Name(last, first): _____

ID (rightmost 4 digits): _ _ _ _

U C L A Computer Science Department

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

Final Exam **Total Time: 2.5 hours** **December 16, 2020**

***** 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. What is the run time of union-find? Discuss details of the data structure and prove your answer.

b. Prove that the k-clustering problem discussed in class is optimal.

2. (20 points)

a. Find necessary and sufficient conditions for an undirected graph to be partitionable into a set of disjoint cycles C1 to Ck. (two cycles are disjoint if they do not share an edge). Prove the correctness of your answer.

b. Show an example of graph on 5 vertices that satisfies the above condition.

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

Consider two arrays of strings A and B each with n elements. Design a linear time algorithm to decide if B is a cyclic shift of A. Cyclic shift means you repeatedly move the last element of an array to the front the array.

For example if A = (a a c b) then its cyclic shifts are (b a a c), (c b a a) , (a c b a). And (b c a a) is not a cyclic shift of A.

4. (15 points)

Use dynamic programming to find a subsequence of a given sequence such that the subsequence sum is maximized and the subsequence elements are in increasing order. The subsequence is not necessarily continuous.

Example: Given a sequence (0, 8, 4, 12, 2, 9) The best subsequence is (8, 12) with max sum being 20 = 8 + 12

5. (15 points)
From the class we know that max-clique problem is NP-complete. Consider the class of graphs where degree of all vertices is the same. Is max-clique also NP-complete on this class of graphs? Prove your answer.

6. (15 points) An **edge cover** in a graph is a subset of edges such that the union of edge endpoints corresponds to the entire vertex set of the graph.

Design an algorithm for finding a minimum edge cover in a bipartite graph. Describe the algorithm, prove its correctness and analyze its time complexity.