15-451 Algorithms, Spring 2017
Recitation #6 Worksheet

Dynamic Programming

Off-Line Stock Market Problem You're given a sequence of stock prices $[p_1, p_2, \ldots, p_n]$. You want to find the maximum profit that you could have made on the stock in hindsight. In other words, you want to find i and j with $1 \leq i \leq j \leq n$ such that $p_j - p_i$ is maximal. Your algorithm should run in $O(n)$ time.

Longest Increasing Subseqeuence: Given an array A of n integers like `[7 2 5 3 4 6 9]`, find the longest subsequence that's in increasing order (in this case, it would be `2 3 4 6 9`). Give a dynamic-programming algorithm that runs in time $O(n^2)$ to solve this problem.

1. To keep things simple, first let's say you just need to output the *length* of the longest-increasing subsequence. E.g., in the above case, the length is 5.

   Hint: suppose that for each $i' < i$ you have computed the length of the LIS of $A_{0\ldots i'}$ that ends with $A[i']$. How would you use this to solve the corresponding problem for i?

2. Now extend your solution to actually find the LIS.

Closest Depot in a Tree: You're given a rooted tree T with n vertices. There are $m \leq n$ special vertices called *depots*. You are to compute, for every node v of T, the distance from v to the nearest depot. The distance is the number of tree edges that must be traversed to get there. (If v is a depot the distance is 0.)

Your algorithm should work by doing one or two depth-first searches (DFS) of the tree starting from the root, and it should run in $O(n)$ time.

Making Change: You are given denominations $v_1, v_2, \ldots, v_n$ (all integers) of the various kinds of currency you have. (Say $v_1 = 1$, so you can make change for any integer amount $C \geq 1$.) Given C, give a dynamic programming solution which makes change for C with the fewest bills possible.

(Again, as a first stab, compute the number of bills required, and then extend the solution to output the number of bills of each denomination needed.)

Making Change (Part II): Now suppose you have only one bill of each denomination i. Given C, give a dynamic programming solution which makes change for C using the fewest bills, using no more than one bill of each denomination i (or says this is not possible).

Making Change (Part III): Can you solve the problem if you have ℓ_i bills of denomination i?

Balanced Partition. You have a set of n integers each in the range $0, \ldots, K$. In time $O(n^2K)$, partition these integers into two subsets such that you minimize $|S_1 - S_2|$, where S_1 and S_2 denote the sums of the elements in each of the two subsets.