CSCI 570 Fall 2025
Homework 6

Problem 1

Suppose you have a rod of length N, and you want to cut up the rod and sell the pieces in a way that maximizes the total amount of money you get. A piece of length i is worth p_i dollars ($1 \leq i \leq N$). Devise a Dynamic Programming algorithm to determine the maximum amount of money you can get by cutting the rod strategically and selling the cut pieces. (Rod lengths are integers).

1. Define (in plain English) subproblems to be solved.
2. Write a recurrence relation for the subproblems.
3. Using the recurrence formula in part b, write pseudocode to find the solution. Make sure you specify
   (a) base cases and their values,
   (b) where the final answer can be found.
4. What is the complexity of your solution?

Problem 2

Alice and Bob are playing a turn-based game. It involves N stones placed in a row, numbered 0 to $N-1$ from the left to the right. Each stone i has a positive value s_i. On each player's turn, they can remove either the leftmost stone or the rightmost stone from the row and receive points equal to the sum of the remaining stones' values in the row. The winner is the one with the higher score when there are no stones remaining. Alice goes first in this game. Both players play to maximize their score by the end of the game. Devise a Dynamic Programming algorithm to return the maximum difference in score that Alice can achieve over Bob. Your algorithm must run in $O(N^2)$ time.

1. Define (in plain English) subproblems to be solved.
2. Write a recurrence relation for the subproblems.
3. Using the recurrence formula in part b, write pseudocode to find the solution. Make sure you specify
   (a) base cases and their values,
   (b) where the final answer can be found.
4. What is the complexity of your solution?

Problem 3

The Math Club consisting of n members hurries to line up in a straight line to take a group photo. Since the members are not positioned by height, the line is looking very messy. The club president decides to remove a few members so that the line follows a formation with the remaining members staying where they are. We say that k members remaining in the line with heights $r_1, r_2, \ldots, r_k$ are in formation if $r_1 < r_2 < \ldots < r_i > \ldots > r_k$, for some $1 \leq i \leq k$. For example, if the initial sequence of heights in inches is

$$(58, 60, 62, 65, 63, 61, 59, 57)$$

then, removing member #5 and #7 gives us the formation:

$$(58, 60, 62, 65, 61, 57)$$

Give an algorithm that runs in $O(n^2)$ time to find the minimum number of members to remove from the line, so that we get a formation as described.

1. Define (in plain English) subproblems to be solved.
2. Write a recurrence relation for the subproblems.
3. Using the recurrence formula in part b, write pseudocode to find the solution. Make sure you specify
   (a) base cases and their values,
   (b) where the final answer can be found.
4. What is the complexity of your solution?

Problem 4

Consider the 0-1 knapsack problem where you have infinitely many coins of each type. Namely, there are n different types of coins. Coins of type i have a weight of w_i and a value of v_i ($1 \leq i \leq n$). There is an unlimited supply of coins of each type. Design a dynamic programming algorithm to compute the optimal value you can achieve with a knapsack that has a maximum weight capacity of W.

1. Define (in plain English) subproblems to be solved.
2. Write a recurrence relation for the subproblems.
3. Using the recurrence formula in part b, write pseudocode to find the solution. Make sure you specify
   (a) base cases and their values,
   (b) where the final answer can be found.
4. What is the complexity of your solution?

Ugraded Problems

Solve Kleinberg and Tardos, Chapter 6, Exercise 5.

Solve Kleinberg and Tardos, Chapter 6, Exercise 6.