Name(last, first): ___________________

U C L A Computer Science Department

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

____

Midterm **Total Time: 1.5 hours** **October 31, 2017**

Each problem has 20 points.

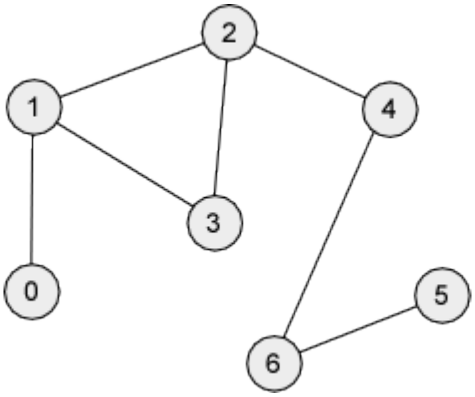
All algorithm should be described in English, bullet-by-bullet

1. Consider a set of intervals [I_1, …, I_n]. **a.** Design a linear time algorithm (assume intervals are sorted in any manner you wish) that finds a maximum subset of mutually non-overlapping intervals. **b.** Prove the correctness of your algorithm.

.**2 . a.** Design an efficient algorithm better than O (n^2) to be used in sparse graphs for finding the shortest path between two vertices S and T in a positive weighted graph. **b**. Justify the correctness of your runtime analysis.

3. Consider a sequence of positive and negative (including zero) integers. Find a consecutive subset of these numbers whose sum is maximized. Assume the weight of an empty subset is zero. **a.** Design a linear time algorithm. **b**. Prove the correctness of your algorithm.
Example: For the sequence [4, -3, 5, -12] the maximum sum is 6.

4. Consider an unweighted graph G shown below. **a.** Starting from vertex **4,** show every step of DFS along with the corresponding stack next to it. **b**. What is the run time of DFS if the graph is not connected (no proof is necessary)?



5. Consider a binary tree (it is not necessarily balanced). The tree is not rooted. Its diameter is the distance between two vertices that are furthest from each other (distance is measured by the number of edges in a simple path). Design a linear time algorithm that finds the diameter of a binary tree.