

Name: _____

ID No: _____

COT 5405: Analysis of Algorithms

Exam II; March 30, 2000

Note: You are supposed to give proofs to the time bounds of your algorithms. Read the questions carefully before attempting to solve them.

1. (18 points) Present a greedy approach for the solution of the traveling salesperson problem. Is your algorithm optimal? If so, prove it. If not, how bad can your algorithm's performance be (i.e., what is the maximum value of $\frac{YC}{OC}$ where YC is the total cost of the tour found by your algorithm and OC is the cost of the optimal tour)? What is the run time of your algorithm?

2. (15 points) Let T be a minimum cost spanning tree (MCST) of a weighted undirected graph $G(V, E)$. If the weight on every edge is now increased by the same value c , will T continue to be a MCST of G ? If yes, prove your answer; if not how will you modify T into a MSCT of G ?

3. (18 points) The SubsetSum problem takes as input a set $X = \{k_1, k_2, \dots, k_n\}$ of integers and another integer K . The problem is to check if there exists a subset X' of X whose elements sum to K . For example, if $X = \{5, 3, 11, 8, 2\}$ and $K = 16$ then the answer is YES since the subset $X' = \{5, 11\}$ has a sum of 16. Present a dynamic programming algorithm for SubsetSum whose run time is $O(nK)$.

4. (16 points) The string editing problem can be extended to three (or more) sequences as follows. We still consider three operations viz., INSERT, DELETE, and CHANGE. In each step, the cost is zero if the corresponding symbols in all the sequences are the same and 1 otherwise (even if two sequences match and only one insertion, deletion, or change is necessary). As an example, consider the sequences $aabb$, bbb , and cbb . One possible edit sequence is inserting a in front of bbb and cbb (which costs 1), and replacing a b in bbb & a c in cbb with an a (for a cost of 1). The total cost is 2. Present an $O(n^3)$ time algorithm to compute the minimum edit cost between three given sequences each of length n .

5. (18 points) An undirected graph $G(V, E)$ is said to be *bipartite* if V can be partitioned into disjoint sets V_1 and V_2 such that no two nodes in V_1 have an edge between them and no two nodes in V_2 have an edge connecting them. (I.e., the only edges in G are from nodes in V_1 to nodes in V_2 .) Present an $O(|V| + |E|)$ time algorithm to check if G is bipartite.

6. (15 points) Input is a weighted undirected graph $G(V, E)$ in which each edge has the same weight w . Present an $O(|V| + |E|)$ time algorithm to solve the single source shortest path problem from the source node $s \in V$.

7. **** Extra Credit **** (10 points) Input is a directed acyclic graph $G(V, E)$ with n vertices. The problem is to label the vertices from 1 to n such that if v is labeled k then all the vertices that can be reached from v by a directed path are labeled with labels $> k$. This problem is known as the *topological sorting problem*. Present an $O(|V| + |E|)$ time algorithm for topological sorting of G .