). Assess four statements about Branch and Bound.

Solution.

1. True under the usual assumptions of valid exhaustive branching and finite variable bounds (or another finite-termination condition).
2. False: an LP-feasible region can contain no integer point.

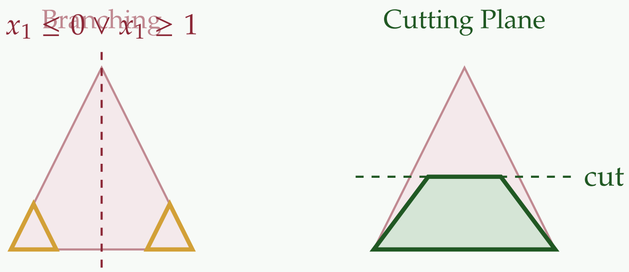


3. False as written: if a feasible integer point is found, the incumbent is feasible; for an infeasible ILP it remains absent.
4. False: worst-case running time is exponential.

Exercise 32 (True or False: Cutting planes and LP relaxations). Assess four claims about cuts.

Solution. All four claims are false. Here is why each one fails:

1. An inactive cut may become violated at a descendant node because branching constraints shift the LP optimum. Adding it at the root can therefore pay off later — it is not useless.
2. B&B partitions the feasible set by branching; cutting planes tighten the relaxation without branching. They explore different "nodes" (none, in the pure cutting plane case) and affect the relaxation in fundamentally different ways.



3. Not every valid inequality of an ILP is a *single-round* Chvátal–Gomory cut from the given constraint description. Multi-round CG cuts exist, and some inequalities (e.g. from odd-hole constraints) may require several rounds.
4. Too many cuts bloat the LP, increase solve time, and cause numerical instability. Modern solvers add cuts selectively and monitor their effectiveness.

Exercise 33 (Proof: Gomory cut is valid for all integer points). Prove validity of the fractional cut.

Solution. Separating integer and fractional parts gives

$$x_i + \sum_j \lfloor \bar{a}_{ij} \rfloor x_j - \lfloor \bar{b}_i \rfloor = f_0 - \sum_j f_j x_j.$$

At an integer point the left side, hence the right side, is an integer. If $\sum_j f_j x_j < f_0$, nonnegativity gives $0 < f_0 - \sum_j f_j x_j < 1$, which cannot be an integer. Therefore $\sum_j f_j x_j \geq f_0$. More precisely, some f_j may equal zero; they belong to $[0, 1)$, not necessarily $(0, 1)$.

Exercise 34 (Proof: Gomory cut is violated by the current LP optimum). Evaluate the cut at the current BFS.

Solution. Every nonbasic x_j is zero at the BFS, so the cut's left side is zero. Its right side is $f_0 > 0$. Thus $0 \geq f_0$ is false and the current fractional optimum is removed.

Exercise 35 (Chvátal closure and iterated cuts). Compute one closure and describe repeated closures.

Solution. With multiplier 1, the only nontrivial rounding is

$$x_1 + x_2 \leq \lfloor 3.5 \rfloor = 3.$$

For this example it already describes the integer hull together with nonnegativity. In general the first closure P' contains the integer hull but can be strictly larger. Repeated CG closures reach the integer hull after finitely many rounds for a rational polyhedron; the required number is its CG rank.

Exercise 36 (B&B on a three-item knapsack). Solve the relaxation and branch on the fractional item.

Solution. Instance: capacity $W = 8$, weights $w = (5, 3, 2)$, values $v = (7, 4, 3)$.

Step 1 — root node. Value/weight ratios are

$$\frac{v_1}{w_1} = \frac{7}{5} = 1.4, \quad \frac{v_2}{w_2} = \frac{4}{3} \approx 1.33, \quad \frac{v_3}{w_3} = \frac{3}{2} = 1.5.$$

The LP (greedy) takes items in descending ratio: item 3 (weight 2, value 3), then item 1 (weight 5, value 7), using 7 of the 8 units. The remaining 1 unit of capacity fits 1/3 of item 2:

$$x = (1,\ 1/3,\ 1), \qquad z_{\text{LP}} = 7 + \frac{4}{3} + 3 = \frac{34}{3} \approx 11.33.$$

Only x_2 is fractional; branch on it.

Step 2 — branch $x_2 = 0$. With item 2 excluded, the remaining capacity 8 fits items 3 and 1 fully:

$$(1, 0, 1), \qquad z = 7 + 3 = 10.$$

This is integer; incumbent 10.

Step 3 — branch $x_2 = 1$. Item 2 uses weight 3, leaving capacity 5. Greedy on the remaining items takes item 3 (weight 2), then 3/5 of item 1 (the best ratio among what fits):

$$x = (3/5,\ 1,\ 1), \qquad z = 7(3/5) + 4 + 3 = \frac{56}{5} = 11.2.$$

The bound $11.2 > 10$ keeps the node open. Variable x_1 is fractional; branch on it.

Step 4a — branch $x_1 = 0$. Capacity 8 with $x_2 = 1$, $x_1 = 0$ leaves weight 3 for item 3, which fits. The point $(0, 1, 1)$ is integer:

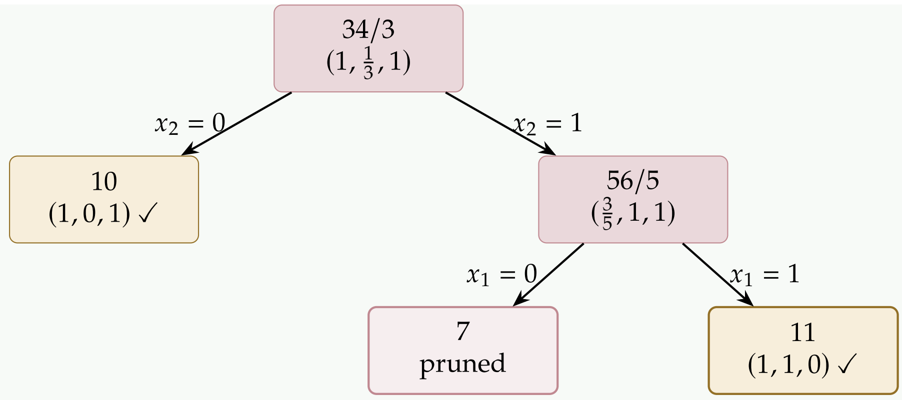
$$z = 0 + 4 + 3 = 7.$$

Since $7 < 10$ (the incumbent), prune by bound.

Step 4b — branch $x_1 = 1$. Both items 1 and 2 are taken (weight $5 + 3 = 8$), leaving no room for item 3:

$$(1, 1, 0), \qquad z = 7 + 4 = 11 > 10.$$

This becomes the new incumbent, and all nodes are closed. The optimal solution is $(1, 1, 0)$ with value 11.



Exercise 37 (Optimality gap and termination criterion). Compute the relative gap for bounds 47 and 51.3.

Solution.

$$\frac{51.3 - 47}{51.3} \cdot 100\% \approx 8.38\%.$$

This is not below 1%, so the requested stopping rule does not apply. The incumbent is feasible and is at most 4.3 objective units below the unknown optimum. Small nonzero tolerances are common because proving the last fraction of a percent may require a disproportionately large tree.

Exercise 38 (B&B for a mixed-integer program). Branch only on the integer variable y.

Solution. Step 1 — root node. The MIP is

$$\max\ 3y + x \quad \text{s.t.} \quad 2y + x \leq 7,\ y + 2x \leq 6,\ y \geq 0\ (\text{int}),\ x \geq 0\ (\text{cont}).$$

The LP relaxation (dropping integrality of y) solves at the intersection of the two constraints:

$$2y+x = 7, \quad y+2x = 6 \implies y = 8/3,\ x = 5/3, \quad z = 3(8/3)+5/3 = 29/3 \approx 9.67.$$

Only y is fractional; branch on it.

Step 2 — branch $y \leq 2$. Substituting $y = 2$: $2(2) + x \leq 7 \Rightarrow x \leq 3$, and $2 + 2x \leq 6 \Rightarrow x \leq 2$. The LP optimum is $(y, x) = (2, 2)$:

$$z = 3(2) + 2 = 8.$$

This is MIP-feasible (y integer, x continuous). Incumbent 8.

Step 3 — branch $y \geq 3$. The LP relaxation at this node is

$$\max\ 3y + x \quad \text{s.t.} \quad 2y + x \leq 7,\ y + 2x \leq 6,\ y \geq 3,\ x \geq 0.$$

From $2y+x \leq 7$ we have $x \leq 7-2y$; from $y+2x \leq 6$ we have $x \leq 3-y/2$. For $y \geq 3$ the first bound is tighter. The objective becomes $z = 3y + (7 - 2y) =$

$y + 7$, which increases with y. At $x = 0$, the constraint $2y \le 7$ gives $y \le 3.5$. Hence the LP optimum is

$$(y, x) = (7/2, 0), \qquad z = 3(7/2) = 21/2 = 10.5.$$

The bound $10.5 > 8$ (above the incumbent), so the node stays open. However $y = 3.5$ is fractional; branch again.

Step 4a — branch $y \le 3$ (child of $y \ge 3$). With $y = 3$, the tightest x is $x = 1$ (from $2 \cdot 3 + x \le 7$). The point $(3, 1)$ is MIP-feasible:

$$z = 3(3) + 1 = 10 > 8 \implies \text{new incumbent 10.}$$

Step 4b — branch $y \ge 4$ (child of $y \ge 3$). From $2y + x \le 7$ we get $8 + x \le 7$, impossible for $x \ge 0$. Infeasible — prune by infeasibility. All nodes closed; the MIP optimum is $(y, x) = (3, 1)$ with value 10.

Exercise 39 (Infeasible node in B&B). Explain infeasibility and pruning.

Solution. The inequalities require the same quantity $x_1 + x_2$ to be at most 3 and at least 4, so even the LP is infeasible. Likewise, a child requiring $x_1 \ge 4$ contradicts a parent bound $x_1 \le 3$. Since every descendant only adds constraints, an infeasible node can never acquire a feasible descendant and its whole subtree is discarded.

Exercise 40 (Warm-start with dual simplex in B&B). Explain reoptimization after adding a branch constraint.

Solution. The parent's optimal basis already satisfies dual optimality conditions, but a new row can make its basic slack negative. Dual simplex restores primal feasibility while preserving dual feasibility. The same applies to $x_1 \ge 3$ provided the new row can be represented with a basic slack and the old reduced costs remain dual feasible; no phase-I artificial variable is then needed. Warm starts are crucial because child LPs differ from their parent by very little and usually require only a few pivots.

Exercise 41 (LP bound quality and the integrality gap). Compute and interpret the ratio gap.

Solution. For $x_1 + x_2 \le 1.5$, the LP value is 1.5 and the integer value is 1, so IG = 1.5. It is always at least one for a maximization relaxation with positive integer optimum because the LP optimizes over a superset. A zero-gap example is $x_1 + x_2 \le 2$; more generally, an integral relaxation, often obtained from a totally unimodular matrix and integral RHS, ensures gap one.

Exercise 42 (Design a Branch and Cut strategy for a set-cover ILP). Outline a practical set-cover strategy.

Solution. Relax $x \in \{0, 1\}^n$ to $0 \le x \le 1$. Fractional values can share the burden of covering a row, so many x_j may be near $1/2$. At the root, separate inexpensive cover, clique, and odd-cycle-type cuts until improvement stalls. Then branch on a fractional variable, commonly one near $1/2$, reoptimize, separate local or global cuts, and prune by infeasibility, bound, or integrality. Node cuts tighten local bounds and can replace many levels of branching.

Exercise 43 (Sensitivity of B&B to objective perturbation). Resolve Instance I with objective $4x_1 + 5x_2$.

Solution. The root constraints still intersect at $(3, 3/2)$, now with value $39/2$; the optimal LP basis is unchanged. Branching on x_2 gives the integer point $(2, 2)$, value 18, in the upper branch. The lower branch has bound $55/3$ at $(10/3, 1)$; branching on x_1 gives values 17 and 16. Hence $(2, 2)$ is optimal. The tree has the same basic topology, but bounds and incumbent pruning differ.

Exercise 44 (Strengthening Gomory cuts). Apply the two stated formulas to $f_0 = 3/4$, $f = 1/4$.

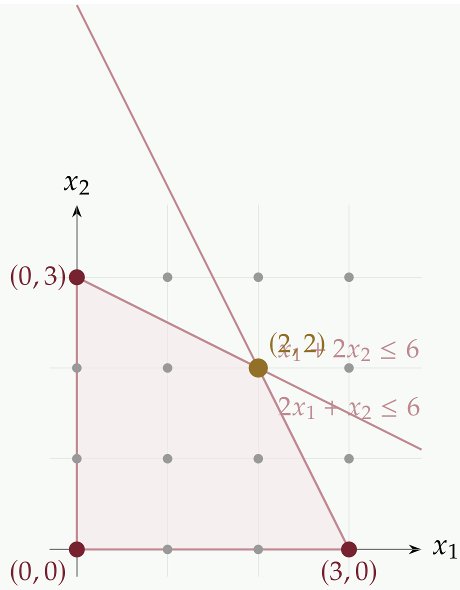
Solution. The standard fractional cut is $fy \geq f_0$, hence $\frac{1}{4}y \geq \frac{3}{4}$, or $y \geq 3$. Since $f \leq f_0$, the formula printed for the mixed-integer cut gives exactly the same inequality. Thus neither is tighter for the supplied numbers; both prove that the row has no feasible binary completion. Indeed $y = 0$ or 1 makes $x_i = 7/4 - (5/4)y$ noninteger. A comparison showing strict strengthening would require different data or the full MIR formula with continuous variables.

Exercise 45 (Enumerate all integer points in an LP feasible region). List the lattice points and optimize $2x_1 + 3x_2$.

Solution. The vertices are $(0, 0), (3, 0), (2, 2), (0, 3)$. The integer points are

x_1	x_2
0	0, 1, 2, 3
1	0, 1, 2
2	0, 1, 2
3	0

The maximum is attained at $(2, 2)$, with value 10. Evaluating the LP vertices gives the same optimum, so the additive gap is 0 and the ratio gap is 1.



Exercise 46 (Correctness of the B&B upper bound update). Justify the maximum open-node bound.

Solution. Initially every integer solution lies in the root relaxation. A branch partitions all integer points of its parent among its children. Therefore, until a point is either explicitly evaluated or excluded by a valid fathoming rule, it belongs to some open node. That node's LP value bounds its objective from above. Taking the maximum over all open-node bounds therefore bounds every still-possible integer solution.

Exercise 47 (Chvátal rank). Determine the rank of $x_1 + x_2 \le 1.5$.

Solution. One CG rounding gives $x_1 + x_2 \le 1$, which together with nonnegativity is exactly the integer hull; the rank is 1. There are families of parity and knapsack-type 0-1 polytopes whose rank grows linearly with dimension. High rank means that many solve–cut rounds may be necessary, making pure cutting planes unattractive.

Exercise 48 (Separation oracle concept). Describe tableau separation and its complexity.

Solution. Inspect each basic variable required to be integer. If its tableau RHS is fractional, form the fractional-parts inequality; it is automatically violated by the current BFS. Efficient separation matters because a solver may call it at thousands of nodes and cannot enumerate an exponential cut family. A polynomial-time oracle for *all* valid inequalities is not generally available: such an oracle would permit optimization over the integer hull and would solve NP-hard ILPs in polynomial time.

Exercise 49 (Node selection and the B&B lower bound). Order three nodes and update the incumbent.

Solution. A lower-bound node can still lead quickly to an integer solution, whereas the largest-bound node may remain highly fractional; depth and heuristics therefore matter as well as the bound. Best-first orders the nodes $21.5, 19.3, 18.0$. If the first yields an integer value 21, the incumbent becomes 21, and both remaining nodes are pruned because their upper bounds are below it.

Exercise 50 (Cutting plane algorithm — two iterations). Apply cuts to the two-constraint ILP.

Solution. The root intersection is

$$(x_1, x_2) = \left(\frac{9}{7}, \frac{11}{7}\right), \qquad z = \frac{20}{7}.$$

Multipliers $2/7$ and $1/7$ combine the two inequalities into

$$x_1 + x_2 \leq \frac{20}{7},$$

whose CG rounding is $x_1 + x_2 \leq 2$. After adding it, the LP optimum value is 2, and an optimal integer BFS is $(0, 2)$ (also $(1, 1)$ is feasible). Thus the algorithm already terminates after the first cut; a second cut is unnecessary. The request for two nontrivial iterations does not match this instance.

Exercise 51 (Comparing B&B and cutting plane efficiency). Explain why node and cut counts are not directly comparable.

