CS 242: Data Structures and Algorithms

Take Home Part of Final Examination

(Due December 17th, Thursday, in class.)

Name: ____________________ *Total Points: 150*

This exam is **take-home, open-book, and closed-person**. *You may consult any textbook and your notes, run simulations on the computer, but you may not consult any persons. The intent is that you should attempt to solve these problems on your own. I am available for questions of interpretation.*

This exam is due on 17th December 1998, at 1:00pm in class before the in-class final begins. **Absolutely no extensions will be granted!**

Some of these problems are meant to challenge you. Do not worry if you do not get a complete solution. If you do not get the solution to a problem, write out a **succinct** *outline of your thoughts on how to solve the problem. Partial credit will be given, if the partial solution is presented logically, coherently, and legibly. Random ramblings done with a dull pencil and stream-of-consciousness literary efforts will receive no credit. Please write legibly or type on one side of a page.*

There are three problems in this take home exam.

Please attach the take home exam to your solutions before returning the solutions.

I. (50 Points) Singly-connected graphs. A directed graph $G = (V, E)$ is **singly connected** if $u \leadsto v$ implies that there is at most one simple path from u to v for all vertices $u, v \in V$. Give an algorithm to determine whether or not a directed graph is singly connected. Carefully argue the correctness of your algorithm. Also analyze the worst-case run-time of your algorithm in terms of m (the number of edges) and n (the number of vertices).

(*Hint:* Consider the dfs (depth-first search) forest of a singly connected graph. What properties does it have? Can you use these properties in recognizing singly connected graph?)

II. (50 Points) Most Reliable Path. 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 real number in the range $0 \le r(u, v) \le 1$ 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. We interested in finding the most reliable path between two given vertices.

- How would you modify Dijkstra's shortest path algorithm to find the most reliable path between two given vertices? (30 points)

- Suppose you are given an implementation of Dijkstra's shortest path algorithm. Describe how you would modify the input graph such that running the *unmodified* Dijkstra's algorithm would allow us to find out the most reliable path between two given vertices. (*Note: if you can answer this part correctly, then you don't need to answer the first part.*) (20 points)

III. (50 Points) Updating a Minimum Spanning Tree. We are given a connected undirected graph $G = (V, E)$ with weighted edges and a minimum spanning tree (MST) T of G. Assume that the weight of an edge (u, v) in the minimum spanning tree T changes.

- Under what conditions will T remain an MST of G? (15 points)

- Design an algorithm MST-Update$(G, T, (u, v), w)$, in which G is a graph with weighted edges, T is an MST of G, (u, v) is an edge in T, and w is the new weight for edge (u, v). This algorithm should return a minimum spanning tree T' for the new graph G' consisting of G with the weight of (u, v) changed to w. You may assume that T is given as an undirected graph represented in adjacency list format. Your algorithm should be asymptotically faster than simply computing a new minimum spanning tree from scratch. (35 points)