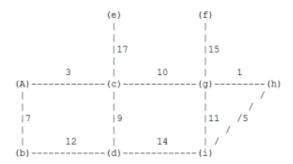
CS 477: Homework #7

Due on December 8th, 2016 at 2:30pm $Monica\ Nicolescu$

Matthew J. Berger

[20 points](U & G Required)

Answer the questions below regarding the following graph:



- a.) [5 points] In what order are edges added to the Minimum Spanning Tree (MST) using Kruskal's Algorithm? List the edges by giving their endpoints.
- b.) [5 points] In what order are edges added to the MST using Prim's Algorithm starting from vertex A? List the edges by giving their endpoints.

Solution

- a.) g-h, A-c, h-i, A-b, c-d, c-g, f-g, c-e
- b.) A-c, A-b, c-d, c-g, g-h, h-i, f-g, c-e

[30 points](U & G Required) Exercise 22.2-9 (page 602).

22.2-9) Let G = (V, E) be a connected, undirected graph. Give an $\mathcal{O}(V + E)$ -time algorithm to compute a path in G that traverses each edge in E exactly once in each direction. Describe how you can find your way out of a maze if you are given a large supply of pennies.

[30 points](U & G Required) Exercise 22.5-6 (page 621).

22.5-6) Given a directed graph G = (V, E), explain how to create another graph G' = (V, E') such that

- a.) G' has the same strongly connected components as G
- b.) G' has the same component graph as G
- c.) E' is as small as possible.

Describe a fast algorithm to compute G'.

[20 points](U & G Required) Exercise 24.3-2 (page 663).

24.3-2) Give a simple example of a directed graph with negative-weight edges for which Dijkstras algorithm produces incorrect answers. Why doesn't the proof of Theorem 24.6 go through when negative-weight edges are allowed?

[20 points](Extra Credit) Exercise 24.3-6 (page 663).

24.3-6) We are given a directed graph G=(V,E) on which each edge $(u,v)\in E$ has an associated value r(u,v), which is a rela number in the range $0\leq r(u,v)\leq 1$ that represents the reliability of a communication channel from vertex u to vertex v. We interpret r(u,v) as the probability that the channel from u to v will not fail, and we assume that these probabilities are independent. Give an efficient algorithm to find the most reliable path between two given vertices.