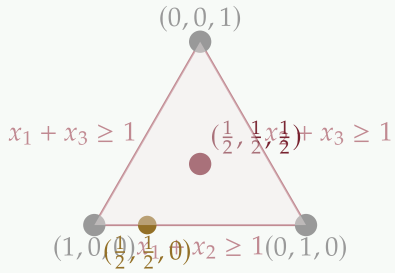
Solution. A node count hides LP sizes, simplex iterations, warm starts, and heuristic work; a cut count hides separation cost, density, and reoptimization cost. B&B is attractive when branching quickly fixes variables and node LPs remain cheap. Cuts are attractive when a few inequalities close a large root gap. Branch and Cut combines both: useful cuts improve bounds, while branching supplies robust finite progress when separation stalls.

Exercise 52 (LP relaxation of a binary ILP). Define the relaxation and a half-integral example.

Solution. Replace every $x_i \in \{0, 1\}$ by $0 \leq x_i \leq 1$. A standard half-integral example is the triangle vertex-cover relaxation

$$\min x_1 + x_2 + x_3, \qquad x_i + x_j \geq 1 \quad \text{for all three edges.}$$

Its optimum is $(1/2, 1/2, 1/2)$. A half-integer solution has every component in $\{0, 1/2, 1\}$; such values often arise from the structure of covering and matching matrices. Most-fractional branching may choose any component equal to $1/2$.



Exercise 53 (Gomory cut for a pure binary program). Derive the cut and inspect all binary completions.

Solution. The cut is

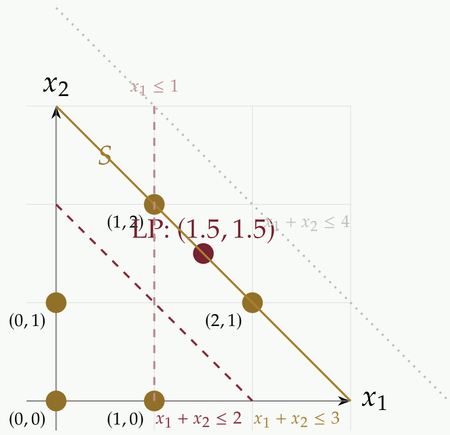
$$\frac{1}{2}x_2 + \frac{3}{4}x_3 \geq \frac{1}{4}.$$

For $(x_2, x_3) = (0,0), (1,0), (0,1), (1,1)$, the corresponding x_1 values are $5/4, 3/4, 1/2, 0$. Only the last is integer and satisfies the original row; it also satisfies the cut. The current BFS $(x_1, x_2, x_3) = (5/4, 0, 0)$ is not integer and violates the cut.

Exercise 54 (Recognise valid and invalid cuts). Classify four proposed inequalities.

Solution.

cut	valid	useful	reason
$x_1 + x_2 \leq 3$	yes	no	maximum integer sum and LP sum are both 3
$x_1 + x_2 \leq 2$	no	yes	removes $(2,1), (1,2)$ and the LP point
$x_1 \leq 1$	no	yes	removes $(2,1)$ and the LP point
$x_1 + x_2 \leq 4$	yes	no	the LP point satisfies it

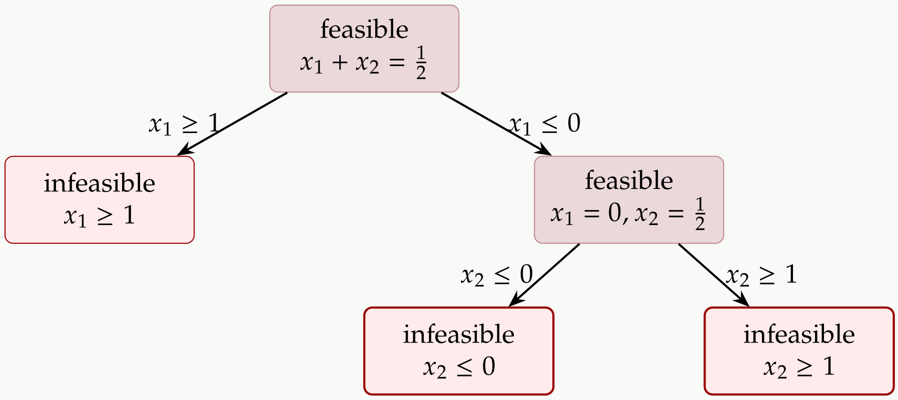


A useful inequality that is invalid is not a legitimate cut.

Exercise 55 (ILP infeasibility certificate via B&B). Interpret a tree with no integer incumbent.

Solution. If exhaustive valid B&B closes every node without an incumbent, the ILP is infeasible. LP-infeasible leaves do not imply that the root LP was infeasible: branching may be needed to expose the absence of lattice points. With incumbent $-\infty$, no finite node can genuinely be fathomed by bound, so such a log would indicate an error.

For example, let $x_1, x_2 \in \mathbb{Z}_{\geq 0}$ and $x_1 + x_2 = 1/2$. The root LP is feasible. Branching on x_1 makes $x_1 \geq 1$ infeasible; the branch $x_1 \leq 0$ fixes $x_1 = 0$ and $x_2 = 1/2$. Branching on x_2 then makes both $x_2 \leq 0$ and $x_2 \geq 1$ infeasible, certifying integer infeasibility.



CHAPTER 7

TUM & Integral Polyhedra

Exercise 1 (TU check: a 2×3 matrix). Check all square subdeterminants of A.

Solution. The 1×1 minors are the entries $1, 0, -1, 0, 1, 1$. Choosing column pairs $(1, 2), (1, 3), (2, 3)$ gives determinants

$$1, \qquad 1, \qquad 1.$$

Every square minor belongs to $\{-1, 0, 1\}$, so A is TU.

Exercise 2 (TU check: a matrix with a violation). Find a forbidden minor of B.

Solution. The three 2×2 determinants are

$$\det B_{\{1,2\}} = -2, \qquad \det B_{\{1,3\}} = 1, \qquad \det B_{\{2,3\}} = 1.$$

The first already violates total unimodularity; hence B is not TU.

Exercise 3 (TU check: another 2×3 matrix). Determine whether C is TU.

Solution. All entries are $0, \pm 1$, and the 2×2 minors for the three column pairs are $1, -1, 1$. Therefore C is TU.

Exercise 4 (TU check: a 3×3 identity-like matrix). Compute the minors of D.

Solution. $\det D = -1$. The nine 2×2 minors, listed row-pair by row-pair, are

$$(1, 0, 0), \qquad (1, -1, 0), \qquad (-1, 0, 0).$$

Together with the entries, all determinants are $0, \pm 1$; D is TU.

Exercise 5 (TU check: off-diagonal entries). Test the matrix E.

Solution. The rows sum to zero, so $\det E = 0$. Its 2×2 minors are

$$1, 1, 1, \quad 1, -1, -1, \quad 1, 1, 1$$

up to their row/column ordering. All entries and minors are $0, \pm 1$; therefore E is TU.

Exercise 6 (Non-TU 3×3 matrix). Exhibit a forbidden submatrix of F.

Solution. The 1×1 submatrix containing $f_{11} = 2$ already has determinant 2. Thus no larger calculation is needed: F is not TU.

Exercise 7 (Incidence matrix of a directed path). Write the incidence matrix and verify its structure.

Solution. Using $+1$ at an arc's tail and -1 at its head,

$$M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}.$$

Its four rows sum to zero, so $\det M = 0$. Every two-column restriction has at most one $+1$ and one -1, which makes every 2×2 minor $0, \pm 1$. More generally, every directed node–arc incidence matrix is TU.

Exercise 8 (Incidence matrix of a directed triangle). Write and analyze the directed-cycle matrix.

Solution.

$$M = \begin{pmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}.$$

The rows sum to zero, hence $\det M = 0$ and $\operatorname{rank} M = 2$. It is TU by the directed-incidence theorem; singularity does not conflict with TU.

Exercise 9 (Applying TUM to a min-cost flow LP). Explain integrality of bounded network flow.

Solution. B is TU because every arc column has one $+1$ and one -1. Appending I and $-I$ for $0 \le x \le u$ preserves TU. With integer b, u, the resulting polyhedron has integer vertices, so a bounded feasible LP has an integer optimum. Dropping $x \le u$ does not harm integrality; it only changes boundedness.

Exercise 10 (Bipartite graph: incidence matrix is TU). Analyze the incidence matrix of $K_{2,2}$.

Solution. Ordering rows u_1, u_2, w_1, w_2 and columns as in the exercise,

$$N = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

Any 2×2 binary minor is $0, \pm 1$. Put the u-rows in R_1 and the w-rows in R_2: each column has one 1 in each set, so its signed sum is zero. Ghouila-Houri gives TU.

Exercise 11 (Non-bipartite incidence matrix). Test the undirected triangle.

Solution.

$$N = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \qquad \det N = 2.$$

Therefore N is not TU. In general, an undirected vertex–edge incidence matrix is TU exactly when the graph is bipartite.

Exercise 12 (Odd cycle implies non-TU). Apply the bipartite criterion to C_5.

Solution. With cyclic edge order,

$$N = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

C_5 is odd and hence not bipartite, so N is not TU. In fact the full displayed 5×5 square submatrix has determinant 2. The hint suggesting a 3×3 witness is misleading: the natural witness for a chordless C_5 is the whole cycle matrix.

Exercise 13 (Ghouila-Houri on a network matrix). Find a row signing for the network matrix.

Solution. Take R_1 to be all rows and $R_2 = \varnothing$. Every column contains one $+1$ and one -1, so every signed column sum is zero. Ghouila-Houri says that a $0, \pm 1$ matrix is TU when every row subset can be signed so its column sums lie in $\{-1, 0, 1\}$; the same all-positive signing works for every restricted row set of a directed incidence matrix.

Exercise 14 (Applying Ghouila-Houri to a $0/1$ matrix). Find a valid partition for M.

Solution. Put rows $1, 3$ in R_1 and rows $2, 4$ in R_2. Each column has one 1 in each set, so every signed sum is zero. Thus M is TU: it is the incidence matrix of the bipartite cycle C_4. Consequently the exercise's fallback request for a determinant of magnitude 2 does not occur.

Exercise 15 (Ghouila-Houri: constructive verification). Prove TU from the column sign pattern.

Solution. With all rows in R_1, a column sum is $0, +1$, or -1, because it has at most one entry of either sign. Hence Ghouila-Houri applies. For example,

$$\begin{pmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & -1 & -1 \end{pmatrix}$$

has this pattern and is TU.

Exercise 16 (True/False: basic TUM statements). Classify five elementary claims.

Solution.
1. False: the triangle incidence matrix has determinant 2.
2. False: nonzero entries of $2M$ equal ± 2.
3. True: corresponding minors have equal determinants.
4. True: every square minor of a submatrix is also a minor of M.
5. True: its determinant is 0, and its entries are 1.

Exercise 17 (True/False: TUM and operations). Classify five closure claims.

Solution. Statements 1, 2, 4, and 5 are true by zero-column closure, sign closure, the TU integrality theorem, and transpose closure. Statement 3 is false:

$$A = (1 \;\; 1), \qquad B = (1 \;\; -1)$$

are TU, but stacking them gives determinant -2.

Exercise 18 (True/False: integral polyhedra). Separate matrix properties from properties of one RHS.

Solution.
1. False: integrality for one b does not force A to be TU.
2. True for a rational polyhedron when the statement quantifies over all integer objectives with finite optimum; a fractional vertex can be exposed by an integer objective.
3. True, provided the feasible LP has a finite optimum.
4. False: $2x \le 3,\ 0 \le x \le 1$ has fractional vertex $3/2$ without the upper bound, and ordinary knapsacks likewise have fractional vertices.

Exercise 19 (Integral vertices via TUM). Reconstruct the basis proof.

Solution. Choose n independent active rows among $Ax \le b$ and $-Ix \le 0$. They form $\widehat{A}x = \widehat{b}$. Since stacking $-I$ preserves TU, the nonsingular basis satisfies $\det \widehat{A} = \pm 1$. Cramer's rule gives

$$x_j = \frac{\det(\widehat{A}_j)}{\det(\widehat{A})} \in \mathbb{Z},$$

because $\widehat{A}_j$ has integer entries. Every vertex is integral.

Exercise 20 (Integral polyhedron: small example). Check the claimed TU explanation and list the vertices.

Solution. Here

$$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad \det A = -2,$$

so A is *not* TU; the first request in the exercise is false. Nevertheless this particular polyhedron is integral. Its vertices are

$$(0,0), \quad (1,0), \quad (2,1), \quad (0,3).$$

This illustrates that TU is sufficient for integrality for every integer RHS, not necessary for one particular polyhedron.

Exercise 21 (When does LP equal ILP?). State the exact consequence of TU.

Solution. "For free" means equal feasible optimum values and the existence of an integer LP optimum, without branching or cuts. TU and integer b make every LP vertex integer, so an attained LP optimum is ILP-feasible. For example,

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad b = (3,2), \quad \max x_1 + 2x_2$$

has the integer LP optimum $(0,3)$, value 6.

Exercise 22 (Transpose preserves TU). Prove transpose closure and give an example.

Solution. Every square submatrix of $A^\top$ is $S^\top$ for a square submatrix S of A, and $\det S^\top = \det S$. Thus TU is preserved. For

$$A = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix},$$

the minors computed in Exercise 1 prove A TU, and the same values appear among the minors of $A^\top$.

Exercise 23 (Adding a zero row preserves TU). Prove the zero-row closure property.

Solution. A square minor containing the new row has determinant zero; one omitting it is already a minor of A. For example,

$$\begin{pmatrix} 1 & 0 \\ -1 & 1 \\ 0 & 0 \end{pmatrix}$$

is TU because its nonzero minors come from the first two rows.

Exercise 24 (Adding a standard basis row preserves TU). Prove the identity-row closure property.

> **Solution.** If a selected square minor contains the new row but not its unique nonzero column, its determinant is zero. Otherwise Laplace expansion along that row reduces it, up to sign, to a minor of A. Therefore appending $e_j^\top$ preserves TU. Consequently integral bounds $x_j \le u_j$ preserve integrality of a TU formulation.

Exercise 25 (Negating a row preserves TU). Prove sign closure and interpret doubled inequalities.

> **Solution.** Every minor containing the negated row changes sign; all others are unchanged. Row duplication also preserves TU, so duplicating every row and negating the copy proves $\binom{A}{-A}$ is TU. The system
>
> $$\begin{pmatrix} A \\ -A \end{pmatrix} x \le \begin{pmatrix} b \\ -\ell \end{pmatrix}$$
>
> means $\ell \le Ax \le b$.

Exercise 26 (Shortest-path LP and TU). Explain why the flow relaxation is integral.

> **Solution.** The equality matrix is a signed node–arc incidence matrix, hence TU, and b is integral. Thus an optimal vertex flow is integer. With nonnegative costs one may remove cycles and choose a $0/1$ unit flow, whose positive arcs form an s-t path. Without a no-negative-cycle assumption, "hence $0/1$" would not automatically follow from integrality alone.

Exercise 27 (Network flow: TU and integrality). Derive the integral max-flow theorem.

> **Solution.** The conservation matrix is TU; source/sink bookkeeping columns have the same network pattern, and the capacity rows append I and $-I$. Thus integral capacities give integral vertices and an integral maximum flow. This is the LP/TU form of Ford–Fulkerson's integrality theorem: integer capacities admit an integer maximum flow.

Exercise 28 (Assignment problem: LP formulation and TU). Write the $n = 2$ matrix and interpret an integer solution.

> **Solution.** For columns $x_{11}, x_{12}, x_{21}, x_{22}$,
>
> $$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$
>
> It is the vertex–edge incidence matrix of $K_{2,2}$, hence TU. With RHS $\mathbf{1}$, every vertex is integral. An integer feasible vector selects a perfect matching: exactly one job per worker and one worker per job.

Exercise 29 (Bipartite matching: LP equals ILP). Connect matching integrality to graph bipartiteness.

> **Solution.** The coefficient matrix is the vertex–edge incidence matrix of the bipartite graph, so it is TU. With integer RHS 1, an optimal LP vertex is 0/1 and its value is the maximum matching cardinality. For a non-bipartite graph an odd-cycle submatrix has determinant 2; on a triangle, $x_e = 1/2$ for all edges is the standard fractional solution.

Exercise 30 (Transportation problem: matrix and TU). Analyze the 2×2 transportation formulation.

> **Solution.** The matrix is the same $K_{2,2}$ matrix displayed in the assignment solution. Every shipment column contains one 1 in a supply row and one in a demand row. Signing supply rows + and demand rows − gives zero column sums, proving TU. Integer supplies and demands therefore admit an integer optimal shipment plan.

Exercise 31 (Knapsack is not TU). Give a fractional knapsack relaxation.

> **Solution.** $A = (3 \;\; 2)$ is not TU because its entries already include $3, 2$. Moreover
>
> $$\det \begin{pmatrix} 3 & 2 \\ 1 & 0 \end{pmatrix} = -2.$$
>
> For $3x_1 + 2x_2 \leq 4$, $x \geq 0$, and objective $\max x_1$, the LP chooses $x_1 = 4/3$, while the integer optimum is $x_1 = 1$.

