CSE361 Computational Complexity of Algorithms. Spring 2008

Home work 2. Due on April 7 (3:30 PM).

1. (a) Find an optimal solution to the knapsack instance $n = 7, m = 20$, $(p_1, p_2, \ldots, p_7) = (12, 8, 5, 11, 8, 12, 6)$, and $(w_1, w_2, \ldots, w_7) = (4, 5, 2, 8, 3, 6, 2)$.

   (b) Let $F(I)$ be the value of the solution generated on problem instance I by GreedyKnapsack when the objects are input in nonincreasing order of the p_i's. Let $F^*(I)$ be the value of an optimal solution for this instance. How large can the ratio $F^*(I)/F(I)$ get?

   (c) Answer (b) for the case in which the input is in nondecreasing order of the w_i's.

2. Let $G(V, E)$ be any weighted connected graph. If C is any cycle of G, then show that the heaviest edge of C cannot belong to a minimum-cost spanning tree of G.

3. Input is a weighted directed graph $G(V, E)$, where the edge weights are in the range $[1, C]$ (C being a constant). Present an $O(|V| + |E|)$ time algorithm to solve the single-source shortest paths problem on this graph.

4. Modify the function AllPaths so that a shortest path is output for each pair of vertices $(i,\ j)$. What are the time and space complexities of the new algorithm?

5. Given a sequence X of symbols, a subsequence of X is defined to be any contiguous portion of X. For example, if $X = x_1, x_2, x_3, x_4, x_5$, then x_2, x_3 and x_1, x_2, x_3 are subsequences of X. Given two sequences X and Y, present an algorithm that will identify the longest subsequence that is common to both X and Y. This problem is known as *the longest common subsequence problem*. What is the time complexity of your algorithm?

6. Let $M_1 \times M_2 \times \cdots \times M_r$ be a chain of matrix products. This chain may be evaluated in several different ways. Two possibilities are $(\cdots((M_1 \times M_2) \times M_3) \times M_4) \times \cdots) \times M_r$ and $(M_1 \times (M_2 \times (\cdots \times (M_{r-1} \times M_r) \cdots)$. The cost of any computation of $M_1 \times M_2 \times \cdots \times M_r$ is the number of multiplications used. Let M_{ij} denote the matrix product $M_i \times M_{i+1} \times \cdots \times M_j$. Let $D(i), 0 \leq i \leq r$, represent the dimensions of the matrices, i.e., M_i has $D(i-1)$ rows and $D(i)$ columns. Let $C(i, j)$ be the cost of computing M_{ij} using an optimal product sequence for M_{ij}. Observe that $C(i, i) = 0, 1 \leq i \leq r$, and that $C(i, i+1) = D(i-1)D(i)D(i+1), 1 \leq i \leq r$.

   (a) Obtain a recurrence relation for $C(i, j), j > i$.

   (b) Write an algorithm to solve the recurrence relation of (a) for $C(1, r)$. Your algorithm should be of complexity $O(r^3)$.

7. Present an $O(|V|)$ time algorithm to check whether a given undirected graph $G(V, E)$ is a tree. The graph G is given in the form of an adjacency list.