CIS 315, Intermediate Algorithms
Spring 2018

Assignment 3

due Monday, April 30, 2018

1. Consider the graph below. You will be building a MST for this graph in two ways. When there is a tie on the edge weights, consider the edges or nodes in alphabetical order. (For an edge, this means (a,f) before (b,c) and (c,d) before (c,g).)

   (a) Use Kruskal's method.

   (b) Use Prim's method.

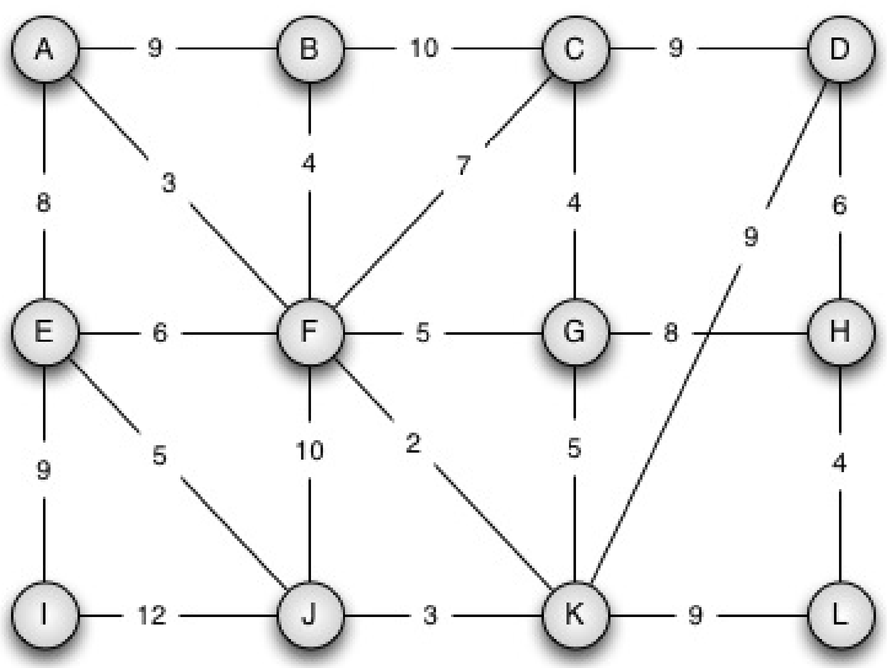

Figure 1: for question 1

   [8 points]

2. Modify Dijkstra's algorithm to count the number of shortest paths from the start node to each other node. It will still need to determine the length of the shortest path from the start node to each other node as well. **[5 points]**

3. Suppose you are given the diagram of a telephone network, which is a graph G whose vertices represent switching centers and whose undirected edges represent communication lines between two centers. The edges (u, v) are marked by their bandwidth $B[u, v]$. The bandwidth

of a path is the bandwidth of its lowest bandwidth edge. Give an algorithm that, given a diagram and two switching centers s and t, will output the bandwidth of a path with maximum bandwidth between s and t. [**6 points**]

4. We are given a graph $G = (V, E)$ where V represents a set of locations and E represents a communications channel between two points. We are also given locations $s, t \in V$, and a reliability function $r : V \times V \to [0, 1]$. You need to give an efficient algorithm which will output the reliability of the most reliable path from s to t in G.

   For any points $u, v \in V$, $r(u, v)$ is the probability that the communication link (u, v) will not fail: $0 \le r(u, v) \le 1$. Note that if there is a path with two edges, for example, from u to v to w, then the reliability of that path is $r(u, v) \cdot r(v, w)$. [**6 points**]

5. exercise 24.1-3 from CLRS
   Given a weighted, directed graph $G = (V, E)$ with no negative-weight cycles, let m be the maximum over all vertices $v \in V$ of the minimum number of edges in a shortest path from the source s to v. (Here, the shortest path is by weight, not the number of edges.) Suggest a simple change to the Bellman-Ford algorithm that allows it to terminate in $m + 1$ passes, even if m is not known in advance. [**5 points**]

Total: 30 points