Homework 4: Dynamic Programming

Due: October 10, 2023

Problem 1. You are given a roll (array) of n magical coins. Each coin has an initial value c_i which increases at each time step linearly. Therefore, at time step t_j the coin has a value of $c_i j$.

You want to sell the coins to maximize your total profit, however, you are only able to sell the coins one at a time. Furthermore, you are only able to sell the coins from either end of the roll of coins.

(a) You initially spring for a greedy approach to sell your magical coins, i.e. at each time step you compare both ends of the array and sell the coin with least value. Is this strategy flawed? Explain your reasoning.

(b) Design an algorithm which determines the optimal order to sell the magical coins to maximize profit, given that you know the starting value of each coin. Your algorithm should run in $O(n^2)$ time. Provide a proof of correctness for your algorithm, and justify its runtime and memory utilization.

Solution. □

Problem 2. Given an *unsorted* array of n positive integers, your task is to remove the least number of elements from the start or end of the array until twice the minimum of the array is larger than the maximum.

(a) Design a **dynamic programming** algorithm which runs in $O(n^2)$ time. Provide a proof of correctness for your algorithm, and justify its runtime and memory utilization.

(b) Suppose the integers are sorted in non-decreasing order. Design an algorithm that runs in $O(n)$ time. Provide a proof of correctness for your algorithm, and justify its runtime and memory utilization.

Solution. □

Problem 3. You are in charge of a botanical garden, and you're designing a pathway where visitors can admire a sequence of unique flower beds arranged in strictly increasing order of their charm. Each flower bed currently has a charm rating stored in the array A.

There is a problem; the current order of these ratings is not strictly increasing, which could disrupt the visitor experience. To correct this, you have access to a nursery full of replacement flowers, each with its own charm rating stored in another array, B. In one operation, you can choose any flower bed along the path and replace its charm rating with one from the nursery. It is guaranteed that $|B| \geq |A|$.

(a) Design a **dynamic programming** algorithm that determines the minimum number of operations to ensure the charm ratings in A are strictly increasing.

(b) Provide a proof of correctness of your algorithm.

(c) Justify your algorithm's runtime and memory utilization.

Solution. □