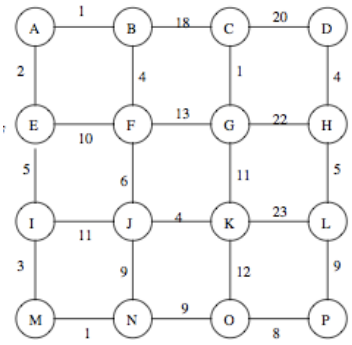
CSE 373: Analysis of Algorithms

Assignment 4 (Nov. 2nd 2016)

**Due date and time**: Nov 14th 17:00p.m.
Submit in class (handwritten hardcopy).

You can work in pairs. If so, turn in one copy and write both participants.

1. For the following graph




(a) Draw the spanning forest after every iteration of the main loop in Kruskal's algorithm.

(b) Draw the spanning forest after every iteration of the main loop in Prim's algorithm.

(c) Find the shortest path spanning tree rooted in A.

2. Modify Prim's algorithm so that it runs in time O(nlogk) on a graph that has only k different edges costs.

3. Let G=(V,E) be an undirected weighted graph, and let T be the shortest-path spanning tree rooted at a vertex v. Suppose now that all the edge weights in G are increased by a constant number k. Is T still the shortest-path spanning tree from v?

7. 4. If the minimum spanning tree of the graph is unique. Is the path between a pair of vertices in a minimum spanning tree of an undirected graph necessarily the shortest (minimum weight) path? Prove or provide a counter example.

8.

9. 5. Let G=(V,E) be a directed weighted graph such that all the weights are positive. Let v and w be two vertices in G and k≤|V| be an integer. Design an algorithm to find the shortest path from v to w that contains exactly k edges. Note that the path need not be simple.

6. *Multisets* are allowed to have repeated elements. A *multiset* of n items may thus have fewer than n! distinct permutations. For example, {1,1,2,2} has only six different permutations: {1,1,2,2}, {1,2,1,2}, {1,2,2,1}, {2,1,1,2}, {2,1,2,1}, and {2,2,1,1}. Design and implement an efficient algorithm for constructing all permutations of a *multiset*.