CMPSCI 311: Introduction to Algorithms **Spring 2017**

Homework 3

Your Name: ____________________

Collaborators and sources: ______________________________________

You make work in groups, but you must write solutions yourself. List collaborators on your submission.

If you are asked to design an algorithm, please provide: (a) the pseudocode or a *precise* English description of the algorithm, (b) an explanation of the intuition for the algorithm, (c) a proof of correctness, (d) the running time of your algorithm and (e) justification for your running time analysis.

Submission instructions. This assignment is due by 8:00pm on Friday, March 10 in Gradescope (as a pdf file). Please review the course policy about Gradescope submissions on the course website.

NOTE: *Before doing problems 2 and 3, read about the Cycle Property starting on the lower half of page 147 in the book. You will need to use the Cycle Property in both problems.*

1. **(10 points) Connectivity in Weighted Graphs.** Let $G = (V, E, W)$ be a connected weighted graph where each edge e has an associated non-negative weight $w(e)$. We call a subset of edges $F \subset E$ *unseparating* if the graph $G' = (V, E \setminus F)$ is connected. This means that if you remove all of the edges F from the original edge set, this new graph is still connected. For a set of edges $E' \subset E$ the *weight* of the set is just the sum of the weights of the individual edges that is $w(E') = \sum_{e \in E'} w(e)$. Give a polynomial time algorithm to find the unseparating set with maximum weight.

2. **(20 points) K&T Chapter 4, Exercise 9.** This problem explores a variation of the network design problem: designing a spanning network for which the *most expensive* edge is as cheap as possible. Specifically, let $G = (V, E)$ be a connected graph with n vertices, m edges, and positive edge costs that you may assume are all distinct. Let $T = (V, E')$ be a spanning tree of G; we define the *bottleneck edge* of T to be the edge of T with the greatest cost.

   A spanning tree T of G is a minimum-bottleneck spanning tree if there is no spanning tree T' of G with a cheaper bottleneck edge.

   (a) Is every minimum-bottleneck tree of G a minimum spanning tree of G? Prove or give a counterexample.

   (b) Is every minimum spanning tree of G a minimum-bottleneck tree of G? Prove or give a counterexample.

3. **(20 points) K&T Chapter 4, Exercise 10.** (minor variation in part (b))

   Let $G = (V, E)$ be an (undirected) graph with costs $c_e \geq 0$ on the edges $e \in E$. Assume you are given a minimum-cost spanning tree T in G. Now assume that a new edge is added to G, connecting two nodes v and w with cost c. You may assume that edge costs are distinct.

   (a) Give an efficient algorithm to test if T remains the minimum-cost spanning tree with the new edge added to G (but not to the tree T). Make your algorithm run in time $O(|E|)$. Can you do it in $O(|V|)$ time? Please note any assumptions you make about what data structure is used to represent the tree T and the graph G

   (b) Suppose T is no longer the minimum-cost spanning tree. Give a linear-time algorithm (time $O(|E|)$) to update the tree T to a lower cost spanning tree.

4. **(0 points).** How long did it take you to complete this assignment?