Fundamental Algorithms

Homework 7

Due: July 26th, 5:30 PM EST

Instructions

Please answer each **Problem** on a separate page. Submissions must be uploaded to your account on Gradescope by the due date and time above.

Problems To Submit

Problem 1 (25 points)

(a) Assume that all edge weights of an undirected connected graph G are 1. Develop an $O(|V|+|E|)$-time algorithm to find the minimum spanning tree (MST) of G. Justify the correctness of your algorithm and its run-time.

(b) Now assume that all the edge weights are 1, except for a single edge $e_0 = \{u_0, v_0\}$ whose weight is w_0 (note that w_0 might be either larger or smaller than 1). Show how to modify your solution in part (a) to compute the MST of G. What is the run-time of your algorithm and how does it compare to the run-time you obtained in part (a)?

Problem 2 (25 points)

Suppose we are given an undirected graph G with weighted edges and a minimum spanning tree T of G. In each of the following cases, we change the weight of one edge e of G, and obtain the new undirected weighted graph G'. Justify why T remains an MST of the new graph G' in each of these cases.

(a) The weight of one edge $e \in T$ is decreased.

(b) The weight of one edge $e \notin T$ is increased.

Problem 3 (25 points)

Provide a counterexample to the following false claim. Fully justify your answer.
If a directed graph G contains cycles (i.e., is not acyclic), then topological sort produces a vertex ordering that minimizes the number of "bad" edges that are inconsistent with the ordering produced. More precisely, a bad edge is an edge going from a vertex later in the ordering to an earlier vertex.

Problem 4 (25 points)

A directed graph G is called *semi-connected* if, for all pairs of vertices u and v in G, we have a path from u to v **or** a path from v to u (or both).

Note: Recall that in a connected graph, we must have a path from u to v **and** a path from v to u.

(a) Develop an algorithm which, given a directed graph $G = (V, E)$, determines whether G is semi-connected. Your algorithm must run in $O(|V| + |E|)$ time.

(b) Justify the correctness and run-time of your algorithm.

Hint: First, solve the problem when G is acyclic (DAG). Then, generalize your solution by noting that any directed graph can be decomposed into a DAG of strongly connected components (as seen in the lecture).

Practice Problems

Problem 1

A mother vertex in a directed graph $G = (V, E)$ is a vertex from which there are paths to all other vertices in the graph. Give an $O(|V| + |E|)$ time algorithm to find a mother vertex of G, if it exists.

Problem 2

Find a weighted graph with some edges assigned a negative weight on which Dijkstra's Algorithm fails. Justify your answer.