Exercise 32 (General knapsack: a non-TU submatrix). Compute two augmented determinants.

> **Solution.**
>
> $$\det \begin{pmatrix} 5 & 3 \\ 1 & 0 \end{pmatrix} = -3,$$
>
> so the matrix is not TU. Taking $a_1 = a_2 = 2$ gives the explicit requested example
>
> $$\det \begin{pmatrix} 2 & 2 \\ 1 & 0 \end{pmatrix} = -2.$$

Exercise 33 (ILP solved by LP relaxation: two variables). Solve the TU relaxation.

> **Solution.**
>
> $$A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$
>
> has determinant −2, so it is not TU; the premise is again incorrect. Direct optimization still gives the intersection $(3, 1)$, value 11, which is the LP and ILP optimum. This equality is accidental for this RHS and objective, not a consequence of TU.

Exercise 34 (ILP solved by LP relaxation: assignment). Find the minimum-cost assignment.

Solution. The six-by-nine matrix is the incidence matrix of $K_{3,3}$, hence TU. The LP is

$$\min \sum_{i,j} c_{ij}x_{ij}, \quad \sum_j x_{ij} = 1, \quad \sum_i x_{ij} = 1, \quad x \geq 0.$$

Inspection gives $1 \to B$, $2 \to C$, $3 \to A$, of total cost $2 + 4 + 1 = 7$. TU proves that this assignment is also an LP optimum.

Exercise 35 (ILP: checking TU before solving). Test the triangle-type matrix.

Solution. Every 2×2 minor is $0, \pm 1$, but the full determinant is

$$\det \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} = 2.$$

Hence the matrix is not TU and the theorem cannot be invoked. The LP has the fractional optimum $(1/2, 1/2, 1/2)$, value 2, while the integer point $(0, 1, 0)$ also has value 2; equality of values here does not make the polyhedron integral.

Exercise 36 (Recognising TU: equality form). Verify the structured 3×4 matrix.

Solution. The fourth column is a standard basis column. In the first three columns, all 2×2 minors are $0, \pm 1$, and every 3×3 minor is $0, \pm 1$. Expanding any minor containing column 4 reduces to one of these smaller minors. Therefore A is TU.

Exercise 37 (Which problems are naturally integer?). Classify five standard formulations.

Solution.

problem	matrix	LP integral for integer data?
directed shortest path	node–arc incidence	yes
bipartite matching	bipartite incidence	yes
general matching	general incidence	not without blossom cuts
knapsack	arbitrary weight row	no
transportation	bipartite incidence	yes

The dividing structures are directed incidence and bipartiteness.

Exercise 38 (Column scaling does not preserve TU). Compare scaling by 2 and by -1.

Solution. Scaling the first identity column by 2 creates a 1×1 minor equal to 2, so the result is not TU. Scaling it by -1 only changes signs of minors and preserves TU. Row/column permutations, deletions, sign changes, and zero or identity augmentation are safe; general nonunit scaling is not.

Exercise 39 (TU and box constraints). Add integral lower and upper bounds.

Solution. Write
$$Ix \le u, \qquad -Ix \le -\ell.$$
The augmented matrix $\begin{pmatrix} A \\ I \\ -I \end{pmatrix}$ is TU by identity-row augmentation, duplication, and sign change. With integer b, ℓ, u, the TU integrality theorem makes every vertex of the bounded polyhedron integer.

Exercise 40 (TU versus unimodular). Distinguish the two definitions.

Solution. A square integer matrix is unimodular when its determinant is ± 1. A possibly rectangular matrix is TU when *every* square minor is $0, \pm 1$. The matrix $\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$ is unimodular but not TU. I_2 is square and TU. A square TU matrix need not be unimodular because it may be singular. Thus checking only $\det A \in \{-1, 0, 1\}$ is not sufficient for TU.

Exercise 41 (TU check: directed-arc-type matrix). Apply the column pattern and check two large minors.

Solution. Each column has one $+1$ and one -1, so putting all rows in one Ghouila-Houri set proves TU. The partition $R_1 = \{1, 3\}, R_2 = \{2\}$ printed in the exercise is not valid: column 2 would have signed sum -2. For example, the minors using columns $1, 2, 3$ and $1, 2, 4$ both have determinant 0, since the three rows sum to zero.

Exercise 42 (Rank and TU: low-rank example). Decide whether low rank is enough.

Solution. $\det A = 0$, every entry is 1, and these are all square minors, so A is TU. Its rank is 1. Low rank alone does not imply TU: $(2 \;\; 2)$ also has rank 1 but contains a minor equal to 2.

Exercise 43 (Stacking two TU matrices). Disprove unrestricted stacking.

Solution. Let
$$A = (1 \;\; 1), \qquad B = (1 \;\; -1).$$
Each one-row matrix is TU, but
$$\det \begin{pmatrix} A \\ B \end{pmatrix} = -2.$$
Hence separate TU descriptions do not guarantee that their intersection has a TU stacked matrix; integrality must be re-established for the combined system.

Exercise 44 (Max-weight bipartite matching via LP). Formulate and solve the $K_{2,3}$ instance.

Solution. Use variables $x_{ij} \geq 0$, constraints $\sum_j x_{ij} \leq 1$ for each worker and $\sum_i x_{ij} \leq 1$ for each job, and maximize $\sum_{ij} w_{ij} x_{ij}$. The matrix is the incidence matrix of $K_{2,3}$, hence TU. The optimal matching is $u_1 v_1$ and $u_2 v_2$, of weight $4 + 5 = 9$.

Exercise 45 (Network simplex and integer bases). Relate bases to spanning trees.

Solution. Delete one redundant conservation row, leaving rank $n - 1$. A set of $n - 1$ independent incidence columns contains no cycle and connects all vertices, hence is a spanning tree. Its reduced incidence matrix is a nonsingular square submatrix of a TU matrix, so its determinant is ± 1. Therefore $B_{\mathcal{B}}^{-1} b$ is integer for integer b.

Exercise 46 (Sensitivity to RHS perturbations). Compare integer and fractional perturbations.

Solution. If δ is integer, then $b + \delta$ is integer and TU still guarantees integrality. A fractional perturbation need not preserve it. The one-dimensional example

$$A = (1), \qquad x \leq \tfrac{3}{2}, \quad x \geq 0$$

has TU matrix but fractional vertex $x = 3/2$.

Exercise 47 (Directed complete graph K_3). Write and check the tournament incidence matrix.

Solution. For arcs $(1, 2), (1, 3), (2, 3)$,

$$M = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{pmatrix}.$$

$\det M = 0$; direct calculation gives only $0, \pm 1$ for all 2×2 minors, and entries are $0, \pm 1$. Thus M is TU, as also follows from the directed-incidence theorem.

Exercise 48 (Constructing an integral polyhedron). List the vertices and optimize.

Solution.

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad b = (5, 3, 4).$$

Its entries and all 2×2 minors are $0, \pm 1$, so it is TU. The vertices are

$$(0, 0), (3, 0), (3, 2), (1, 4), (0, 4).$$

For $\max 3x_1 + 2x_2$, the largest value is 13 at $(3, 2)$.

Exercise 49 (Min-cost assignment: proof from TU). Prove assignment integrality without an algorithm.

Solution. Columns are edges of $K_{n,n}$, and rows are its two vertex classes. Because $K_{n,n}$ is bipartite, its incidence matrix is TU. The RHS of all degree equalities is 1, so every assignment-polytope vertex is integral. Nonnegativity and degree one then force every coordinate to be 0 or 1. A linear objective attains an optimum at such a vertex, proving LP = ILP.

Exercise 50 (Two-commodity flow and loss of TU). Explain why coupling can destroy network integrality.

Solution. A single B is TU by the directed-incidence theorem. For the two-node, one-arc matrix specified in the exercise, however, the stacked matrix has only two columns and every 2×2 minor is $0, \pm 1$; the requested determinant > 1 does not exist. In larger coupled networks, submatrices such as

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}, \qquad \det = -2,$$

can arise from shared-capacity rows. Thus the block-diagonal network structure plus coupling is not TU in general, and integral multi-commodity routing may require ILP methods.

Exercise 51 (Recognising TU from column structure). Construct and verify a consecutive-ones example.

Solution. Take

$$A = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Every row has consecutive ones and each column has at most two. Put rows $1, 3$ in one set and row 2 in the other; columns with two ones meet both sets, so the sufficient partition condition holds. Hence all 2×2 and 3×3 minors are $0, \pm 1$, and A is TU.

Exercise 52 (Interval scheduling matrix). Write the example and establish integrality.

Solution.

$$A = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

For these rows, signing times $1, 3$ positive and time 2 negative gives column sums $0, 0, 1, 1$. More generally, in every selected row subset, alternate signs in time order; each interval column then has signed sum $0, \pm 1$. Ghouila-Houri proves that interval matrices are TU, so the LP has an integer optimum.

Exercise 53 (TU under augmentation with slacks). Prove that $[A \mid I]$ is TU.

Solution. For any square minor selecting identity columns, repeatedly expand along an identity column. The determinant becomes zero or, up to sign, a minor of A. Hence $[A \mid I]$ is TU. For

$$A = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix}, \qquad [A \mid I] = \begin{pmatrix} 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{pmatrix},$$

the same expansion makes every 2×2 minor $0, \pm 1$.

Exercise 54 (TU and the LP integrality gap). Relate total unimodularity to gap one.

Solution. With TU A and integer b, an LP optimum can be chosen at an integer vertex, so $z^*_{\text{LP}} = z^*_{\text{ILP}}$ and the ratio is 1. For a non-TU one-variable example,

$$\max x \quad \text{s.t.} \quad 2x \le 3, \quad x \in \mathbb{Z}_{\ge 0},$$

the LP and ILP values are $3/2$ and 1, giving gap $3/2$. TUM identifies formulations whose relaxation is already as tight as possible.

Exercise 55 (Summary: TUM proof roadmap). Reconstruct the integrality proof.

Solution. A vertex is the unique solution of n independent active constraints. Their coefficient matrix $\widehat{A}$, drawn from $\left(\begin{smallmatrix} A \\ -I \end{smallmatrix}\right)$, is a nonsingular TU square matrix. Thus

$$x^*_j = \frac{\det(\widehat{A}_j)}{\det(\widehat{A})}, \qquad \det(\widehat{A}) = \pm 1.$$

The numerator is an integer because the RHS is integer. Hence every component of every vertex is integer, which is exactly the definition of an integral polyhedron.

CHAPTER 8

Graph Algorithms

Exercise 1. Represent the first undirected graph.

Solution.

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

The lists are $1 : 2, 3;\ 2 : 1, 4;\ 3 : 1, 4;\ 4 : 2, 3, 5;\ 5 : 4$. Degrees are $2, 2, 2, 3, 1$, whose sum $10 = 2|E|$.

Exercise 2. Represent the directed graph and compute degrees.

Solution. In order a, b, c, d,

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The pairs (in,out) are $a : (0, 2), b : (2, 1), c : (1, 2), d : (2, 0)$. It is not strongly connected: d reaches no other vertex.

Exercise 3. Maximum size of a simple undirected graph.

Solution. Each edge is an unordered pair of distinct vertices, so

$$m \le \binom{n}{2} = \frac{n(n-1)}{2},$$

with equality for K_n.

Exercise 4. Compare sparse and dense representations.

Solution. An adjacency list uses $\Theta(n + m)$, hence is proportional to the input edges for sparse graphs. Listing one vertex's neighbors costs $\Theta(n)$

with a matrix and $\Theta(\deg v)$ with a list. Paths and trees are sparse; complete graphs are dense.

Exercise 5. True or false: elementary graph facts.

Solution. All four statements are true: a tree has $n-1$ edges; every connected graph contains a spanning tree; k-regular means degree k at every vertex; and every arc contributes once to the total out-degree.

Exercise 6. Run BFS from vertex 1.

Solution. Discovery order: $1, 2, 3, 4, 5, 6$. Tree edges:

$$\{12, 13, 24, 25, 36\}.$$

Distances are $d = (0, 1, 1, 2, 2, 2)$. The only non-tree edge is 45; it joins vertices on the same layer and is a BFS cross edge.

Exercise 7. Run directed BFS from s.

Solution. Queue evolution is

$$[s] \to [a, b] \to [b, c] \to [c, d] \to [d, e] \to [e] \to [].$$

Tree arcs are sa, sb, ac, bd, ce. Distances are $0, 1, 1, 2, 2, 3$ for s, a, b, c, d, e. The shortest paths to e are $s - a - c - e$, $s - b - c - e$, and $s - b - d - e$.

Exercise 8. Prove BFS shortest-path correctness.

Solution. BFS processes layers in nondecreasing distance. Inductively, before layer k is processed, every discovered vertex in earlier layers has true distance equal to its label. Every new neighbor has a path of length $k+1$, and no shorter path exists, since its predecessor on such a path would have appeared in an earlier processed layer. Thus every assigned label is exact.

Exercise 9. Relate BFS layers across an undirected edge.

Solution. An edge uv extends a shortest s-u path, so $d[v] \le d[u] + 1$; exchanging u, v gives $|d[u] - d[v]| \le 1$. Hence an undirected non-tree edge joins the same or adjacent layers, never layers separated by two or more.

Exercise 10. Test bipartiteness with BFS.

Solution. Assign the source color 0, then give every newly discovered neighbor the opposite color. If an explored edge has equally colored endpoints, reject; otherwise accept after all edges are scanned. Equal colors produce an odd cycle through the BFS tree, while every odd cycle forces such a conflict. Complexity is $\Theta(n + m)$.

Exercise 11. Count connected components by BFS.

Solution. Starting BFS from each still-white vertex gives

$$\{1,2,3\}, \qquad \{4,5\}, \qquad \{6,7\}.$$

There are three connected components.

Exercise 12. Analyze BFS representations.

Solution. Adjacency lists give $\Theta(n+m)$; scanning a full matrix row for each vertex gives $\Theta(n^2)$. They coincide asymptotically when $m = \Theta(n^2)$.

Exercise 13. Run undirected DFS from 1.

Solution.

v	1	2	3	4	5	6
$d[v]$	1	2	5	3	9	4
$f[v]$	12	11	6	8	10	7

Tree edges are 12, 24, 46, 63, 25. The remaining edge 13 is a back edge (viewed from descendant 3 to ancestor 1); undirected DFS has no forward or cross edges.

Exercise 14. Run directed DFS from 1.

Solution.

v	1	2	3	4	5
d	1	2	8	3	4
f	10	7	9	6	5

Tree arcs: 12, 24, 45, 13. Back arc: 52. Cross arcs: 34, 35. There are no forward arcs. The back arc certifies the cycle $2 \to 4 \to 5 \to 2$.

Exercise 15. State the white-path theorem.

Solution. At discovery of u, a vertex v becomes a descendant of u exactly when a path from u to v consists entirely of white vertices. DFS follows such a path recursively; conversely, if v is a descendant, the tree path from u to v was white when u was discovered.

Exercise 16. Can undirected DFS have cross edges?

Solution. No. When uv is inspected from the endpoint discovered first, the other endpoint is either white and becomes a descendant, or gray and is an ancestor. It cannot already belong to a disjoint finished interval, because the undirected reverse edge would have discovered it earlier.

Exercise 17. True or false: DFS facts.

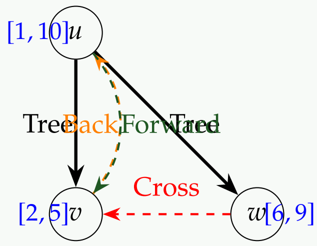
Solution.

(a) **True.** DFS explores every reachable vertex. Since the graph is connected, it reaches all n vertices. The $n-1$ discovery edges form a spanning tree.

(b) **True.** A back arc (u, v) points from a descendant u to an ancestor v. Because v is still active when u is fully explored, v's finish time must be later than u's finish time ($f[v] > f[u]$).

(c) **True.** A forward arc (u, v) points from an ancestor u to a descendant v. Since u discovered v (or a path to v), u's discovery time precedes v's ($d[u] < d[v]$). We also have $f[v] < f[u]$.

(d) **False.** Undirected graphs only have tree edges and back edges. A cross edge would imply one endpoint was fully explored before the other was discovered, but an undirected edge would have caused the first endpoint to discover the second.



Exercise 18. Prove the DFS cycle criterion.

Solution. A back arc $u \to v$ plus the tree path $v \rightsquigarrow u$ forms a cycle. Conversely, take the first discovered vertex on a directed cycle. DFS reaches around the still-white cycle until an arc returns to a gray ancestor, producing a back arc.

Exercise 19. Identify three DAGs.

Solution. (a) is a DAG, with order $1, 2, 3, 4$. (b) is not: $1 \to 2 \to 3 \to 1$. (c) is a DAG, for example in order a, b, c, d, e.

Exercise 20. Topological order implies acyclicity.

Solution. Along every arc, positions strictly increase. On a directed cycle they would have to increase around the cycle and then return to the starting position, an impossibility.

