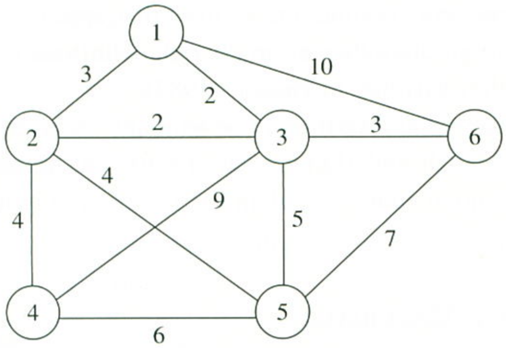
Exercises

3.1 Use Kruskal's algorithm to find a minimum spanning tree of the following graph.



3.2 Use Prim's algorithm to find the minimum spanning tree of the graph in Problem 3.1.

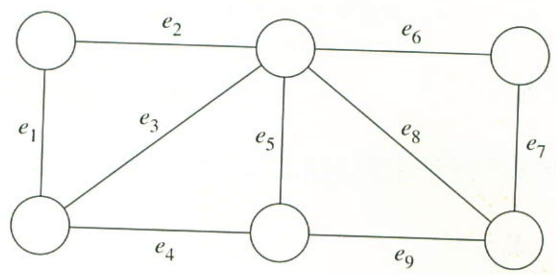
3.3 Prove the correctness of Dijkstra's algorithm.

3.4 (a) Show why Dijkstra's algorithm will not work properly when the considered graph contains negative cost edges.

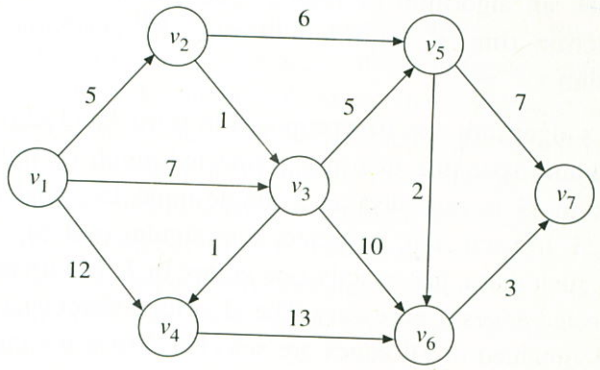
(b) Modify Dijkstra's algorithm so that it can compute the shortest path from source node to each node in an arbitrary graph with negative cost edges, but no negative cycles.

3.5 Obtain a set of optimal Huffman codes for the eight messages $(M_1, M_2, \ldots, M_8)$ with access frequencies $(q_1, q_2, \ldots, q_8) = (5, 10, 2, 6, 3, 7, 12, 14)$. Draw the decode tree for this set of code.

3.6 Obtain a minimum cycle basis for the following graph.



3.7 Use Dijkstra's algorithm to find the shortest paths from V_1 to all other nodes.



3.8 Give a greedy method, which is heuristic, to solve the 0/1 knapsack problem and also give an example to show that it does not always yield an optimal solution.

3.9 Implement both Prim's algorithm and Kruskal's algorithm. Perform experiments to compare the performances of these two algorithms.

3.10 Read Section 12–4 of Papadimitriou and Steiglitz (1982), for the relationship between matroids and the greedy algorithm.

3.11 The knapsack problem is defined as follows:
Given positive integers $P_1, P_2, \ldots, P_n, W_1, W_2, \ldots, W_n$ and M.
Find $X_1, X_2, \ldots, X_n$, $0 \le X_i \le 1$ such that

$$\sum_{i=1}^{n} P_i X_i$$

is maximized subject to

$$\sum_{i=1}^{n} W_i X_i \le M.$$

Give a greedy method to find an optimal solution of the knapsack problem and prove its correctness.

3.12 Consider the problem of scheduling n jobs on one machine. Describe an algorithm to find a schedule such that its average completion time is minimum. Prove the correctness of your algorithm.

3.13 Sollin's algorithm was first proposed in Boruvka (1926) for finding a minimum spanning tree in a connected graph G. Initially, every vertex in G is regarded as a single-node tree and no edge is selected. In each step, we select a minimum cost edge e for each tree T such that e has exactly one vertex in T. Eliminate the copies of selected edges if necessary. The algorithm terminates if only one tree is obtained or all edges are selected. Prove the correctness of the algorithm and find the maximum number of steps of the algorithm.