Design and Analysis of Algorithms

Practice-sheet 3 : **Dynamic Programming**

Date: *20 September, 2015*

1. (Monotonically increasing subsequence)

   Given a sequence $A = a_1, \ldots, a_n$, a subsequence $a_{i_1}, a_{i_2}, \ldots, a_{i_k}$ is said to be monotonically increasing if $a_{i_j} < a_{i_{j+1}}$ for all $1 \leq j < k$. Design an $O(n^2)$ time algorithm to compute the longest monotonically increasing subsequence of sequence A.

2. (Bellman Ford algorithm)

   Let $G = (V, E)$ be a directed graph on n vertices and m edges where each edge has a weight which is a real number. Show that there exists an order among the vertices such that if we process the vertices according to that order in the inner For loop of the Bellman-ford algorithm, then just after one iteration, $D[v]$ will store the distance from s to v.

3. (Bellman Ford algorithm)

   Given a directed graph $G = (V, E)$ on n vertices and m edges, our aim is to detect if there is any negative weight cycle in G. Design an $O(mn)$ time algorithm to compute one such cycle, if exists.

4. (Box stacking)

   Box Stacking. You are given a set of n types of rectangular 3-D boxes, where the ith box has height $h(i)$, width $w(i)$ and depth $d(i)$ (all real numbers). You want to create a stack of boxes which is as tall as possible, but you can only stack a box on top of another box if the dimensions of the 2-D base of the lower box are each strictly larger than those of the 2-D base of the higher box. Of course, you can rotate a box so that any side functions as its base. It is also allowable to use multiple instances of the same type of box.

5. (Edit Distance)

   Given two text strings A of length n and B of length m, you want to transform A into B with a minimum number of operations of the following types: delete a character from A, insert a character into A, or change some character in A into a new character. The minimal number of such operations required to transform A into B is called the edit distance between A and B. Design a polynomial time algorithm to compute edit distance between A and B.

6. (Floyd Warshal algorithm)

   Recall the Floyd Warshal algorithm discussed in the class. Your aim is augment this algorithm with an $O(n^2)$ size data structure which can store the all-pairs shortest paths information implicitly. The time complexity of the algorithm should still be $O(n^3)$. In addition, you have to design an algorithm *Report-shortest-path*(i, j) which outputs the shortest path from i to j using this data structure. The time taken by *Report-shortest-path*(i, j) has to be of the order of the number of edges on the shortest path from i to j.