Exercise 21. Every DAG has a topological order.

Solution. If every vertex had positive in-degree, repeatedly following incoming arcs in a finite graph would revisit a vertex and create a cycle. Thus a DAG has a source. Remove it, topologically order the remaining DAG by induction, and prepend the source.

Exercise 22. Sources and sinks in a DAG.

Solution. Every nonempty finite DAG has a source and, by reversing all arcs, a sink. An isolated vertex has in-degree and out-degree zero, so it is both; in a one-vertex DAG this always happens.

Exercise 23. DFS topological sort.

Solution.

v	a	b	c	d	e	f
d	1	2	8	3	9	4
f	12	7	11	6	10	5

Decreasing finish time gives a, c, e, b, d, f. Every arc points from a larger to a smaller finish time, as required.

Exercise 24. Kahn's algorithm on the same DAG.

Solution. Initial in-degrees are $0, 1, 1, 2, 1, 2$. With alphabetical queue:

$$a \to (b, c), \quad b \to (c), \quad c \to (d, e), \quad d \to (e), \quad e \to (f), \quad f.$$

The order is a, b, c, d, e, f, different from the DFS order. Topological orders need not be unique.

Exercise 25. Detect cycles with Kahn's algorithm.

Solution. Count removed vertices. If fewer than n are removed, the remaining subgraph has no source and therefore contains a directed cycle. If all are removed, the output itself is a topological ordering and proves the graph is a DAG.

Exercise 26. Count topological orders of the six-vertex DAG.

Solution. Vertex 1 must be first and 6 last. The five admissible middle orders are

$$2345, \quad 2354, \quad 3245, \quad 3254, \quad 3524.$$

Thus there are five orders. Only 1 can be first and only 6 last.

Exercise 27. Complexity of DFS topological sort.

Solution. DFS initializes and finishes each vertex once and scans every adjacency entry once, costing $\Theta(n + m)$. Prepending a vertex on finish is $O(1)$, so producing the order adds only $O(n)$.

Exercise 28. Shortest paths in the first weighted DAG.

Solution. Use order s, a, b, c, t. Relaxation gives

$$d[s] = 0, \; d[a] = 3, \; d[b] = 5, \; d[c] = 6, \; d[t] = 8.$$

Predecessors trace $t \leftarrow c \leftarrow b \leftarrow a \leftarrow s$; the shortest path is $s - a - b - c - t$, weight 8.

Exercise 29. Why negative weights are safe in a DAG.

Solution. When a vertex is processed topologically, every possible predecessor has already been processed, so its distance is final regardless of edge sign. A DAG has no cycle, hence no negative cycle. Dijkstra instead finalizes vertices by current smallest label, a rule invalidated by a later negative edge.

Exercise 30. Longest path in a DAG.

Solution. Negate every weight, compute shortest paths topologically, and negate the result. Here the longest path is

$$s \to a \to c \to t, \qquad 2 + 4 + 6 = 12.$$

Exercise 31. Count shortest paths in a unit-weight DAG.

Solution. Process topologically, maintaining distance d and count q. On a better relaxation set $q[v] = q[u]$; on an equal relaxation add $q[u]$. With $q[s] = 1$, the example gives

$$(d, q): \; s(0, 1), a(1, 1), b(1, 1), c(2, 2), d(2, 1).$$

The direct route $s - a - d$ is the unique shortest route to d.

Exercise 32. Compute transitive closure by repeated search.

Solution. Including zero-length reachability,

$$R = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

One BFS/DFS per source costs $\Theta(n(n + m))$.

Exercise 33. Warshall on a directed four-cycle.

Solution.

$$R^{(0)} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

At $k = 1$, add $4 \to 2$; at $k = 2$, add $1 \to 3, 4 \to 3$; at $k = 3$, add all consequences through 3, notably $1, 2, 4 \to 4$; at $k = 4$, every row becomes $(1, 1, 1, 1)$. Boolean matrix powers encode paths of fixed lengths; their Boolean union gives the same result.

Exercise 34. Warshall on the diamond DAG.

Solution.

$$R^{(0)} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

$R^{(1)} = R^{(0)}$. At $k = 2$, entry R_{14} becomes 1; no later iteration changes anything. Thus the final first row is all ones and the matrix stabilizes after $k = 2$.

Exercise 35. Compare transitive-closure complexities.

Solution. Repeated search costs $\Theta(n(n + m))$; Warshall costs $\Theta(n^3)$. For $m = \Theta(n^\alpha)$, equality occurs at $\alpha = 2$.

Exercise 36. Warshall as a semiring algorithm.

Solution. Warshall computes Boolean reachability rather than numeric distance. Path concatenation uses logical AND in place of addition, while choosing among alternative paths uses logical OR in place of minimum.

Exercise 37. BFS/DFS cost at different densities.

Solution. Both cost $\Theta(n + m)$. A tree has $m = n - 1$, hence $\Theta(n)$. For K_n, $m = n(n - 1)/2$, hence $\Theta(n^2)$.

Exercise 38. Graph search from an adjacency matrix.

Solution. When a vertex is dequeued, scan its whole matrix row to find white neighbors. Total cost is $\Theta(n^2)$ (a layer can require up to $O(n^2)$ scanning in the worst case), worse than $\Theta(n + m)$ on sparse graphs. Boolean matrix closure via repeated products can be organized in $O(n^3)$ with standard closure methods, matching Warshall's asymptotic bound but usually with more overhead.

Exercise 39. True or false: mixed graph claims.

Solution. (a) True. (b) False for weighted MST terminology: BFS ignores weights; with all weights equal, every spanning tree is minimum, so the special unit-weight case is trivially minimum. (c) True. (d) True: adding shortcuts cannot create a cycle absent before. (e) True: reachability uniquely determines the closure.

Exercise 40. DFS timestamp intervals.

Solution. For a forward arc, the intervals are properly nested: $d[u] < d[v] < f[v] < f[u]$. For a cross arc they are disjoint, typically $d[v] < f[v] < d[u] < f[u]$. In an undirected graph, the reverse edge would have discovered the later endpoint before the earlier subtree finished, contradicting disjointness.

Exercise 41. Cycles, SCCs, and condensation.

Solution. The graph is not a DAG. Its two directed cycles are $1 \to 2 \to 3 \to 1$ and $4 \to 5 \to 6 \to 4$, including their rotations. The SCCs are $\{1, 2, 3\}$ and $\{4, 5, 6\}$, with one condensation arc from the first to the second. The condensation has one source and one sink.

Exercise 42. Label connected components.

Solution. Run BFS/DFS from every unlabelled vertex, assigning a fresh component number during that search. The components are

$$\{1, 2, 3\}, \quad \{4, 5\}, \quad \{6, 7, 8\},$$

so there are three. The total cost is $\Theta(n + m)$.

Exercise 43. Compute graph diameter with BFS.

Solution. Run BFS from every vertex and take the largest finite distance found. This is exact in $\Theta(n(n + m))$. A single BFS reveals only the eccentricity of its source; an arbitrary source need not be an endpoint of a diametral pair.

Exercise 44. DAG shortest paths with unit weights.

Solution. The graph is a DAG; one order is s, a, c, d, b, t. Distances are

$$d[s] = 0, \quad d[a] = d[c] = 1, \quad d[d] = d[b] = 2, \quad d[t] = 3.$$

There are two shortest paths: $s - a - b - t$ and $s - c - d - t$.

Exercise 45. Critical path in a project DAG.

Solution.

$$ES[s] = 0, \qquad ES[v] = \max_{(u,v) \in A} \{ES[u] + w(u, v)\}.$$

Evaluate this recurrence in topological order. Here

$$ES[a] = 3, \; ES[b] = 2, \; ES[c] = 7, \; ES[t] = 12.$$

The critical path is $s - a - t$, of length 12.

Exercise 46. Detect an undirected cycle with DFS.

Solution. Store each vertex's parent. When scanning uv, a visited neighbor $v \neq \text{parent}(u)$ proves a cycle; the invariant is that the parent edges form a forest. Conversely, the first non-tree edge on any cycle triggers this test.

Complexity is $\Theta(n+m)$.

Exercise 47. Properties of a BFS tree.

Solution. Connectivity makes BFS discover every vertex, and each nonroot receives one parent, so the parent edges form a spanning tree. The layer inequality $|d[u]-d[v]| \le 1$ holds for every non-tree edge. BFS is not an MST algorithm for arbitrary weights: in a triangle rooted at s, give edges sa, sb weight 10 and ab weight 1; BFS may choose both expensive root edges.

Exercise 48. Shortest paths in the second weighted DAG.

Solution. Use order $1, 3, 2, 4, 5$. Relaxations give

$$d = (0, 3, 2, 6, 8).$$

The improvements are $d_2 : 4 \to 3$ via 3, $d_4 : 10 \to 6$ via 2, and $d_5 : 9 \to 8$ via 4. The shortest path is $1 \to 3 \to 2 \to 4 \to 5$, weight 8.

Exercise 49. Finish times and topological order.

Solution. For any DAG arc $u \to v$, DFS cannot classify it as back. If v is a descendant, $f[v] < f[u]$; if it is already finished, again $f[v] < f[u]$. Therefore every arc points forward in decreasing finish order, which is a valid topological order.

Exercise 50. All-pairs reachability output and cost.

Solution. Output an $n \times n$ Boolean reachability matrix, requiring $\Theta(n^2)$ bits. Repeated DFS costs $\Theta(n(n+m))$; Warshall costs $\Theta(n^3)$. If $m = O(n \log n)$, repeated DFS costs $O(n^2 \log n)$, asymptotically faster.

Exercise 51. Kahn, counting orders, and longest path.

Solution. Choosing the smallest source produces

$$1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,$$

with source sets successively $\{1\}, \{2, 3\}, \{3, 4\}, \{4, 5, 6\}, \{5, 6\}, \{6\}, \{7\}$. Counting the admissible interleavings gives 16 topological orders. The longest path has three arcs, for example $1 \to 2 \to 4 \to 7$.

Exercise 52. Equivalence of DAG and topological order.

Solution. A topological order forbids a cycle because positions would strictly increase around it. Conversely, every nonempty DAG has a source; remove it, order the remaining DAG inductively, and prepend it. Thus the two statements are equivalent.

Exercise 53. Why BFS and DFS are linear.

Solution. Initialization, discovery, queue/stack insertion, and removal cost $O(1)$ per vertex, totaling $O(n)$. Every adjacency-list entry is scanned once: once per arc in a digraph and twice per edge in an undirected graph, totaling $O(m)$. Hence both run in $O(n + m)$.

Exercise 54. Bipartite density and BFS coloring.

Solution. For parts of sizes $p, n - p$, $m \le p(n - p)$, maximized at $\lfloor n^2/4 \rfloor$. The given graph has coloring

$$L = \{1, 2, 3\}, \qquad R = \{4, 5, 6\}.$$

BFS rejects when an edge joins equal colors; on a triangle, after the source colors its two neighbors alike, their connecting edge creates the contradiction.

Exercise 55. Shortest and longest paths in the final DAG.

Solution. An order is $1, 2, 3, 4, 5$. Relaxation gives

$$d_1 = 0, \quad d_2 = 2, \quad d_3 = 4, \quad d_4 = 7, \quad d_5 = 5.$$

The unique shortest path to 5 is $1 \to 2 \to 3 \to 5$, weight 5. Maximizing instead gives $1 \to 3 \to 4 \to 5$, weight $5 + 3 + 1 = 9$.

CHAPTER 9

Minimum Spanning Trees

Exercise 1. Spanning-tree basics.

Solution. A spanning tree is a connected, acyclic subgraph containing every vertex. Deleting a leaf and inducting proves that every tree on n vertices has $n - 1$ edges. In the cycle C_5, deleting any one edge gives a spanning tree, so there are already five distinct examples.

Exercise 2. Unique paths and fundamental cycles.

Solution. A tree has exactly one u-v path; two would form a cycle. Adding a non-tree edge uv closes precisely that path, creating one cycle. For an MST, every such edge e satisfies $w(e) \geq \max\{w(f) : f \text{ lies on the tree path}\}$.

Exercise 3. Fundamental cuts.

Solution. Deleting a tree edge e creates two components; their cut is the fundamental cut of e. With distinct weights, e is lighter than every other crossing edge. Cycle and cut properties are exchange statements in opposite directions: insert a non-tree edge or delete a tree edge.

Exercise 4. True or false: MST structure.

Solution.

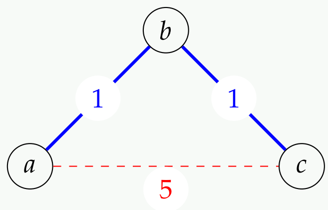
(a) **True** (assuming the edge is not a self-loop). By the cut property, considering a cut separating the endpoints of the globally minimum edge, this edge is strictly the lightest edge crossing the cut, so it must be in every MST.

(b) **True.** The cut property and cycle property uniquely determine the inclusion or exclusion of every edge when weights are distinct.

(c) **False.** Equal weights do not automatically create multiple MSTs if those edges do not form a cycle with other equally weighted edges.

(d) **True** (assuming it is not a self-loop and the graph is connected). You can always build a spanning tree by starting with that edge and extending it.

(e) **False.** A spanning tree, by definition, contains no cycles! (If the statement meant "removing the heaviest edge from a cycle in the *graph* preserves the ability to form an MST", that would be true).



Counterexample for (c): Two edges have the same weight (1), yet the MST (in blue) is unique.

Exercise 5. Apply the cut property to $S = \{a, c, d\}$.

Solution. Crossing edges are $ab(4), bc(5), bd(3), ce(6), de(7)$. The unique minimum is $bd(3)$, so every MST contains bd.

Exercise 6. Prove the cut property.

Solution. If an MST T omits the unique lightest crossing edge e, adding e creates a cycle containing another crossing edge f. Replacing f by e gives a lighter spanning tree, contradiction.

Exercise 7. Two cuts and an MST.

Solution. For $\{1, 2, 3\}$, crossing edges are $14(9), 24(7), 34(6), 35(2)$, so 35 is forced. For $\{1\}$, $13(5)$ is forced. Kruskal gives

$$T = \{46(1), 35(2), 23(3), 45(4), 13(5)\}, \qquad w(T) = 15.$$

Exercise 8. Prove the cycle property.

Solution. If an MST contained the unique heaviest cycle edge e, deleting it would split the tree. Some other cycle edge reconnects the parts and is lighter, yielding a cheaper tree, contradiction.

Exercise 9. Check a proposed MST and one cycle.

Solution. $\{ac, bd, ab\}$ connects all four vertices without a cycle and has weight $1 + 2 + 3 = 6$. Adding $cd(5)$ creates $c - a - b - d - c$; cd is its unique heaviest edge and cannot belong to an MST.

Exercise 10. Kruskal trace I.

Solution. Accept $ac(1), de(1), bd(2), ab(3)$; reject $bc(4), cd(5)$ as cyclic; accept $ce(6)$. Thus

$$T = \{ac, de, bd, ab, ce\}, \qquad w(T) = 13.$$

Exercise 11. Kruskal trace II.

Solution. In the given sorted order, accept

$$56(1), 23(2), 14(3), 35(4), 12(5).$$

All six vertices are then connected; the remaining edges would create cycles. The MST weight is 15.

Exercise 12. Kruskal trace III.

Solution. Sorted order:

$$rs;\ pq, tu, sv;\ qr, qu;\ st;\ uv;\ pv;\ rt.$$

Accept rs, pq, tu, sv, qr, qu, whose successive unions connect all seven vertices. The total weight is $1 + 2 + 2 + 2 + 3 + 3 = 13$.

Exercise 13. Kruskal trace IV.

Solution. Accept $DE(1), AC(2), CD(3), AB(4)$; reject BD, BC, CE because their endpoints are already connected; accept $DF(8)$. The MST weight is 18.

Exercise 14. Prim from a.

Solution. The extraction sequence is a, c, b, d, e, using

$$ac(1),\ ab(3),\ bd(2),\ de(1).$$

Keys are updated from incident crossing edges at each extraction. The MST weight is 7.

Exercise 15. Prim from d.

Solution. One extraction sequence is d, e, b, a, c, selecting

$$de(1), db(2), ba(3), ac(1).$$

It is the same edge set and weight 7. Distinct weights make this MST unique, so the start vertex cannot change it.

Exercise 16. Prim from two roots.

Solution. Both traces produce

$$\{13(1), 24(2), 34(3), 35(7), 46(9)\}, \qquad w = 22.$$

The extraction order changes with the root, not the optimum.

Exercise 17. Prim heap complexity.

Solution. Binary heaps perform n ExtractMin and at most m DecreaseKey operations, each $O(\log n)$, for $O((n+m)\log n) = O(m \log n)$. Fibonacci heaps make decrease-key amortized $O(1)$, giving $O(m + n \log n)$.

Exercise 18. Compare Prim and Kruskal.

