1. (20) Consider the union-find problem on a set of n elements $\{1, 2, \ldots, n\}$. Initially each element is a subset by itself. There are two types of operations on these subsets of S.

   (a) The *find* operation $F(x)$ returns the subset to which the element x belongs.

   (b) The *union* operation $U(u, v)$ makes the two subsets u and v into one subset.

   Note that each subset has a unique name, and the name cannot be changed. Suppose that each subset is represented by a tree. Each vertex has a pointer to its parent, except the root whose pointer points to itself.

   (a) (5) Assume that $n = 7$, and each subset is named by the smallest element in that subset. Show the subsets by drawing the forest after each of the following sequence of union operations:
   $U(1, 2), U(3, 4), U(1, 3), U(5, 6), U(1, 5)$.

   (b) (15) Design efficient algorithms for the union and find operations so that each find operation can be done in $O(\log n)$ time and the union can done in $O(1)$ time.

2. (20) In an undirected graph, the *degree* $d(u)$ of a vertex u is the number of neighbors u has, or equivalently, the number of edges incident upon it. In a directed graph, we distinguish between the *indegree* $d_{in}(u)$, which is the number of edges into u, and the *outdegree* $d_{out}(u)$, the number of edges leaving u.

   (a) Show that in an undirected graph, $\sum_{u \in V} d(u) = 2|E|$.

   (b) Use part (a) to show that in an undirected graph, there must be an even number of vertices whose degree is odd.

   (c) Does a similar statement hold for the number of vertices with odd indegree in a directed graph?

3. (20) Let $G = (V, E, w)$ be a connected weighted undirected graph with positive weight function on the edge set E.

   (a) Describe the Kruskal's algorithm for finding a minimum spanning tree of G.

   (b) Give an efficient implementation for the Kruskal's algorithm, and analyze the running time of your algorithm.

4. (20) Given a weighted graph $G = (V, E, w)$. Suppose that we want to find a spanning tree with maximum weight, instead of a minimum one. Design an algorithm for finding a maximum spanning tree, and show that your algorithm is correct.

5. (20) Let G be an undiretced graph with positive weights on each edges of the graph. Supposed that a constant $a > 0$ is added to the weight of each edge.

   (a) Show that the minimum spanning tree does not change.

   (b) Give an example to show that the shortest path from some vertex to another vertex may not be the same after adding the constant a.

   (c) Give a convincing reason why the two problems behalf differently with respect to adding a constant.

6. (20) Give a dynamic programming solution to the 0-1 knapsack problem that runs in $O(nW)$ time, where n is the number of items and W is the maximum weight of items that the thief can put in his knapsack.

7. (20) A contiguous sub-sequence of a sequence $x_1, x_2, \ldots, x_n$ is a sub-sequence $x_i, x_{i+1}, x_{i+2}, \ldots, x_j$ for some $1 \le i \le j \le n$. Given a sequence of integers $x_1, x_2, \ldots, x_n$, find a contiguous sub-sequence whose sum is maximized. Design a linear-time algorithm for the problem by using dynamic programming approach. That is, define the subproblems and write down the recurrence equation for solving the problem. Then use the input

   $$5, 15, -30, 10, -5, 40, 10$$

   to show how to compute the solution.