Name(last, first): _____

ID (rightmost 4 digits): _ _ _ _

U C L A Computer Science Department

CS 180 **8 am** **Algorithms & Complexity**

Final Exam **Total Time: 3 hours** **December 16, 2018**

***** Write all algorithms in bullet form (as done in the past) *****

You need to prove EVERY answer that you provide.

There are a total of 8 pages including this page.

You need to upload ONE file in PDF to Gradescope.

You can include at most 15 pages in your PDF.

Name(last, first): _____

1. (20 points: each part has 10 points)

a. Consider an instance of the closet pair problem. In the merge step, for every point on the left, how many points do we consider and compare? Prove your answer.

b. Consider an instance of the closest pair problem. How do we maintain a list of points sorted in the y-direction at each step? Discuss and prove the details.

2. (20 points) a. Consider a weighted connected graph G and a Minimum spanning tree T on G. Assume a new vertex X and a set of weighted edges from X to other vertices are added to G. The new graph is called G'. To find an MST of G' can we just focus on edges of T and the newly added weights? (that is, can we ignore all edges of G that are not part of T?) Prove your answer.

b. We have a weighted graph G. We increase each weight by a number K to obtain a graph G'. We then find an Minimum Spanning Tree T' of G'. We decrease weight of each edge in T' by K to obtain a tree T. Is T a minimum spanning tree of G? Prove your answer.

3. (15 points: Each part has 10 points)

Consider a sorted sequence A = (a1, a2, an). B is obtained from A by a cyclic shift. So B is (aj, a(j+1), …, an, a1, a2, … a(j-1)). Design an O(log n) time algorithm that finds if an element X is in B.
Example: If A= (2 , 4 , 7, 9) then a cyclic shift of it can be B = (7, 9, 2, 4).

4. (15 points)

Use dynamic Programming: Given an array of integers A[], find maximum value of the expression

(A[s] – A[r] + A[q] – a[p]) such that s > r > q > p

Example A[] = [3,9,10,1,30,40] Maximum value is 46 : (40 – 1 + 10 – 3)

5. (15 points) A **Hamiltonian path** in a graph with **n** vertices is a path of length **n−1**, i.e., it is a path that visits all vertices of the graph exactly once (which also means no edges can be repeated). Hamiltonian path problem is known to be NP-Complete.
An ST-Hamiltonian path problem is a version of the Hamiltonian path problem where we have to start at given vertex **s** and end at a given vertex **t**. Prove that ST-Hamiltonian path problem is also NP-Complete.

6. (15 points) N teams attend a dinner. Team **i** has **ti** members. There are **M** tables at the dinner, with **M ≥ N**. Table **i** has **ci** chairs. We wish to seat all teams such that no two team members are at the same table. Design an algorithm for solving this problem. Prove its correctness. Analyze its time complexity.