Solution. Binary-heap Prim: $O(m \log n)$; Fibonacci Prim: $O(m + n \log n)$; Kruskal: $O(m \log m)$. Fibonacci Prim is best asymptotically on dense graphs; on sparse graphs all are $O(n \log n)$. Without union-find, repeated connectivity tests can cost $O(n+m)$ per edge.

Exercise 19. Union-find in Kruskal.

Solution. Find returns a component representative; Union merges two components. Add uv iff $\text{Find}(u) \neq \text{Find}(v)$. Rank plus path compression gives $O(m\alpha(n))$ total. A fresh DFS connectivity test would cost $O(n+m)$ per edge.

Exercise 20. Extendable sets and greedy correctness.

Solution. $\varnothing$ is contained in every MST. The exchange proof for the cut property shows that adding a safe edge preserves extendability. Prim and Kruskal begin empty and repeatedly add safe edges until they have $n-1$ edges, hence finish with an MST.

Exercise 21. Safe edges.

Solution. On a five-vertex weighted path plus chords, every lightest edge crossing a component cut is safe; a unique heaviest edge on a cycle is not. Kruskal uses the cut between two current components. Prim uses the cut between its current tree and the remaining vertices.

Exercise 22. MST uniqueness examples.

Solution. With distinct weights, the lightest edge in the symmetric difference of two alleged MSTs yields an exchange contradiction. A unit-weight C_4 has four MSTs. A path on four vertices with weights $1, 1, 2$ has repeated weights but only one spanning tree, hence a unique MST.

Exercise 23. Characterize a unique MST.

Solution. Inserting non-tree e and removing the heaviest path edge changes weight by $w(e) - w_{\max}$. Strict positivity for every e is equivalent to uniqueness; equality creates an alternative MST. Unique minimum across every cut is sufficient but not necessary, as a graph may have tied irrelevant edges while all exchange differences stay strict.

Exercise 24. Enumerate spanning trees of the four-vertex graph.

Solution. The eight trees and weights are

$$\begin{array}{c|l} \{ab,ac,bd\},\{ab,ac,cd\} & 5 \\ \{ab,bc,bd\},\{ab,bc,cd\},\{ac,bc,bd\},\{ac,bc,cd\} & 6 \\ \{ab,bd,cd\},\{ac,bd,cd\} & 7 \end{array}$$

There are two MSTs. The cut $\{a\}$ has tied minimum edges ab, ac.

Exercise 25. Maximum spanning tree.

Solution. Negating weights turns maximization into minimization. Equivalently, descending Kruskal uses the maximum-edge cut property. Here it accepts

$$de(7), bc(6), ab(5), bd(4),$$

for maximum total weight 22.

Exercise 26. Bottleneck spanning trees.

Solution. At the first Kruskal weight that connects the graph, no tree can avoid an edge at least that heavy; hence every MST is bottleneck-optimal. The converse fails on a triangle of weights $1, 2, 2$: both weight-2 bottlenecks are optimal, but totals 3 and 4 differ. The given MST is $\{34(2), 13(3), 45(4), 23(5)\}$, bottleneck 5.

Exercise 27. Negative edge weights.

Solution. Prim and Kruskal remain correct: their cut proofs compare weights but never assume nonnegativity. The MST is the spanning tree with the smallest, i.e. most negative, total weight.

Exercise 28. An ILP for MST.

Solution.

$$\min \sum_e w_e x_e, \quad \sum_e x_e = n-1, \quad \sum_{e \in E(S)} x_e \leq |S|-1 \ (\varnothing \neq S \subsetneq V), \quad x_e \in \{0,1\}.$$

$n-1$ edges alone may form a cycle plus a disconnected vertex. Equivalent connectivity constraints are $\sum_{e\in\delta(S)} x_e \geq 1$; one family forbids internal cycles, the other forces every cut to be crossed.

Exercise 29. MST polytope and integrality.

Solution. TU means every square minor is $0, \pm 1$, implying integrality with an integer RHS. The exponential spanning-tree description is instead the base polytope of the graphic matroid; its integrality follows from matroid polyhedral theory. Linear objectives can therefore be optimized greedily, without branch-and-bound.

Exercise 30. Second-best MST.

Solution. For each non-tree edge, insert it and remove the heaviest edge on its fundamental cycle; the best positive increase gives the second tree. Checking all edges and paths directly costs $O(n^2)$. Here the MST is $\{ab, bc, cd, de\}$, weight 10; inserting $ae(5)$ and removing $de(4)$ gives the second-best weight 11.

Exercise 31. Numerical MST sensitivity.

Solution.

$$T = \{bc(1), ab(2), cd(3), de(6)\}, \qquad w(T) = 12.$$

Across the fundamental cut of cd, the alternative is $bd(5)$, so cd may rise by 2 before a tie. For $ac(4)$, the heaviest edge on path $a - b - c$ weighs 2, so it must fall by 2 to tie.

Exercise 32. General sensitivity margin.

Solution. If e's increase is below $\delta = \min_f(w(f) - w(e))$, it remains strictly cheapest on its fundamental cut. At δ, an exchange ties; above it, the alternative is better. A non-tree edge f can decrease until it equals the maximum tree-edge weight on its fundamental cycle.

Exercise 33. Telecommunications network.

Solution. Kruskal selects

$$C_4C_5(5), C_5C_6(6), C_3C_4(7), C_2C_3(8), C_1C_3(10).$$

This connects all cities for 36 thousand euros.

Exercise 34. Electrical-grid model.

Solution. Vertices are substations, candidate links are weighted edges, and the cheapest nonredundant connected network is an MST. Positive-cost cycles are removable. Single-link fault tolerance requires a 2-edge-connected spanning subgraph, not a tree.

Exercise 35. Correctness of Kruskal.

Solution. When Kruskal joins two components, its edge is lightest across their cut and hence safe. It never creates a cycle and stops after $n - 1$ accepted edges, so the result is a spanning tree. Repeated safety proves it is extendable to an MST; at size $n - 1$, it is that MST.

Exercise 36. Correctness of Prim.

Solution. At every step use the cut $(V(T), V \setminus V(T))$. Prim chooses its lightest crossing edge, which is safe by the cut property. Induction preserves extendability until all vertices are included.

Exercise 37. Equal-weight spanning trees.

Solution. Equal total weight does not imply either tree is optimal; both may be suboptimal. If one is an MST, the other with equal weight is also an MST. Compute one MST value (Kruskal/Prim) and compare their common weight with it.

Exercise 38. Insert a new edge into an MST.

Solution. Add $e^* = uv$, forming one cycle. If it is heavier than every path edge, the cycle property discards it and T remains optimal. Otherwise remove a maximum-weight path edge f with $w(f) > w(e^*)$; the updated tree is lighter and is an MST of the augmented graph.

Exercise 39. Delete an MST edge.

Solution. Deleting e creates components T_1, T_2. Scan all remaining edges and choose the lightest crossing their cut; adding it gives the new MST by the cut property. A direct scan costs $O(m)$.

Exercise 40. True or false: MST algorithms.

Solution. (a) True for ordinary edges. (b) True. (c) False when multiple MSTs exist. (d) False: MST minimizes total tree weight, not root distances. (e) True, because every tree has exactly $n-1$ edges and gains the same $(n-1)c$.

Exercise 41. Kruskal sorting complexity.

Solution. Union-find costs $O(m\alpha(n))$, while comparison sorting costs $O(m \log m)$, which dominates. Integer weights in $[1, W]$ permit counting/bucket sort in $O(m+W)$. For K_n: binary Prim and comparison Kruskal are $O(n^2 \log n)$, Fibonacci Prim is $O(n^2)$.

Exercise 42. Prim implementations.

Solution. An array finds each minimum in $O(n)$, giving $O(n^2)$, optimal for dense adjacency-matrix input. A binary heap uses n extracts and at most m decreases for $O(m \log n)$. Fibonacci heaps make DecreaseKey $O(1)$ amortized, yielding $O(m + n \log n)$.

Exercise 43. Complete graph with $w(ij) = i + j$.

Solution. Weights are

$$12:3, 13:4, 14:5, 23:5, 15:6, 24:6, 25:7, 34:7, 35:8, 45:9.$$

Kruskal selects 12, 13, 14, 15, the star centered at 1, of total weight $3 + 4 + 5 + 6 = 18$.

Exercise 44. Graphic matroid.

> **Solution.** Matroid axioms are $\varnothing \in \mathcal{I}$, hereditary closure, and augmentation: a smaller independent set can take an element from a larger one. Forests satisfy these because deleting edges preserves acyclicity and two forests of different size admit a component-joining edge. Matroid greedy gives a maximum-weight basis; using negated weights gives Kruskal's minimum spanning tree.

Exercise 45. Prim on the seven-vertex instance.

> **Solution.** Starting at A, one trace selects
>
> $$AB(4), AC(8), CF(2), CE(1), FD(2), EG(9).$$
>
> The extraction/key updates may tie at weight 8, but the total is 26.

Exercise 46. Kruskal on the same instance.

> **Solution.** Sorted processing accepts $CE(1), CF(2), DF(2), AB(4)$, rejects $DE(6), CD(7)$, accepts $AC(8)$, rejects $BD(8)$, and accepts $EG(9)$. The resulting weight is again 26.

Exercise 47. MST versus shortest-path tree.

> **Solution.** In a triangle rooted at s, weights $sa = 2, sb = 2, ab = 1$ make the MST $\{sa, ab\}$, but $d_T(s, b) = 3 > 2$. Any weighted tree is itself both its MST and its shortest-path tree. Prim minimizes the next crossing edge; Dijkstra minimizes the total tentative distance from the root.

Exercise 48. Spanning trees of K_4.

> **Solution.** Cayley gives $4^{4-2} = 16$. They consist of four stars (one per center) and twelve labeled paths (each undirected Hamiltonian path counted once). All have weight 3, so all sixteen are MSTs and the MST is highly nonunique.

Exercise 49. Water-pipe network.

> **Solution.** Kruskal selects
>
> $$D_4D_5(1), D_3D_4(2), D_1D_2(3), D_5D_6(3), D_2D_3(4), D_6D_7(4),$$
>
> total 17. If D_3D_4 rises to 6, edge $D_2D_4(5)$ replaces it; the new total is 20.

Exercise 50. MST clustering.

> **Solution.** Removing $k-1$ tree edges creates exactly k components. Choosing the heaviest edges separates clusters across the largest MST gaps; for $k = 2$, the removed edge defines the two sides of its fundamental cut. This

captures non-spherical shapes, but is sensitive to noise and single-link chaining, unlike centroid-based k-means.

Exercise 51. Prim versus Kruskal output.

Solution. Their edge sets may differ when the MST is not unique; distinct weights guarantee equality. Both algorithms are correct, so their total weights always equal the optimum. A unit-weight cycle lets different tie orders delete different cycle edges and produce different MSTs.

Exercise 52. MST with fundamental cycles and cuts.

Solution. Kruskal selects

$$ab(1), sa(2), cd(2), dt(3), ac(3), \qquad w = 11.$$

For each omitted edge, its weight is at least the maximum on its tree path ($sb : 4 \geq 2$, $bc : 5 \geq 3$, $bd : 6 \geq 3$, $ct : 7 \geq 3$). Equivalently, every tree edge is minimum on its fundamental cut.

Exercise 53. All edge weights equal.

Solution. Every spanning tree has weight $n - 1$, so the number of MSTs equals the number of spanning trees. Kruskal accepts any edge joining different components and rejects cycle-closing edges. A connected graph with $m > n - 1$ contains a cycle, and deleting different cycle edges yields different spanning trees; hence it cannot have exactly one.

Exercise 54. LP description of the MST polytope.

Solution.

$$x(E) = n - 1, \quad x(E(S)) \leq |S| - 1 \; (\emptyset \neq S \subsetneq V), \quad 0 \leq x_e \leq 1.$$

This is the graphic-matroid base polytope, whose extreme points are tree incidence vectors. Its integrality is not a simple node–arc TU result. Thus MST needs neither branching nor cuts: greedy algorithms solve it directly.

Exercise 55. Global weight transformations.

Solution. Multiplication by $\lambda > 0$ preserves all comparisons of tree weights. Adding any constant c, even negative, adds exactly $(n - 1)c$ to every spanning tree. Both transformations therefore preserve the full set of MSTs.

Shortest Paths

Exercise 1. Dijkstra trace I.

Solution. Extract $1, 2, 3, 4, 5$. Updates are $(2, 5, \infty, \infty)$, then $d_3 = 3, d_4 = 9$, then $d_4 = 6, d_5 = 9$, then $d_5 = 7$. Final $d = (0, 2, 3, 6, 7)$.

Exercise 2. Dijkstra trace II.

Solution. Extraction order A, C, B, E, D, F, distances $(0, 3, 2, 8, 5, 8)$. Predecessors are $\pi_C = A, \pi_B = C, \pi_E = B, \pi_D = B, \pi_F = E$.

Exercise 3. Dijkstra trace III.

Solution. Final distances are $(0, 7, 9, 20, 20, 11)$. The shortest path to 5 is $1 \longrightarrow 3 \longrightarrow 6 \longrightarrow 5$, of cost 20.

Exercise 4. Dijkstra predecessor array.

Solution. Distances are $(0, 7, 3, 8, 9)$, with

$$\pi = (\text{NIL}, 3, 1, 2, 2).$$

Exercise 5. Shortest-path tree arcs.

Solution. The tree is AB, BC, CD, CE, with distances $0, 1, 3, 4, 6$. Each tree arc is tight: $d[v] = d[u] + c(u, v)$.

Exercise 6. Reconstruct a path and check reduced costs.

Solution. $5 \longleftarrow 3 \longleftarrow 4 \longleftarrow 2 \longleftarrow 1$, so the path is $1, 2, 4, 3, 5$. For any actual arc, check $\bar{c}_{ij} = c_{ij} + d[i] - d[j] \geq 0$; tree arcs must have equality. Arc costs are not supplied, so no numerical three-arc check is possible.

Exercise 7 (Dijkstra correctness). Prove label finality.

Solution. If extracted u had a shorter path, let y be its first unvisited vertex and x its predecessor. When x was processed, nonnegative weights would have given $d[y] \le \delta(s, u) < d[u]$, contradicting extraction of u.

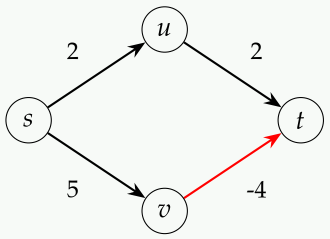
Exercise 8. Dijkstra complexity.

Solution. The binary-heap bound is $O((n + m) \log n)$, not universally $O(n^2)$. For dense graphs it can exceed n^2; array Dijkstra gives $O(n^2)$.

Exercise 9 (True or False). Shortest-path facts.

Solution.

(a) **False.** Dijkstra's algorithm permanently finalizes a vertex's distance when it is extracted from the priority queue. A negative arc can later offer a shorter path to a finalized vertex.



1. u extracted ($d = 2$)
2. t extracted ($d = 4$)
3. v extracted ($d = 5$), discovers t at $5 - 4 = 1$

Too late! t was already finalized at 4.

(b) **False.** Bellman–Ford is $O(nm)$ because it relaxes all m arcs across $n - 1$ passes. If the graph is dense ($m = \Theta(n^2)$), this becomes $O(n^3)$.

(c) **True.** Floyd–Warshall computes the shortest path from v to v. If $D[v, v] < 0$ at the end of the algorithm, vertex v belongs to a negative-weight cycle.

(d) **True.** By processing nodes in topological order, we ensure that when a node is processed, all paths to it have been fully evaluated. This requires exactly one pass over all vertices and arcs, hence $O(n + m)$, regardless of arc signs.

(e) **True.** Both algorithms correctly solve the single-source shortest path problem when there are no negative cycles. Therefore, they mathematically output the exact same final shortest-path tree and distance vector.

Exercise 10. Bellman–Ford trace I.

Solution. With arcs scanned in the listed order, pass 1 gives $(0, 6, 7, 2, 9)$; passes 2–4 do not change it. Predecessors: $(\text{NIL}, 1, 1, 2, 4)$.

Exercise 11. Bellman–Ford trace II.

Solution. Pass 1 already gives $(0, 1, -2, 1)$; later passes are unchanged. The shortest A-D path is $A - B - C - D$, cost 1.

Exercise 12. Bellman–Ford trace III.

Solution.

$$(d, \pi) : 1(0, -),\ 2(2, 1),\ 3(-1, 2),\ 4(1, 1),\ 5(0, 3),\ 6(1, 5).$$

The cycle $3 \to 5 \to 6 \to 3$ has weight 0, not negative.

Exercise 13. Detect a negative cycle by inspection.

Solution. Cycle $2 \to 3 \to 4 \to 2$ has weight $4 - 2 - 5 = -3$. Therefore an additional Bellman–Ford pass still decreases a reachable label.

Exercise 14. Bellman–Ford with a negative cycle.

