CIS 315, Intermediate Algorithms
Spring 2019

Assignment 3

due Monday, April 29, 2019

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.

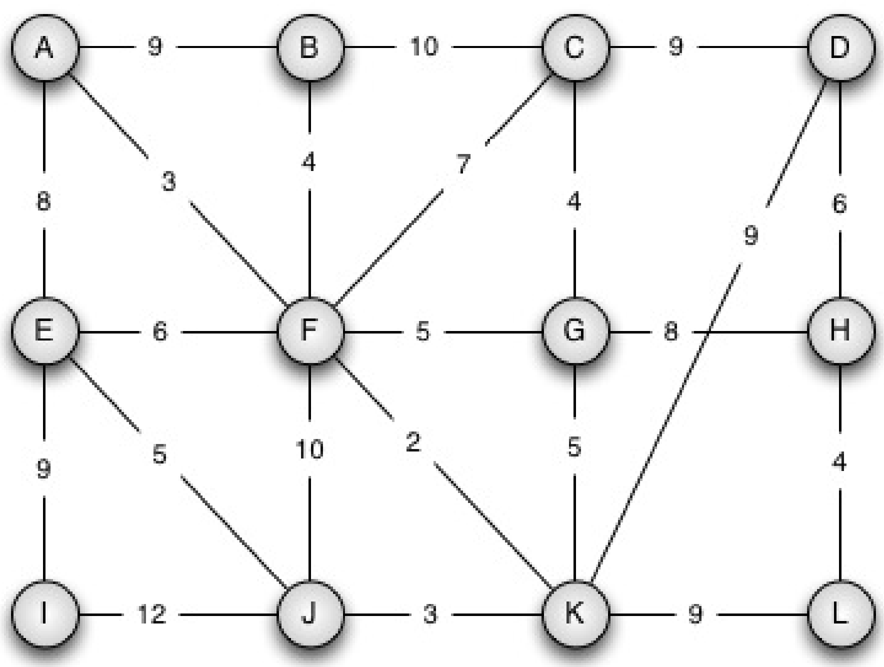

Figure 1: for question 1

   [8 points]

2. We are given a weighted graph $G = (V, E)$ with weights given by W. The nodes represent cities and the edges are (positive) distances between the cities. We want to get a car that can travel from a start node s and reach all other cities. Gas can be purchased at any city, and the gas-tank capacity needed to travel between cities u and v is the distance $W[u, v]$. Give an algorithm to determine the *minimum* gas-tank capacity required of a car that can travel from s to any other city **[6 points]**

3. 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: $0 \leq r(u, v) \leq 1$. 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]**

4. 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**]

5. exercise 25.2-4 from CLRS. More specifically, show that $O(n^2)$ space can be achieved **without sacrificing correctness** by dropping the superscripts in the Floyd-Warshall algorithm by computing the distance matrices $\mathcal{D}^{(k)}$ in place using a single matrix $\mathcal{D}$. [**5 points**]

Total: 30 points

note(s)

- For questions 2 and 3 you need only modify the Relax method and perhaps change the priority queue of Dijkstra's/Prim's method.