Name(last, first): ___________________

U C L A Computer Science Department

CS 180 **Algorithms & Complexity** **UID : ______________**

Midterm **Total Time: 90 minutes** **October 27, 2022**

Each problem has 20 points: 5 problems, 5 pages (upload ONE pdf that has at most 2 pages per problem to gradescope and then hand in your exam to me).

For all 5 problems: algorithms should be described in bullet point format (<u>with justification/proof</u>). You need to prove the correctness of your algorithm. You need to analyze its time complexity with proof.

Problem 1: Consider a DAG G=(V, E).
a. (10 points) Describe the topological sort algorithm on G.
b. (10 points) Analyze its complexity.

Problem 2: Consider a DAG G with n vertices and e edges. Its longest path is of length K.
A. Design an O(e+n) algorithm that partitions the vertices into at most K+1 groups such that there are no edges within a group.
B. Prove the correctness of the algorithm (you do NOT need to analyze its time complexity).

Name(last, first): ____________________

Problem 3
Let $x_1, x_2, \ldots, x_n$ be a sequence of integers (not necessarily positive and not sorted).

A. (16 points) Design an O(n) algorithm to find the subsequence $x_i, \ldots, x_j$ (of consecutive elements) such that the product of the numbers in it is maximum over all consecutive subsequences. The product of the empty subsequence is defined as 1. You need to prove its correctness as well.

B. (4 points) analyze the time complexity of your algorithm.
Example: If the original sequence is -4 -3 4 1 -5 the answer is $(-3)\times4\times1\times(-5)$ or +60.

Name(last, first): ___________________

Problem 4:
(20 points) Consider the greedy algorithm of finding a maximum subset of non-overlapping intervals that we covered in class. **Prove** that the algorithm indeed maximizes the resulting subset of intervals (that is, prove the correctness of the algorithm). You can assume all start and end points are unique.

Problem 5:
Let G = (V, E) be a connected undirected weighted graph with **n** vertices. Assume for simplicity that the weights are positive and distinct. Let **e** be an edge of G. Denote by T(**e**) the minimum spanning tree MST of G that has minimum cost among all spanning trees of G that contain **e**.

a. (8 points) Design an $O(n^2)$ algorithm that finds T(**e**) given a graph G and an edge **e**.
b. (4 points) Prove its correctness.
c. (8 points) Design an algorithm to find T(**e**) for all edges **e** in E. The algorithm should run in time $O(n^2)$.

(You do **not** need to re-prove that finding an MST on a graph can be done in $O(n^2)$ time – we already proved that in class).