Homework 5
Com S 311
Due: Nov 3, 11:59 pm

5% bonus for submission by Nov 2, 11:59 pm

No late submissions will be accepted

Note: The "Do not grade" option is not available for this HW

For all algorithm design problems, part of the grade depends on efficiency. For every graph $G = (V, E)$, the vertex set is $\{1, 2, \cdots, n\}$. We use n to denote the number of vertices and m to denote number of edges. Unless otherwise stated, all graphs are represented as adjacency lists. For weighted graphs, assume that each entry in the adjacency list for a vertex u is a tuple of the form $(v, c(u, v))$, where $c(u, v)$ denotes the cost associated with the edge (u, v).

1. (40 points) Let $G = (V, E)$ be an undirected, connected and weighted graph and let T be a minimum spanning tree (MST) for G. Let G' be a graph obtained by subtracting some positive value δ from the weight of an edge $e \in E$ that is not in T. Give an algorithm that, given G, T, e and δ, computes a MST of G'. Prove correctness of your algorithm and state its runtime.

2. (20 points) Let $G = (V, E)$ be a weighted, undirected graph and let S be a subset of V. We define $MST(G, S)$ to be a minimum spanning tree for G with the additional property that every vertex in S is a leaf node. Give an algorithm that gets G and S as input and outputs $MST(G, S)$, if it exists. State the runtime of your algorithm. *If you wish, you may assume that all edge weights are distinct (as the text does when proving the Cut Property (4.17)).*

3. (20 points) Let $G = (V, E)$ be a directed, weighted graph with positive edge weights. You are given an integer array d of length n, where n is the number of vertices of G, and you are given a vertex s. You can assume that every vertex is reachable from s. Come up with an $O(m)$ algorithm to determine the following: Is it the case that for every vertex v, $d[v]$ is the length of the shortest path from s to v? Here "length" and "shortest" refer to the sum of the edge weights for the path, not the number of edges in the path. (*Tip:* Remember that if $s, u_1, u_2, \ldots, u_k, v$ is a shortest path from s to v, then $s, u_1, u_2, \ldots, u_k$ must be a shortest path from s to u_k.)

4. (40 points) A bakery gets orders for many different kinds of cakes with fancy decorations. They have plenty of guys who can decorate cakes (at least n of them), and plenty of supplies for decoration, but there is only one oven, and it can only hold one cake at a time! Each cake

C_i requires b_i minutes of baking time by itself in the oven, plus d_i minutes of decoration time, which can occur in parallel with other cakes being decorated or baked. Write an algorithm that, given a set of n cakes, determines a schedule $C_{i_1}, C_{i_2}, \ldots, C_{i_n}$ that such that all the cakes are completely finished in the shortest overall time. Derive the runtime and prove the correctness of your algorithm.

GUIDE LINES:

- Please write your recitation number, time and TA name.
- You must work on the homework problems on your own. You should write the final solutions alone, without consulting any one. Your writing should demonstrate that you understand the proofs completely.
- When proofs are required, you should make them clear and rigorous.
- Any concerns about grading should be made within one week of returning the homework.
- **Please submit your HW via Canvas. If you type your solutions, then please submit pdf version. If you hand-write your solutions, then please scan your solutions and submit a pdf version. Please make sure that the quality of the scan is good, and your hand writing is legible. HW's submitted in incorrect format (non pdf) may incur a penalty of 20%**
- If you hand writing is not legible or the quality of the scan is poor, your homework will not be graded.