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 Subsequence: 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.

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

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 \( \ell_i \) bills of denomination \( i \)?

**Balanced Partition.** You have a set of \( n \) integers each in the range 0, \ldots, \( K \). In time \( O(n^2 K) \), 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.