Solution. The cycle $2 \to 3 \to 4 \to 2$ weighs $2 - 4 + 1 = -1$. Each pass around it lowers d_2, d_3, d_4; the fifth pass still changes a label, certifying the cycle. Distances are therefore not finite shortest-path values.

Exercise 15. Floyd–Warshall negative-cycle test.

Solution. A negative cycle exists iff some final $D_{ii} < 0$. All displayed diagonal entries are zero, so this matrix does not certify one.

Exercise 16. DAG shortest paths I.

Solution. Processing $1, \ldots, 6$ gives final

$$d = (0, 3, 6, 7, 8, 10).$$

The key improvements are $d_4 = 7$ via 2, $d_5 = 8$ via 4, and $d_6 = 10$ via 5.

Exercise 17. DAG shortest paths with a negative arc.

Solution.

$$d_A = 0,\ d_B = 2,\ d_C = 3,\ d_D = 1,\ d_E = 3.$$

Arc $CD = -2$ violates Dijkstra's hypothesis; topological DP remains valid because all predecessors are finalized before a vertex.

Exercise 18. DAG shortest-path tree.

Solution.

$$d = (0, 1, 2, 4, 5), \qquad \pi = (\text{NIL}, 3, 1, 2, 4).$$

Exercise 19. Longest path in a four-node DAG.

Solution. Negate weights and run DAG shortest paths, or maximize directly. The longest path is $1 \to 2 \to 3 \to 4$, length $3 + 2 + 4 = 9$.

Exercise 20. Longest path in a six-node DAG.

Solution. Maximum labels give $L_F = 9$. Two longest paths are $A-B-E-F$ and $A-C-D-F$, both of length 9.

Exercise 21 (CPM — Basic project network). Compute earliest starts.

Solution.

$$ES_A = ES_B = 0,\ ES_C = 3,\ ES_D = 5,\ ES_E = 5,\ ES_F = 9,\ ES_G = 12.$$

The critical path is $B-D-F-G$, duration 14.

Exercise 22 (CPM — Forward and backward pass). Compute floats.

Solution. Earliest starts: $A0, B0, C4, D4, E3, F10, G14$; duration 17. Latest starts: $A0, B5, C4, D8, E9, F10, G14$. Floats are $0,5,0,4,6,0,0$; critical path $A-C-F-G$.

Exercise 23 (CPM as longest path). Explain the equivalence.

Solution. Events are nodes and activities are duration-weighted arcs. The forward pass uses $E[v] = \max_{uv}(E[u] + w_{uv})$. Precedence cannot contain a cycle in a feasible project, so the network is a DAG.

Exercise 24 (CPM — Float and resource management). Analyze delays.

Solution. Critical path $A-B-D-F$, duration 14. Branch $A-C-E-F$ lasts 10, so C, E have float 4. Delaying C by 2 does not change completion.

Exercise 25 (Floyd-Warshall — Full trace). Compute APSP.

Solution. Successive intermediates add paths through $1,2,3,4$. The final matrix is

$$\begin{pmatrix} 0 & 3 & 5 & 6 \\ 5 & 0 & 2 & 3 \\ 3 & 6 & 0 & 1 \\ 2 & 5 & 7 & 0 \end{pmatrix}.$$

Exercise 26 (Floyd-Warshall — Intermediate node argument). Prove the recurrence.

Solution. An optimal path whose allowed internal vertices are $1, \dots, k$ either avoids k, or splits at k into i-k and k-j paths using only earlier intermediates. Hence the stated minimum; in-place updates are safe because zero-length reuse of k adds nothing without a negative cycle.

Exercise 27 (Floyd-Warshall — Path reconstruction). Store intermediates.

Solution. When $D_{ij} > D_{ik} + D_{kj}$, set D_{ij} to that sum and $\Pi_{ij} = k$. Recursively print $i \to j$: if Π_{ij} is empty, print the direct endpoints; otherwise reconstruct $i \to k$ and $k \to j$, suppressing the duplicate k.

Exercise 28 (Floyd-Warshall with a negative arc). Compute APSP.

Solution. The final matrix is

$$\begin{pmatrix} 0 & 2 & 5 & 1 \\ 7 & 0 & 3 & -1 \\ 4 & 6 & 0 & 5 \\ \infty & \infty & \infty & 0 \end{pmatrix}.$$

All diagonal entries remain zero, so there is no negative cycle.

Exercise 29 (ILP formulation for SSSP). Write the flow model.

Solution.

$$\min \sum c_{ij} x_{ij}, \quad \sum_j x_{ij} - \sum_j x_{ji} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise}, \end{cases} \qquad x \in \{0,1\}.$$

The matrix is directed node–arc incidence, hence TU; the LP relaxation has an integral optimal vertex.

Exercise 30 (TUM and integrality). Prove network integrality.

Solution. Each incidence column has one +1, one −1, and zeros elsewhere, which satisfies the Ghouila-Houri condition. With integral unit-flow RHS, every LP vertex is integral, so binary restrictions are unnecessary.

Exercise 31 (Reduced costs and optimality). Check potentials.

Solution. Optimality requires $\bar{c}_{ij} = c_{ij} + d_i - d_j \geq 0$, with equality on tree arcs. The listed arcs give reduced costs

$$0, \; 1, \; 0, \; 0, \; 0,$$

so $d = (0, 3, 4, 5)$ is feasible and optimal.

Exercise 32 (Optimality conditions: counterexample). Correct the labels.

Solution. Reduced costs are $0, -2, 1$ for $12, 13, 23$. The negative value on 13 proves the labels are not optimal. The true distances are $(0, 4, 2)$.

Exercise 33 (Complexity comparison). Complete the table.

Solution. Dijkstra: nonnegative, no cycle detection, $O((n + m) \log n)$. Bellman–Ford: negative allowed, detects cycles, $O(nm)$. DAG-DP: any weights, $O(n + m)$. Floyd–Warshall: any weights, detects cycles, $O(n^3)$.

Bellman–Ford changes from $O(n^2)$ sparse to $O(n^3)$ dense. For $m = O(n \log n)$, DAG-DP is fastest when applicable, otherwise Dijkstra for nonnegative weights.

Exercise 34 (Dense vs sparse). Choose a Dijkstra implementation.

Solution. Heap Dijkstra is preferable when $m \ll n^2$; the array version is $O(n^2)$ and preferable for dense graphs. At $m = n^2/\log n$, the heap bound is $O(n^2)$, equal asymptotically.

Exercise 35. Bellman–Ford as dynamic programming.

Solution.

$$d^{(k)}[v] = \min\left(d^{(k-1)}[v], \min_{uv \in A}\{d^{(k-1)}[u] + c_{uv}\}\right).$$

Without a negative cycle, a shortest simple path has at most $n - 1$ arcs, so $d^{(n-1)}$ is final.

Exercise 36. Compute shortest paths using at most k arcs.

Solution.

k	1	2	3	4	5
1	0	1	4	∞	∞
2	0	1	3	5	7
3	0	1	3	4	6
4	0	1	3	4	6

Minimum arc counts are $0, 1, 2, 3, 3$, respectively.

Exercise 37. Compare shortest- and longest-path complexity.

Solution. Shortest paths are polynomial via Dijkstra/Bellman–Ford. Longest simple path contains Hamiltonian path as a special case and is NP-hard. In a DAG, topological order removes cyclic dependencies and enables linear-time DP.

Exercise 38 (Negative arc weights and Dijkstra failure). Give a counterexample.

Solution. Use $s \to a = 2$, $s \to b = 5$, $b \to a = -10$. Dijkstra finalizes a at 2, but the true distance is -5 through b. A later negative arc invalidates label finality.

Exercise 39 (Re-weighting and Johnson's idea). Explain Johnson.

Solution. $\hat{c}_{ij} = c_{ij} + h_i - h_j \geq 0$ exactly when $h_j \leq h_i + c_{ij}$. Bellman–Ford from a new zero-cost super-source computes such potentials. Then n Dijkstra runs cost $O(nm + n^2 \log n)$ with Fibonacci heaps, better than $O(n^3)$ on sparse graphs.

Exercise 40 (Dijkstra on an undirected graph). Compute the tree.

Solution. Distances are $(0, 3, 2, 8, 10)$; tree edges are $13, 32, 24, 45$. Replacing each edge by two equal opposite arcs exactly preserves all undirected walks and therefore all shortest distances.

Exercise 41 (SSSP uniqueness). Discuss ties.

Solution. Arcs $sa = 1, at = 4, sb = 2, bt = 3$ have distinct weights but give two length-5 paths. If every vertex has one shortest path, its last arc uniquely determines every predecessor, hence the tree. A reproducible rule is to choose the smallest-index predecessor on ties.

Exercise 42 (Sensitivity analysis). Decrease one arc cost.

Solution. Labels remain feasible iff the new reduced cost $\bar{c}_{uv} - \delta \geq 0$. If it is negative, $d[v]$ initially falls by $\delta - \bar{c}_{uv}$, and descendants may also improve. Example: $s \to u = 2, s \to v = 5, u \to v = 4$; lowering uv from 4 to 1 changes $d[v]$ from 5 to 3.

Exercise 43 (Floyd-Warshall and transitive closure). Use Boolean algebra.

Solution. $T_{ij}^{(k)} = T_{ij}^{(k-1)} \vee (T_{ik}^{(k-1)} \wedge T_{kj}^{(k-1)})$. The example yields

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

Warshall costs $O(n^3)$; repeated search costs $O(n(n+m))$.

Exercise 44 (Shortest path duality). Write and interpret the dual.

Solution. With source supply 1 and sink demand 1, the dual is

$$\max \pi_s - \pi_t \quad \text{s.t.} \quad \pi_i - \pi_j \le c_{ij}.$$

Complementary slackness says $x_{ij} > 0$ only on tight arcs $c_{ij} - \pi_i + \pi_j = 0$: optimal flow follows zero reduced-cost arcs.

Exercise 45 (Dijkstra with a Fibonacci heap). Compare heaps.

Solution. Fibonacci heaps give $O(m + n \log n)$, asymptotically better when m dominates n. DecreaseKey is $O(1)$ amortized and ExtractMin $O(\log n)$. Constants often limit the practical dense-graph advantage.

Exercise 46 (Bellman-Ford: early termination). Prove the optimization.

Solution. If a full pass changes nothing, every relaxation inequality already holds, so repeating the same operations cannot change a label. Graphs whose shortest paths use $O(1)$ arcs stop after $O(1)$ passes, saving a constant fraction. Worst-case complexity remains $O(nm)$.

Exercise 47 (Dijkstra trace with tie-breaking). Analyze equal labels.

Solution. Vertices 2, 3 tie at 4. Final distances are $(0, 4, 4, 7, 9)$ regardless of order. One tree uses 24, another 34; both then use 45, together with 12, 13.

Exercise 48 (Bellman-Ford on a complete graph). Run three passes.

Solution. There are twelve arcs, with weights $|i-j|$. The first pass produces $(0, 1, 2, 3)$, and later passes do not change it. All weights are nonnegative, so the extra pass detects no negative cycle.

Exercise 49 (DAG DP: counting paths). Count paths.

Solution. With $q_1 = 1$ and $q_v = \sum_{uv} q_u$,

$$q = (1, 1, 2, 3, 5).$$

Thus there are five paths to 5. For parity, maintain $(q_v^{\text{even}}, q_v^{\text{odd}})$ and swap parity when extending by one arc.

Exercise 50 (CPM: crashing activities). Minimize crash cost.

Solution. The critical path is $A - C$, duration 11. Crashing A by one unit costs 3 and reduces duration to 10. To reach 9, crash A by two units for minimum total cost 6.

Exercise 51 (Negative cycle: weight computation). Inspect cycles.

Solution. The only directed cycle is $3 \to 4 \to 5 \to 3$, with weight $-2 - 4 + 2 = -4$. It is negative, so a reachable Bellman–Ford run still improves labels on the n-th pass.

Exercise 52 (Floyd-Warshall: asymmetric graph). Compute APSP.

Solution. Starting from the stated asymmetric matrix, the final result is

$$\begin{pmatrix} 0 & 1 & 3 & 4 \\ 4 & 0 & 7 & 5 \\ 3 & 2 & 0 & 1 \\ 2 & 3 & 5 & 0 \end{pmatrix}.$$

For example $d_{12} = 1 \neq 4 = d_{21}$.

Exercise 53 (ILP and LP relaxation on a small graph). Solve the flow LP.

Solution. Use balance $+1$ at s, -1 at t, and 0 at nodes $1, 2$, with $0 \leq x \leq 1$. TU makes the LP optimum integral:

$$x_{s1} = x_{12} = x_{2t} = 1,$$

all others zero. Path $s - 1 - 2 - t$ has cost $2 + 1 + 1 = 4$.

Exercise 54 (Shortest path on a grid graph). Run DAG DP.

Solution. Order nodes by nondecreasing $i + j$. DP gives distance 12 to $(3, 3)$, for example along $(1, 1), (2, 1), (3, 1), (3, 2), (3, 3)$. There are $\binom{4}{2} = 6$ right/down paths.

Exercise 55 (Algorithm comparison). Compare Bellman–Ford and Dijkstra.

Solution. Dijkstra finalizes $1(0), 3(4), 2(5), 5(7), 4(10)$. Bellman–Ford reaches $(0, 5, 4, 10, 7)$ on the first listed-order pass and then stabilizes. Dijkstra scans each of seven arcs once; standard Bellman–Ford scans $4 \cdot 7 = 28$ arcs.

Network Flows

Exercise 1. Check the proposed four-node flow.

Solution. All values lie within capacity. At a, $4 = 1 + 3$; at b, $2 + 1 = 3$. The source sends $4 + 2 = 6$, equal to the sink inflow. Thus the flow is feasible and has value 6.

Exercise 2. Find a maximum-total-flow circulation.

Solution. Conservation is

$$f_{12} + f_{13} = f_{41}, \quad f_{23} + f_{24} = f_{12}, \quad f_{34} = f_{23} + f_{13}, \quad f_{41} = f_{34} + f_{24}.$$

One optimum is $f_{12} = f_{23} = f_{34} = f_{41} = 3$ and $f_{13} = f_{24} = 0$, with total arc flow 12. Positive circulation decomposes into directed cycles; without a directed cycle only zero flow is possible.

Exercise 3. Prove the divergence identities.

Solution. Every arc appears once with $+$ sign at its tail and once with $-$ sign at its head, so total divergence is zero. If internal divergences vanish, $\operatorname{div}(s) = -\operatorname{div}(t)$; either quantity therefore defines the same flow value.

Exercise 4. Check the five-node flow.

Solution. All capacities hold. Conservation gives $3 = 2 + 1$ at a, $4 = 4$ at b, and $2 + 4 = 6$ at c. The assignment is feasible and has value $3 + 4 = 7$; no correction is needed.

Exercise 5. Prove flow decomposition.

Solution. Follow positive-flow arcs from s. Conservation prevents termination before t, unless a vertex repeats and forms a cycle. Subtract the minimum flow on the found path or cycle; at least one positive arc vanishes. Iteration terminates and decomposes f. Path coefficients sum to $|f|$, while cycles contribute zero net source flow.

Exercise 6. Enumerate all cuts in a four-node network.

Solution.

S	$\text{cap}(S,\bar{S})$
$\{s\}$	5
$\{s,a\}$	8
$\{s,b\}$	6
$\{s,a,b\}$	7

The unique minimum cut is $\{s\}$. Sending 3 on $s-a-t$ and 2 on $s-b-t$ gives a flow of value 5.

Exercise 7. Formulate max flow as an LP and use TU.

Solution. Maximize $f_{sa}+f_{sb}$, impose conservation at a,b,c, and $0 \le f_e \le k_e$. The equality matrix is directed node–arc incidence; appending bound rows preserves TU. Integer capacities therefore admit an integer optimal basic flow.

Exercise 8. Prove flow–cut weak duality.

Solution. Summing conservation over S gives

$$|f| = f(S,\bar{S}) - f(\bar{S},S) \le f(S,\bar{S}) \le k(S,\bar{S}).$$

Strong duality asserts equality for some flow and cut. A maximum flow may be strictly below a nonminimum cut; for two parallel unit paths, the maximum is 2, while a cut augmented by an unused high-capacity arc can have larger capacity.

Exercise 9. Find all minimum cuts in the five-node network.

Solution. One maximum flow is

$$f_{s1}=5,\ f_{s2}=4,\ f_{12}=2,\ f_{13}=0,\ f_{23}=4, f_{1t}=3,\ f_{2t}=2,\ f_{3t}=4.$$

Its value is 9. The residual vertices reachable from s are only $\{s\}$; exhaustive cut checking shows this is the unique minimum cut.

Exercise 10. True or false: max-flow statements.

Solution.

