Name(last, first): _____

U C L A Computer Science Department

CS 180 **Algorithms & Complexity** **ID : _____**

10 pm Midterm **Total Time: 1.5 hours** **November 2, 2020**

Each problem has 20 points .

All algorithm should be described in bullet format (with justification/proof). You cannot quote any time complexity proofs we have done in class: you need to prove it yourself.

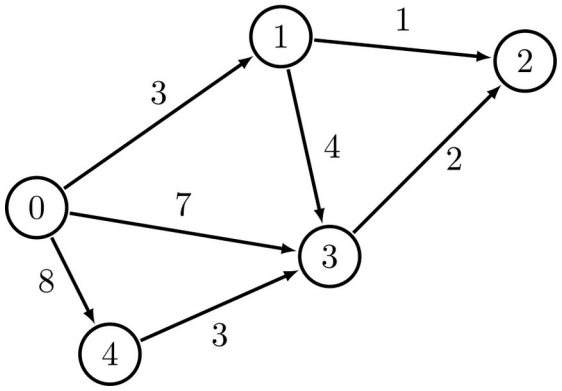
Problem 1: Describe the Depth First Search algorithm in a DAG. Prove its correctness. Analyze its complexity.

Problem 2: Consider a set of intervals/tasks. A. Design an algorithm that finds the maximum number of non-overlapping intervals/tasks. B. Analyze the time complexity of your algorithm. C. prove the correctness of your algorithm.

Problem 3: Design an algorithm that decides if a connected undirected graph is bipartite. And finds a two sets of vertices.
(Definition: A bipartite graph is one where can partition the vertices into two groups V1 and V2 such that there are not edges within V1 and within V2: all edges are between one vertex in V1 and one vertex on V2).

Name(last, first): _____

Problem 4: Apply Dijsktras shortest path algorithm to the graph shown below (step by step) starting from vertex zero (0). Analyze the time complexity of the algorithm (you can assume insertion and deletion in a heap takes O(log n) time) .



Problem 5: Let G be an undirected binary tree with n vertices (and no weights) . The distance between two vertices is the length of the shortest path between the two vertices. We want to construct an nxn matrix such that its ij-th entry is the distance between two vertices vi and vj. Design an O(n^2) algorithm for doing this. Describe your algorithm,analyze its time complexity and prove its correctness.