COL 351

TUTORIAL SHEET 7

1. Suppose you own two stores, A and B. On each day you can be either at A or B. If you are currently at store A (or B) then moving to store B the next day (or A) will cost C amount of money. For each day i, $i = 1, \ldots, n$, we are also given the profits $P^A(i)$ and $P^B(i)$ that you will make if you are store A or B on day i respectively. Give a schedule which tells where you should be on each day so that the overall money earned (profit minus the cost of moving between the stores) is maximized.

2. Given a tree T where vertices have weights, an independent set is a subset of vertices such that there is no edge joining any two vertices in this set. Give an efficient algorithm to find an independent set of maximum total weight.

3. Given a tree $T = (V, E)$, where each vertex $v \in V$ has a weight w_v. Give a polynomial time algorithm to find the smallest weight subset of vertices whose removal results in a tree with exactly K leaves.

4. You are given N boxes, where box i has height h_i, width w_i and length l_i. Give an algorithm for finding a stacking of a subset of boxes of maximum total height : box i can be stacked on top of box j if $w_i < w_j$ and $l_i < l_j$.

5. A *bitonic* sequence of numbers $x_1, x_2, x_3 \ldots x_k$ is such that there exists an i, $1 \leq i \leq k$ such that $x_1, x_2 \ldots x_i$ is an increasing sequence and $x_i, x_{i+1} \ldots x_n$ is a decreasing sequence. It is possible that one of the sequences is empty, i.e., strictly increasing (decreasing) sequences are also considered bitonic. For example 3, 6, 7 , 5, 1 is a bitonic sequence where 7 is the discriminating number.
   Given a sequence of n numbers, design an efficient algorithm to find the longest *bitonic subsequence*. In 2, 4, 3 , 1, -10, 20, 8, the reader can verify that 2,3,20, 8 is such a sequence of length 4.

6. Given a convex n-gon (number of vertices is n), we want to triangulate it by adding diagonals. Recall that $n-3$ diagonals are required to triangulate. The *cost* of triangulation is the sum of the lengths of the diagonals added. For example, in a parallelogram, we will choose the shorter diagonal for minimizing cost. Design an efficient algorithm to find the minimum cost diagonalization of a given n-gon.