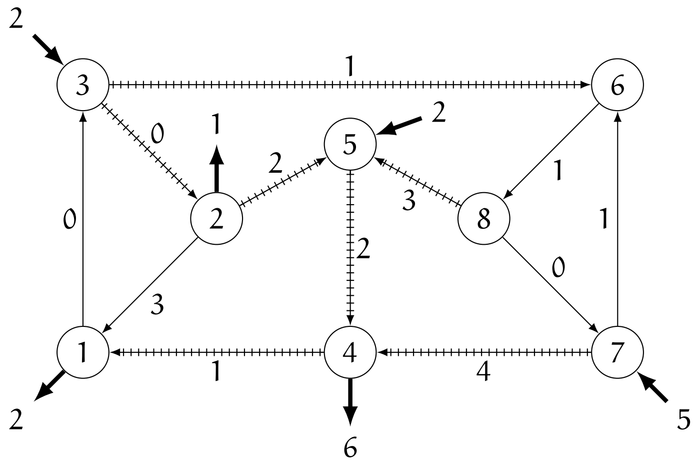
1 Use the network simplex method to find a minimum cost flow of the following uncapacitated flow problem, where each edge is labeled with its cost. Start from the spanning tree indicated by hatched edges. If at some point you should encounter a situation where no positive amount of flow can be pushed around a cycle, continue to apply the network simplex method by pushing an amount of zero and explain what happens.



2 Prove or disprove that for a Hitchcock transportation problem with strictly positive costs, the optimal cost is weakly increasing in supply and demand as long as the problem remains feasible, i.e., that the optimal cost cannot decrease as the supply and demand of some vertices increases.

3 Use the network simplex method to solve the transportation problem described by the following tableau:

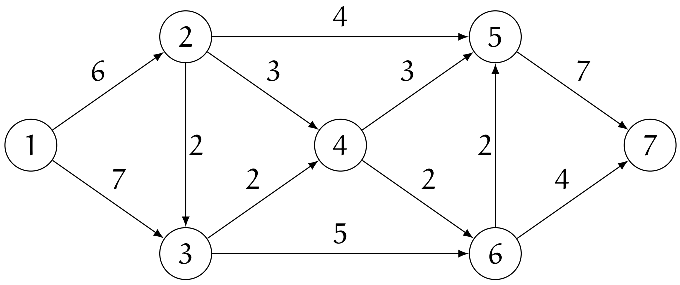
7	4	9	20
8	12	5	42
3	11	7	18
39	34	7	

4 Show that 2SAT, the special case of SAT in which every clause consists of exactly two literals, can be solved in polynomial time. Observe that a clause of size two can be viewed as an implication. For a given formula with two literals per clause construct a directed graph in which vertices correspond to literals and edges correspond to clauses. Then show that satisfiability of the formula is equivalent to the non-existence of a certain type of path in the graph. Argue that existence or non-existence of these paths can be decided in polynomial time.

5 Suppose that the pieces of 5 identical 800-piece jigsaw puzzles are scrambled and grouped into 800 piles of 5 pieces each. Prove or disprove that it is possible to pick exactly one piece from each of the piles and assemble these pieces into a complete puzzle.

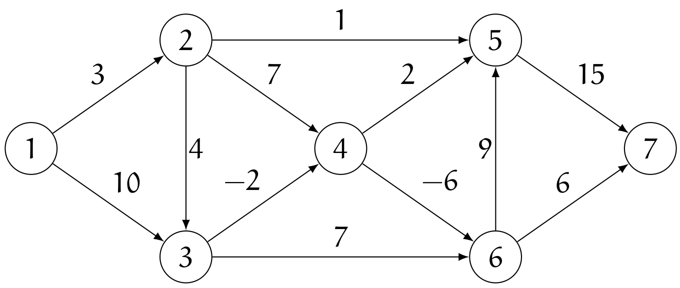
6 Show that a bipartite graph $G = (L \uplus R, E)$ with $|L| = |R|$ has a perfect matching if and only if $|N(X)| \geqslant |X|$ for every $X \subseteq L$, where $N(X) = \{j \in R : i \in X, (i,j) \in E\}$. The implication in one direction is obvious. For the other direction, again consider the flow network with a source s connected to vertices in L and a sink t connected to vertices in R, and show that if G does not have a perfect matching, then this network has a cut S with $s \in S$, $t \in V \setminus S$, and $C(S) < |L|$. Let $L_S = L \cap S$, $R_S = R \cap S$, and $L_T = L \setminus S$, and show that the capacity of S is exactly $|L_T| + |R_S|$. Use this to prove that $|N(L_S)| < L_S$.

7 Consider the following network, where the number on each edge indicates its capacity:



Find the maximum flow from vertex 1 to vertex 7, and prove that this flow is indeed optimal.

8 Consider the following network, where the number on each edge indicates its length:



(a) Use the Bellman-Ford algorithm to find a shortest path from vertex 1 to vertex 7.

(b) Find a shortest path between any pair of vertices, using an approach that minimizes the asymptotic running time.

9 Use Dakin's method to solve the following IP:

$$\begin{aligned} \text{maximize} \quad & 8x_1 + 5x_2 \\ \text{subject to} \quad & x_1 + x_2 \leqslant 6 \\ & 9x_1 + 5x_2 \leqslant 45 \\ & x_1, x_2 \geqslant 0 \\ & x_1, x_2 \in \mathbb{N} \end{aligned}$$

Instead of carrying out iterations of the simplex method, you may draw the feasible set in $\mathbb{R}^2$ and use this drawing to explain carefully how Dakin's method proceeds.

10 Consider a network (V, E) with $V = \{1, \ldots, 6\}$, $E = V \times V$, and edge costs c_{ij} for $(i, j) \in E$ given by

$$C = \begin{pmatrix} 0 & 6 & 7 & 6 & 7 & 6 \\ 6 & 0 & 5 & 6 & 6 & 6 \\ 7 & 5 & 0 & 5 & 7 & 6 \\ 6 & 6 & 5 & 0 & 8 & 9 \\ 7 & 6 & 7 & 8 & 0 & 5 \\ 6 & 6 & 6 & 9 & 5 & 0 \end{pmatrix}.$$

(a) Find a TSP tour by starting from vertex 1 and applying the nearest neighbor heuristic, breaking ties toward vertices with smaller index. Improve this tour as much as possible using local search with the 2-OPT neighborhood.

(b) For each $i \in V$, find a minimum cost spanning tree of the network obtained by removing vertex i from (V, E). Use the information thus obtained to derive a lower bound on the cost of any TSP tour, and conclude that the TSP tour found above is optimal.

(c) Describe a branch-and-bound method for the TSP that combines the above lower bound with an appropriate branching rule.

11 Describe a polynomial-time 2-approximation algorithm for instances of the TSP satisfying the triangle inequality, i.e., instances where $c_{ik} \leqslant c_{ij} + c_{jk}$ for all $i, j, k \in V$. Start from the observation that for an arbitrary tree, there exists a walk that visits every edge in the tree exactly twice and returns to the initial vertex. Then show that following this walk in a minimum-cost spanning tree, and skipping vertices that have already been visited, yields a TSP tour with the desired properties.