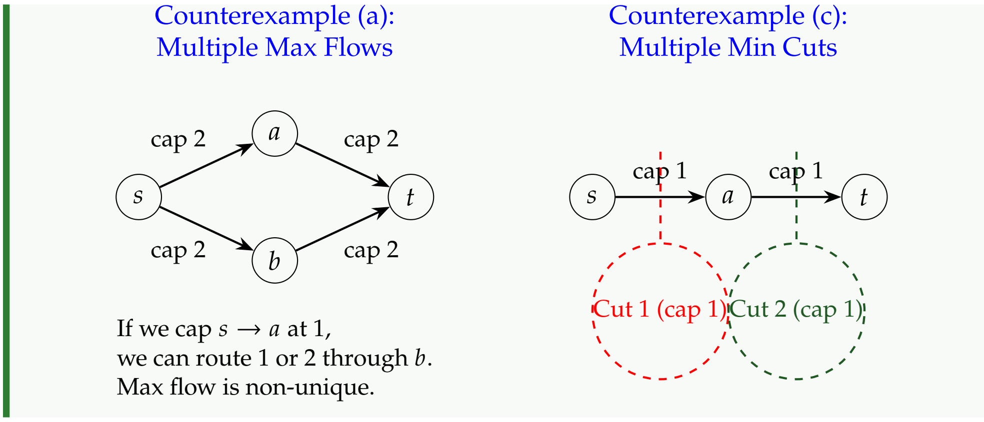
(a) **False.** There can be multiple different assignments of flow that achieve the maximum value.

(b) **True.** This is the integrality theorem, guaranteed by algorithms like Ford–Fulkerson that only add/subtract integers at each step.

(c) **False.** There can be multiple cuts with the same minimum capacity.

(d) **True.** This is the augmenting-path criterion (the core of the Ford–Fulkerson algorithm's correctness).

(e) **True.** This is the fundamental Max-Flow Min-Cut Theorem.



Exercise 11. Construct a residual network and augment.

Solution. The supplied assignment is not feasible: at a, inflow is 2 and outflow 4; at b, inflow 2 and outflow 0. Thus it has no valid flow residual interpretation. A corrected feasible flow $(f_{sa}, f_{sb}, f_{ab}, f_{at}, f_{bt}) = (4, 1, 1, 3, 2)$ has value 5; the residual path $s - b - t$ has bottleneck 2, yielding maximum value 7.

Exercise 12. Analyze an intermediate Ford–Fulkerson state.

Solution. The stated flow is not feasible: a receives 4 but sends 5, and b receives 7 but sends 6. Consequently the requested residual-network conclusion cannot follow from this assignment. After restoring conservation, residual arcs are obtained as usual with forward capacity $k - f$ and backward capacity f.

Exercise 13. Characterize maximum flow by residual paths.

Solution. An augmenting path increases the value, so a maximum flow has none. Conversely, if none exists, let S be the residual vertices reachable from s. Every S-to-$\bar{S}$ arc is saturated and every reverse arc carries zero; hence $|f| = \text{cap}(S, \bar{S})$, proving optimality.

Exercise 14. Ford–Fulkerson trace on four nodes.

Solution. Lexicographic choices may be

$$s - a - b - t \ (2), \qquad s - a - t \ (2), \qquad s - b - t \ (2), \qquad s - b - a - t \ (1),$$

where the last path uses the residual reverse of $a - b$. The final flow can be $f_{sa} = 4$, $f_{sb} = 3$, $f_{ab} = 1$, $f_{at} = 3$, $f_{bt} = 4$, value 7. Cut $\{s\}$ has capacity 7.

Exercise 15. Ford–Fulkerson trace on five nodes.

Solution. Use $s - a - t$ for 3, $s - a - c - t$ for 4, $s - b - t$ for 2, and $s - b - c - t$ for 3. The final flow is $7, 5, 4, 3, 3, 7, 2$ in the exercise's arc order, with value 12. Cuts $\{s\}, \{s, a\}, \{s, b\}, \{s, a, b\}$ all have capacity 12.

Exercise 16. Trace an instance intended to use a backward arc.

Solution. The proposed partial flow is infeasible: node c receives 3 but sends only 2. Hence no legitimate augmentation can start from it. For the actual network, a maximum flow is

$$f_{sa} = 4,\ f_{sc} = 3,\ f_{ab} = 4,\ f_{cb} = 0,\ f_{ba} = 0, f_{bt} = 4,\ f_{ct} = 3,$$

of value 7. The exercise's requested backward-arc trace is therefore based on inconsistent data.

Exercise 17. Find the six-node maximum flow and cuts.

Solution. One optimum sends

$$sa = 8,\ sb = 5,\ ac = 5,\ ad = 3,\ bc = 2,\ bd = 3,\ ct = 7,\ dt = 6.$$

Its value is 13. The unique minimum cut is $S = \{s, a, b, c, d\}$, whose outgoing capacity is $7 + 6 = 13$.

Exercise 18. A network with multiple minimum cuts.

Solution. Augment 3 on each of $s - a - t$ and $s - b - t$. The value is 6. All four cuts $S = \{s\}, \{s, a\}, \{s, b\}, \{s, a, b\}$ have capacity 6. The two independent equal-capacity branches cause nonuniqueness.

Exercise 19. Explain the zig-zag bad example.

Solution. Each alternating path changes the value by only 1: one uses $a - b$, the next its residual reverse. Repeating 100 pairs requires 200 augmentations to reach value 200. Edmonds–Karp instead uses $s - a - t$ and $s - b - t$, each with bottleneck 100, in two rounds.

Exercise 20. Derive Edmonds–Karp complexity.

Solution. Residual BFS costs $O(|A|)$. Shortest residual distances never decrease; after an arc is critical and later becomes critical again, its tail distance rises by at least two. Thus each arc is critical $O(|V|)$ times, giving $O(|V||A|)$ augmentations and $O(|V||A|^2)$ time.

Exercise 21. True or false: termination.

Solution. (a) True after scaling rational capacities to integers. (b) True. (c) True: Edmonds–Karp has a combinatorial iteration bound. (d) True, because each integer augmentation adds at least one. (e) True: its bound depends only polynomially on graph size.

Exercise 22. Explain irrational nontermination.

Solution. Adversarial choices can create an infinite sequence of irrational bottlenecks whose values converge below optimum. Edmonds–Karp avoids this through its finite shortest-path bound. Implementations should use a terminating rule such as Edmonds–Karp or Dinic, especially with floating-point capacities.

Exercise 23. Min-cost flow of four units.

Solution. Minimize

$$2f_{sa} + 3f_{sb} + f_{at} + 2f_{bt} + f_{ab}$$

subject to source supply 4, sink demand 4, conservation, and the listed capacities. Send all 4 units on $s - a - t$; cost $4(3) = 12$. Increasing $a - b$ does not help because $s - a - b - t$ costs 5 per unit.

Exercise 24. Route five units at minimum cost.

Solution. The cheapest path $s - b - c - t$ costs 5 and carries 3 units. The remaining 2 use $s - a - t$, also cost 5 per unit. Total cost is 25.

Exercise 25. Prove TU for min-cost flow.

Solution. The conservation matrix is directed incidence and is TU by Ghouila-Houri. Appending I and $-I$ for capacities preserves TU. Therefore integer supplies, demands, and capacities yield integral basic solutions.

Exercise 26. Write the general min-cost-flow primal and dual.

Solution.

$$\min\{c^\top x : Ex = b,\ 0 \le x \le u\}.$$

A nondegenerate network-simplex basis corresponds to a spanning tree after one redundant row is removed. With node potentials π and upper-bound multipliers $y \ge 0$, the dual may be written

$$\max b^\top \pi - u^\top y, \qquad E^\top \pi - y \le c.$$

Exercise 27. Reduce shortest path to min-cost flow.

Solution. For target t, set $b_s = 1, b_t = -1$, other balances zero, capacities one, and costs w. In the example the optimum sends one unit on $s-a-b-t$, of cost $2 + 1 + 1 = 4$, matching Dijkstra.

Exercise 28. Explain successive shortest paths.

Solution. Each iteration sends as much flow as possible along a cheapest residual path. Potentials reweight residual costs to nonnegative reduced costs, so Dijkstra can find each next path while preserving original path-cost comparisons.

Exercise 29. Reduce bipartite matching to max flow.

Solution. Add unit arcs $s \to U$, orient graph edges $U \to W$, and add unit arcs $W \to t$. Integer flow selects vertex-disjoint graph edges and vice versa. For the example, one perfect matching is $\{u_1w_1, u_2w_2, u_3w_3\}$, so maximum flow is 3.

Exercise 30. Derive Kőnig's theorem from max-flow min-cut.

Solution. A cut is converted to the cover $(U \setminus S) \cup (W \cap S)$; its capacity equals the cover size. Thus minimum cut equals minimum vertex cover, while maximum flow equals maximum matching. In the preceding graph, U itself is a cover of size 3, equal to the perfect matching size.

Exercise 31. Model assignment as min-cost flow.

Solution. Use unit arcs source–worker and task–sink, and unit worker–task arcs of cost c_{ij}. Send n units. TU gives an integer optimum. For the matrix shown, an optimum is $(1,2), (2,3), (3,1)$, of cost $2 + 1 + 7 = 10$.

Exercise 32. Solve the transportation instance.

Solution. An optimal plan is

$$\begin{pmatrix} 8 & 0 & 2 & 0 \\ 0 & 12 & 0 & 3 \\ 0 & 0 & 16 & 4 \end{pmatrix},$$

which meets all supplies and demands. Its minimum total cost is $16 + 2 + 12 + 6 + 32 + 16 = 84$.

Exercise 33. Solve the four-node max-flow LP.

Solution. Use integer variables with conservation $f_{sa} = f_{ab} + f_{at}$, $f_{sb} + f_{ab} = f_{bt}$, and capacity bounds; the relaxation merely changes integer variables to real ones. A maximum flow is

$$f_{sa} = 5, \; f_{sb} = 2, \; f_{ab} = 1, \; f_{at} = 4, \; f_{bt} = 3,$$

of value 7, and is integral by TU.

Exercise 34. Prove directed incidence is TU.

Solution. For a square submatrix, a column with at most one nonzero permits induction by expansion. If every column has two nonzeros, they are $+1, -1$, so the selected rows sum to zero and the determinant is zero. Hence every minor is $0, \pm 1$, which yields integral flow vertices.

Exercise 35. Complete the residual proof of max-flow min-cut.

Solution. If an S^*-to-T^* arc were unsaturated, its forward residual arc would make its head reachable. If a T^*-to-S^* arc carried positive flow, its reverse residual arc would do the same. Thus

$$|f^*| = f(S^*, T^*) - f(T^*, S^*) = k(S^*, T^*).$$

Exercise 36. Unit capacities and arc-disjoint paths.

Solution. An integral maximum flow is 0/1 and decomposes into arc-disjoint paths. Each augmentation adds one and saturates an arc, so at most $|A|$ occur. Thus max flow equals the maximum number of arc-disjoint paths. In the diamond $s \to a, b \to t$, there are exactly two such paths although additional walks can be formed if cross arcs are added.

Exercise 37. Verify the arc version of Menger's theorem.

Solution. Give every arc capacity one; minimum separating arc set is a minimum cut, and maximum integral flow decomposes into arc-disjoint paths. In the example, $s - a - t$ and $s - b - t$ are disjoint, and at least two arcs must be removed to separate s, t.

Exercise 38. Model node capacity by node splitting.

Solution. Replace v by $v^{in} \to v^{out}$ of capacity B, redirecting incoming arcs to v^{in} and outgoing arcs from v^{out}. Every flow correspondence is immediate. Here branch a carries at most 3, branch b at most 3, so maximum flow is 6.

Exercise 39. Ford–Fulkerson with one path.

Solution. There is one augmentation on $s - a - t$, bottleneck 7. The maximum value is 7, limited by $a - t$. Cut $S = \{s, a\}$ has capacity 7.

Exercise 40. Multiple sources and sinks.

Solution. After adding S, T, route

$$s_1a = 5,\ s_1b = 1,\ s_2a = 4, \quad at_1 = 5,\ at_2 = 4,\ bt_2 = 1.$$

This gives value 10, equal to total supply and demand; all requirements can be met.

Exercise 41. Interpret the max-flow dual as min cut.

Solution. Dual node potentials separate s from t, while nonnegative arc variables pay capacity when a potential drops across an arc. An optimal binary potential is a cut indicator, reducing the dual objective to $\sum_{e \in \delta^+(S)} k_e$. LP strong duality therefore gives max-flow min-cut.

Exercise 42. Sensitivity of a maximum flow.

Solution. The maximum value is 7, with both source cut and sink cut of capacity 7. Decreasing $s - a$ immediately lowers the source cut, so its allowable decrease is 0. Increasing $a - b$ alone never helps, because source and sink cuts remain 7. No single unit capacity increase raises the maximum: at least one source-side and one sink-side bottleneck must be enlarged.

Exercise 43. Reduce lower bounds to ordinary capacities.

Solution. Set $f' = f - \ell$, capacity $u' = k - \ell$, and adjust each node balance by lower-bound inflow minus outflow. A feasible original flow is

$$f_{sa} = 3,\ f_{ab} = 1,\ f_{bt} = 2,\ f_{sb} = 1,\ f_{at} = 2,$$

of value 4; all lower and upper bounds and conservation constraints hold.

Exercise 44. Feasibility with node demands.

Solution. The demands sum to $-4 + 1 + 2 + 1 = 0$. A feasible flow is

$$f_{12} = 4, \qquad f_{23} = 3, \qquad f_{34} = 1, \qquad f_{41} = f_{13} = 0.$$

Net inflow minus outflow is $1, 2, 1$ at nodes $2, 3, 4$, while node 1 supplies 4.

Exercise 45. Model project crashing as min-cost flow.

Solution. Crash variables are bounded by $d_i - d_i^{min}$, carry unit crashing cost, and must reduce every critical source–sink path enough to meet the deadline; the dual network form is a min-cost flow. In the two-task case:

$$\min 5y_1 + 3y_2, \quad y_1 + y_2 \geq 2, \quad 0 \leq y_1, y_2 \leq 2.$$

The optimum is $y_2 = 2$, cost 6.

Exercise 46. Prove the integer-capacity integrality theorem.

Solution. Starting from zero, residual capacities remain integers. Every bottleneck is therefore at least one, so each augmentation increases value by at least one. At most φ^* augmentations occur, and all intermediate and final flows are integer.

Exercise 47. Decompose the stated maximum flow.

Solution. The stated assignment is not feasible: node b receives 2 but sends 3. Hence it cannot be decomposed as a flow. If f_{bc} is corrected to 2, one decomposition is

$$2(s - a - t) + 1(s - a - c - t) + 2(s - b - c - t),$$

of value 5. Flow decompositions are generally nonunique when positive-flow branches can be recombined.

Exercise 48. Residual cycles and flow value.

Solution. Augmenting around a cycle has zero source divergence, so value is unchanged. A maximum flow can still have residual cycles, for example two saturated parallel branches with reversible residual arcs. The last claim in the exercise is false: only a *negative-cost* residual cycle proves that a min-cost flow can be improved; a positive cycle does not.

Exercise 49. Trace Edmonds–Karp and its layers.

Solution. Use $s - a - c - t$ with bottleneck 3, then $s - b - d - t$ with bottleneck 3, then $s - b - c - t$ with bottleneck 1. The flow value is 7. Each augmenting path has three arcs; afterward t is unreachable, so the distance sequence $3, 3, 3, \infty$ is nondecreasing.

Exercise 50. Model traffic throughput.

Solution. Arc flows are vehicles per hour, constrained by road capacities and intersection conservation. The minimum cut identifies roads whose combined capacity limits throughput. The example has maximum flow 16; a minimum cut is all vertices except t, with bottleneck roads $a - t(4)$, $b - t(9)$, $c - t(3)$.

Exercise 51. Planar dual cuts.

Solution. Under the standard planar assumptions (with terminals on the appropriate faces), primal s-t cuts correspond to separating dual paths, and capacities become dual edge lengths. A shortest such dual path gives a minimum cut. In the unit diamond, the minimum dual separation crosses two branch edges, so both max flow and min cut equal 2.

Exercise 52. Hospital–resident assignment as max flow.

Solution. Use unit source–resident and resident–hospital arcs, and capacities 2 on hospital–sink arcs. A full assignment is

$$r_2 \to h_1, \qquad r_1 \to h_1, \qquad r_3 \to h_2,$$

so max flow is 3. The source arcs, total capacity 3, form a minimum cut; residents, not hospital capacity, limit throughput.

Exercise 53. Supply-chain min-cost flow.

Solution. Supply and demand both total 50. An optimum ships

$$P_1D_1 = 20, \quad P_1D_2 = 10, \quad P_2D_2 = 5, \quad P_2D_3 = 15,$$

with all other flows zero. Total cost is $40 + 40 + 15 + 15 = 110$.

Exercise 54. Inverse max flow.

Solution. Introduce capacity increases $y_e \geq 0$ and require every cut to have $\sum_{e \in \delta^+(S)} (k_e + y_e) \geq v$, minimizing $\sum y_e$. Current value is 4. To reach 5, increase $s - a$ and $b - t$ by one; the extra unit uses $s - a - b - t$. Minimum total increase is 2.

