CSE431/531, Problem Set 3 Due Monday Oct. 20, in class

Note: There are 5 problems with a total of 100 points. You are required to do all the problems.

1. (20 points) Let M be an $n \times n$ matrix with each entry having a real number. Design a dynamic programming algorithm to find a longest sequence $S = (m_{i_1 j_1}, m_{i_2 j_2}, \cdots, m_{i_k j_k})$ such that $i_r < i_{r+1}$, $j_r < j_{r+1}$, and $m_{i_r j_r} > m_{i_{r+1} j_{r+1}}$ for all $1 \leq r < k$. You should make your algorithm run as fast as possible.

2. (20 points) Let P be a convex polygon with n vertices. A triangulation of P is an addition of a set of non-crossing diagonals (which connect non-neighboring vertices of P) such that the interior of P is partitioned by the set of diagonals into a set of triangles. The weight of each diagonal is the Euclidean distance of the two vertices it connects. The weight of a triangulation is the total weight of its added diagonals. Design a dynamic programming algorithm to find a minimum weighted triangulation of P. You should make your running time as short as possible.

3. (20 points) Let $G = (V, E)$ be a directed graph modeling a communication network. Each link e in E is associated with two parameters, $w(e)$ and $d(e)$, where $w(e)$ is a non-negative number representing the cost of sending a unit-sized packet through e, and $d(e)$ is an integer between 1 and D representing the time (or delay) needed for transmitting a packet through e. Design an algorithm to find a route for sending a packet between a given pair of nodes in G such that the total delay is no more than k and the total cost is minimized. Your algorithm should run in $O(k(|E| + |V|))$ time and $O(k|V|)$ space (additional to space for storing the graph).

4. (20 points) Let T be a rooted tree with n nodes and each node v associated with a weight $w(v)$. A subset of nodes S is an independent set of T if no node in S is the child or parent of another node in S. Design a dynamic programming algorithm to find an independent set of T with maximum weight, where the weight of an independent set is the total weight of its nodes. You should make your algorithm run as fast as possible.

5. (20 points) Let $A = a_1 \cdots a_m$ and $B = b_1 \cdots b_n$ be two strings with length m and n respectively. Design a dynamic programming based algorithm to convert A into B with minimum cost using the following rules. For a cost of 3, one can delete any letter from a string. For a cost of 5, one can insert a letter in any position. For a cost of 7, one can replace any letter by any other letter. For example, you can convert $A = abcabc$ to $B = abacab$ via the following sequence of operations: abcabc with a cost of 7 can be converted to abaabc, which with a cost of 3 can be converted to ababc, which with a cost of 3 can be converted to abac, which with a cost of 5 can be converted to abacb, which with a cost of 5 can be converted to abacab. Thus the total cost for this conversion is 23 (may not be the cheapest one).