PSO 2 — Week 9: Problems

CS381, Fall 2026

Problem 1. Non-adjacent sums

You are given nonnegative numbers $w_1, w_2, \ldots, w_n$. You want to choose some of them, no two adjacent, so that the chosen numbers have the largest possible total.

(a) The numbers are in a row, so w_i is adjacent to w_{i-1} and w_{i+1}. Give an $O(n)$ dynamic programming algorithm, prove its recurrence correct, and show how to recover which numbers to choose.

(b) Now the numbers are arranged in a circle, so w_n and w_1 are adjacent as well (assume $n \geq 3$). Give an $O(n)$ algorithm. (Hint: w_1 and w_n cannot both be chosen.)

Problem 2. Cutting a stick

A stick of length L has to be cut at the positions $c_1 < c_2 < \cdots < c_m$, measured from its left end. The cuts are made one at a time, in any order you like, and each cut costs the length of the piece it is made in. The order matters: with $L = 10$ and cuts at 4, 5 and 6, cutting at 5 first costs 10 and leaves two pieces of length 5, whose cuts cost 5 each, for a total of 20; cutting at 4, then 6, then 5 costs only $10 + 6 + 2 = 18$.

Give an $O(m^3)$-time dynamic programming algorithm that finds the minimum total cost, and explain why it is correct. (Hint: what is left to do after the first cut?)