Exercise 55. Parametric maximum flow.

Solution. The relevant cuts have capacities $\lambda + 2$ and 7, so

$$\varphi^*(\lambda) = \min\{\lambda + 2, 7\}.$$

Each unit increase helps for $0 \leq \lambda < 5$, and has no effect for $\lambda > 5$. At threshold $\lambda^* = 5$, the sink cut becomes minimum and $s - a$ ceases to belong to every minimum cut.

Bipartite Matching

Exercise 1. Define matching, saturated vertices, and exposed vertices.

Solution. A matching $M \subseteq E$ contains no two edges with a common endpoint. A vertex incident to an edge of M is saturated; otherwise it is exposed.

Exercise 2. Can an eight-vertex graph have a perfect matching?

Solution. Yes. A perfect matching has $|V|/2 = 4$ edges. Even cardinality is necessary because vertices are paired, but not sufficient: an isolated vertex prevents a perfect matching.

Exercise 3. Distinguish maximum and perfect matchings.

Solution. A maximum matching has largest possible cardinality; a perfect matching saturates every vertex. On the path P_5, a maximum matching has size 2 but leaves one vertex exposed, so it is not perfect.

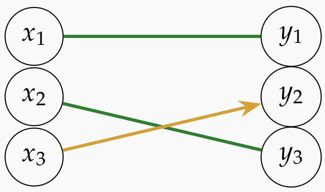
Exercise 4. Define alternating and augmenting paths.

Solution. An alternating path alternates edges outside and inside M. It is augmenting when both endpoints are exposed, so it begins and ends with unmatched edges and contains one more unmatched than matched edge.

Exercise 5. Augment the three-by-three matching.

Solution. The exposed vertices are x_3, y_2. Edge x_3y_2 itself is an augmenting path, hence

$$M' = M \oplus \{x_3y_2\} = \{x_1y_1, x_2y_3, x_3y_2\}.$$



Exercise 6. Build a perfect matching from the empty matching.

Solution. Use the one-edge augmenting paths $x_1y_1, x_2y_2, x_3y_4, x_4y_3$. After four augmentations,

$$M = \{x_1y_1, x_2y_2, x_3y_4, x_4y_3\},$$

which is perfect and therefore maximum.

Exercise 7. Run BFS labelling from exposed X-vertices.

Solution. Only x_3 is exposed in X. BFS labels its unmatched neighbors y_3, y_4; y_4 is exposed, so the one-edge path x_3y_4 augments the matching. Thus

$$M' = \{x_1y_2, x_2y_3, x_3y_4\}.$$

Exercise 8. Augment a matching with four left and three right vertices.

Solution. The direct path x_4y_3 augments to

$$M' = \{x_1y_1, x_3y_2, x_4y_3\}.$$

It has size $3 = |Y|$, the largest possible, so it is maximum without invoking Berge's theorem.

Exercise 9. State Berge's theorem.

Solution. A matching is maximum if and only if no augmenting path exists. An augmenting path enlarges a nonmaximum matching; conversely, comparing a smaller matching with a larger one exposes an augmenting component in their symmetric difference.

Exercise 10. Apply Berge to a perfect matching.

Solution. There are no exposed vertices, hence no augmenting path. Berge's theorem proves M maximum. Its size is 3.

Exercise 11. Prove maximum implies no augmenting path.

Solution. If P were augmenting, $M \oplus E(P)$ would remove k matched edges and add $k + 1$ unmatched edges. Its size would be $|M| + 1$, contradicting maximality.

Exercise 12. Prove no augmenting path implies maximum.

Solution. For a larger matching M^*, every component of $M \oplus M^*$ is an alternating path or even cycle. Since $|M^*| > |M|$, some path component contains one more M^*-edge than M-edge; its endpoints are exposed by M, so it is an M-augmenting path, contradiction.

Exercise 13. Run labelling to exhaustion.

Solution. The only exposed left vertex is x_4. It reaches y_3, then via the matched edge reaches x_3, but no new right vertex follows. No augmenting path exists. Indeed x_3, x_4 share only y_3, so the size-3 matching is maximum.

Exercise 14. Define vertex cover and state Kőnig's theorem.

Solution. A vertex cover meets every edge. In a bipartite graph, $\nu(G) = \tau(G)$: maximum matching size equals minimum cover size. For a triangle, $\nu = 1$ but $\tau = 2$, so the equality fails in general graphs.

Exercise 15. Find a matching and cover in the first 3×3 graph.

Solution.

$$M = \{x_1y_1, x_2y_2, x_3y_3\}$$

is perfect. With no exposed left vertices, the König construction gives the cover $C = X$, of size $3 = |M|$.

Exercise 16. Find a matching and cover in the 4×4 cycle graph.

Solution. One perfect matching is $\{x_1y_1, x_2y_2, x_3y_3, x_4y_4\}$. The set X is a vertex cover of size 4; König's theorem proves it minimum and equal in size to the matching.

Exercise 17. Explain vertex-cover complexity.

Solution. Find a maximum bipartite matching in polynomial time, then apply the alternating-reachability König construction. General minimum vertex cover is NP-hard because the bipartite equality and TU structure no longer apply.

Exercise 18. Write matching and vertex-cover primal–dual LPs.

Solution. A maximum matching is $\{x_1y_1, x_2y_3, x_3y_2\}$, so $\nu = 3$; $C = X$ is a cover of size 3. The primal and dual are

$$\max \sum_e x_e \quad \text{s.t.} \sum_{e \ni v} x_e \le 1,\ x \ge 0,$$

$$\min \sum_v y_v \quad \text{s.t. } y_u + y_v \ge 1\ (uv \in E),\ y \ge 0.$$

Exercise 19. State Hall's theorem.

Solution. A bipartite graph $G = (L \cup R, E)$ has a matching saturating L iff

$$|N(S)| \ge |S| \qquad \text{for every } S \subseteq L,$$

where $N(S)$ is the set of right vertices adjacent to at least one vertex of S.

Exercise 20. Check Hall on the symmetric three-by-three graph.

Solution. Every singleton has two neighbors, every pair has all three, and $N(L) = R$. Hall holds. For example $\{l_1r_1, l_2r_2, l_3r_3\}$ is perfect.

Exercise 21. Find a Hall violation with four left vertices.

Solution. For $S = \{l_1, l_2\}$, $N(S) = \{r_1\}$, so $1 < 2$. Also $|R| < |L|$. No matching can saturate L.

Exercise 22. Check Hall with an extra right vertex.

Solution. Singleton neighborhoods have size 2, every pair has at least 3 neighbors, and all three left vertices have four neighbors. Hall holds; $\{l_1r_1, l_2r_2, l_3r_3\}$ saturates L.

Exercise 23. Identify the common-neighbor Hall obstruction.

Solution. Take $S = L$. Then $N(S) = \{r_1\}$, so $|N(S)| = 1 < 3 = |S|$. At most one left vertex can be matched.

Exercise 24. Check Hall in the four-by-four graph.

Solution. The cyclic neighborhood pattern satisfies Hall; explicitly,

$$\{l_1r_2, l_2r_1, l_3r_4, l_4r_3\}$$

is a perfect matching and therefore certifies the condition.

Exercise 25. Does Hall imply uniqueness?

Solution. No. In $K_{2,2}$, Hall holds but both $\{l_1r_1, l_2r_2\}$ and $\{l_1r_2, l_2r_1\}$ are perfect matchings.

Exercise 26. Describe the max-flow reduction.

Solution. Add unit arcs $s \to X$, orient each graph edge $X \to Y$ with unit capacity, and add unit arcs $Y \to t$. Integrality makes every unit of flow choose one edge without sharing an endpoint, exactly a matching.

Exercise 27. Build and solve the 3×3 flow network.

Solution. Besides source and sink arcs, include the six listed $x_i \to y_j$ arcs, all capacity one. Sending one unit through

$$s - x_1 - y_1 - t, \quad s - x_2 - y_2 - t, \quad s - x_3 - y_3 - t$$

gives maximum flow and matching size 3.

Exercise 28. Solve the four-by-four flow reduction.

Solution. The unit network admits paths corresponding to

$$\{x_1y_1, x_2y_2, x_3y_3, x_4y_4\}.$$

Thus maximum flow and maximum matching size are 4.

Exercise 29. Translate a minimum cut into a cover.

Solution. The maximum matching has size 2, for example $\{x_1y_2, x_2y_1\}$. The source cut $S = \{s\}$ has capacity 2 and maps to cover $X = \{x_1, x_2\}$, also of size 2.

Exercise 30. Write the bipartite matching LP.

Solution.

$$\max \sum_{e \in E} x_e, \qquad \sum_{e \ni v} x_e \le 1 \quad (v \in V), \qquad 0 \le x_e \le 1.$$

The upper bounds are redundant given nonnegativity and endpoint constraints, but make the relaxation explicit.

Exercise 31. Explain integrality of the matching LP.

Solution. The constraint matrix is the vertex–edge incidence matrix of a bipartite graph. Signing the two vertex classes oppositely proves it TU. With integer RHS 1, every LP vertex is integral.

Exercise 32. Solve a small matching primal–dual pair.

Solution. The primal constraints are

$$x_{11} + x_{12} \le 1, \quad x_{21} \le 1, \quad x_{11} + x_{21} \le 1, \quad x_{12} \le 1, \quad x \ge 0.$$

The dual assigns nonnegative weights to the four vertices with $y_u + y_v \ge 1$ per edge. Primal $x_{12} = x_{21} = 1$ and dual $y_{x_1} = y_{x_2} = 1$, others zero, both have value 2 and satisfy complementary slackness.

Exercise 33. True or false: matching LP.

Solution.

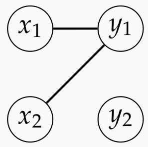
(a) **False.** Hall's condition can fail even if $|X| = |Y|$.

(b) **False.** If a graph has multiple perfect matchings, the LP can choose any of them, or any convex combination of them.

(c) **True.** The incidence matrix of any bipartite graph is totally unimodular.

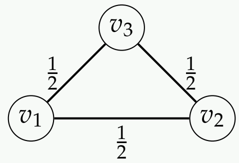
(d) **True.** In non-bipartite graphs (which contain odd cycles), the LP can cheat.

Counterexample (a)
No Perfect Matching



$|X| = |Y| = 2$, but both x only like y_1. Hall fails.

Counterexample (d)
Fractional cheating



Max matching = 1.
LP relaxation = 1.5.

Exercise 34. True or false: Hall and Kőnig.

Solution.

(a) **False.** Same as above, having equal sides or being bipartite is not enough.
(b) **False.** A complete bipartite graph $K_{2,2}$ satisfies Hall's condition for every subset, but it clearly has 2 different perfect matchings.
(c) **False.** Kőnig's theorem only applies to bipartite graphs. In a triangle (K_3), the max matching size is 1, but the min vertex cover size is 2.
(d) **True.** We can use the augmenting path algorithm, Hopcroft–Karp, or cast it as a max-flow problem.

Exercise 35. Formulate the assignment ILP.

Solution.

$$\min \sum_{i,j} c_{ij} x_{ij}, \quad \sum_j x_{ij} = 1 \ \forall i, \quad \sum_i x_{ij} = 1 \ \forall j, \quad x_{ij} \in \{0,1\}.$$

Exercise 36. Relax the assignment ILP.

Solution. Replace $x_{ij} \in \{0,1\}$ by $x_{ij} \geq 0$. Bipartite incidence is TU; equivalently, the Birkhoff–von Neumann theorem says the extreme points of the doubly stochastic polytope are permutation matrices.

Exercise 37. Solve the three-worker assignment.

Solution. The LP uses row and column sums one and $x \geq 0$. An optimum is

$$w_1 \to j_2, \qquad w_2 \to j_3, \qquad w_3 \to j_1,$$

with total cost $1 + 3 + 1 = 5$.

Exercise 38. Outline the Hungarian algorithm.

Solution. Subtract row minima, then column minima; cover all zeros with a minimum number of lines. If fewer than n lines are needed, adjust uncovered and doubly covered entries to create new zeros. Augment among zeros until a perfect assignment is found. Complexity is $O(n^3)$.

Exercise 39. Model domino tiling as matching.

Solution. Chessboard-color the cells. Create one vertex per cell and an edge between orthogonally adjacent black and white cells. A domino is an edge; a complete tiling is exactly a perfect matching.

Exercise 40. Tile the ordinary chessboard.

Solution. Pair adjacent cells horizontally in each row. This explicitly gives 32 disjoint black–white edges covering all cells, hence a perfect matching. Hall's condition follows as a consequence.

Exercise 41. Remove two diagonally opposite corners.

Solution. The exercise incorrectly calls the corners opposite colors. On an 8×8 board, diagonally opposite corners have the *same* color. Removing them leaves 30 cells of that color and 32 of the other, so $|B| \neq |W|$ and no domino tiling exists.

Exercise 42. Remove two squares of the same color.

Solution. The remaining color counts are 30 and 32. Every domino covers one of each color, so a complete tiling is impossible, regardless of the positions.

Exercise 43. Tile a corner-deleted 4×4 grid.

Solution. Delete all four corners and connect orthogonally adjacent cells of opposite colors. A perfect matching is given by domino pairs

$$(1,2)(1,3),\ (2,1)(3,1),\ (2,2)(2,3),\ (2,4)(3,4),\ (3,2)(4,2),\ (3,3)(4,3).$$

Thus the twelve cells are tileable.

Exercise 44. Assign five jobs to machines.

Solution. The bipartite edges are exactly the listed feasible assignments. A perfect matching is

$$\{j_1m_1, j_2m_2, j_3m_3, j_4m_4, j_5m_5\}.$$

All five jobs are scheduled; none is exposed.

Exercise 45. Assign students to lab slots.

Solution. One perfect matching is

$$\{s_1t_1, s_2t_2, s_3t_4, s_4t_3\}.$$

Direct neighborhood checking (or existence of this matching) verifies Hall's condition.

Exercise 46. Formulate task–employee feasibility.

> **Solution.** Put tasks on one side, employees on the other, and connect compatible pairs. All tasks can be assigned iff there is a matching saturating the task side, equivalently iff every task subset S has $|N(S)| \geq |S|$.

Exercise 47. Explain blossoms.

> **Solution.** Bipartite BFS relies on a consistent even/odd layering. An odd alternating cycle can give a vertex conflicting parities and hide an augmenting path. Such an odd cycle with a common base is a blossom.

Exercise 48. Outline Edmonds' blossom algorithm.

> **Solution.** When an odd alternating cycle is found, contract it to one supervertex, continue the augmenting-path search, and later expand it while preserving alternation. Contraction preserves existence of an augmenting path.

Exercise 49. Give a five-vertex blossom example.

> **Solution.** Use triangle $1, 2, 3$ plus edges $1-4, 2-5$, with current matching $\{1-2\}$. The graph has maximum size 2, and path $4-1-2-5$ augments. A naïve bipartite parity labelling encounters the odd triangle and can mislabel or abandon the search; shrinking the triangle preserves and reveals the augmenting structure.

Exercise 50. Compare matching algorithm complexities.

> **Solution.** Repeated BFS augmentations take $O(VE)$. Hopcroft–Karp augments a maximal set of shortest paths per phase in $O(E\sqrt{V})$. Classical Edmonds blossom handles general graphs in $O(V^3)$ by shrinking odd cycles.

Exercise 51. Solve the mixed four-by-four instance.

> **Solution.**
> $$M = \{x_1y_2, x_2y_3, x_3y_1, x_4y_4\}$$
> is perfect. Thus X is a minimum cover of size 4, Hall holds, and the matching LP optimum is the incidence vector of M, value 4.

Exercise 52. Prove a regular bipartite graph has a perfect matching.

> **Solution.** For $S \subseteq L$, exactly $k|S|$ edges leave S, while at most $k|N(S)|$ enter its neighborhood. Hence $|N(S)| \geq |S|$. Hall's theorem gives a perfect matching.

Exercise 53. Solve a three-by-four instance via matching and flow.

Solution.

$$M = \{x_1y_1, x_2y_2, x_3y_3\}$$

has maximum size $3 = |X|$. The corresponding unit flow has value 3. The source cut maps to cover X, of size 3, which is minimum.

Exercise 54. Relate matching, covers, and independent sets.

Solution. Gallai's identity gives $\alpha(G) + \tau(G) = |V|$. In bipartite graphs, König gives $\tau(G) = \nu(G)$. Therefore

$$\alpha(G) = |V| - \nu(G).$$

Exercise 55. Prove the converse direction of Hall's theorem.

Solution. The contrapositive is immediate: if $|N(S)| \geq |S|$ for every $S \subseteq L$, Hall's theorem supplies a matching saturating L, of size $|L|$. Therefore, if every matching has size below $|L|$, some S must satisfy $|N(S)| < |S|$.