CS 535 Design and Analysis of Algorithms	**Fall Semester, 2021**

Extra set of problems

Assigned: October 6 — *Will not be collected*

Variations of the following problems might appear on the midterm. No solutions for this set of problems will be posted (unless a problem becomes part of another homework, which may happen).

Problem 1 Let $G = (V, E, c)$ be a weigted undirected graph where all the costs c_e, for $e \in E$, are strictly positive and distinct. Let T be a minimum spanning tree in $G = (V, E, c)$. Now suppose we replace the cost of each edge $e \in E$ by $c'_e = c_e^2$, creating the instance $G' = (V, E, c')$. Prove or disprove: T is a minimum spanning tree in G'.

Now assume $s, t \in V$ are also given, and P is a shortest $s - t$ path in G. Prove or disprove: P is the shortest $s - t$ path in G'.

Problem 2 Consider the Union-Find data structure presented in class. Suppose that we wish to add the operation PRINT-SET(x), which is given a node x and prints all the members of x's set, in any order. Show how we can add just a single attribute to each node in a disjoint-set forest so that PRINT-SET(x) takes time linear in the number of members of x's set and the asymptotic running times of the other operations are unchanged. Assume that we can print each member of the set in $O(1)$ time.

Give pseudocode as in the textbook for MAKE-SET, UNION, FIND, PRINT-SET (if exactly the same, just say so). You do not have to prove correctness for this problem (but the pseudocode must be correct). Do argue the bounds on the running time.

Problem 3 Given a $O(|V| + |E|)$ algorithm to determine if an undirected graph $G = (V, E)$ is bipartite. Prove that your algorithm is correct. The definition of a bipartite graph apears in the textbook. Hint: modifiy BFS or DFS.

Problem 4 Given a directed graph $G = (V, E)$, explain how to create another graph $G = (V, E)$ such that, for any two vertices u and v, there exist a $u - v$ directed path in G if and only if there exists a $u - v$ directed path in G', and such that $|E'|$ is as small as possible. Describe a polynomial-time algorithm. Note: E' does not have to be a subset of E.

Hint 1: Consider the case where G is strongly connected first.

Hint 2: Consider the case where G is acyclic and use its topological sort.