CMPSC 465 — Data Structures & Algorithms
Fall 2025 — Ke Chen and Yana Safonova

Worksheet 10

Monday, Nov 10, 2025

1. **Optimal Substructure.** Consider the Longest Increasing Subsequence (LIS) problem for some sequence $A = (a_1, a_2, \ldots, a_n)$. Suppose that $(A(i_1), A(i_2), \ldots, A(i_k))$ is an LIS of of $A[1 \ldots i_k]$. Prove that $(A(i_1), A(i_2), \ldots, A(i_j))$ must be an LIS of $A[1 \ldots i_j]$ for any $j = 1, 2, \ldots, k-1$.

2. **Pebbling a Checkerboard.** You are given a checkboard with 4 rows, n columns, and an integer written in each square. You are also given a set of $2n$ pebbles and want to place some or all of these on the checkboard (each pebble can be placed on exactly one square) to maximize the sum of the integers that are covered by the pebbles. There is one constraint: for a placement of pebbles to be legal, no two pebbles can be on horizontally or vertically adjacent squares (diagonal adjacency is fine).

   (a) Determine the number of legal patterns that can occur in a single column and describe these patterns.

   (b) Call two patterns compatible if they can be placed on adjacent columns to form a legal placement. Let us consider subproblems consisting of the first k columns $1 \le k \le n$. Each subproblem can be assigned a type which is the pattern occurring in the last column. Give an $O(n)$ time dynamic programming algorithm for computing an optimal placement. Your solution should use the ideas of compatibility and type to limit which patterns are considered when adding a column to an existing subproblem solution.

3. **Change Making.** You are given an unlimited supply of coins of denominations $v_1, \ldots, v_n \in \mathbb{N}$ and a value $W \in \mathbb{N}$. Your goal is to make change for W using the minimum number of coins, that is, find the smallest set of coins whose total value is W.

   (a) Design a dynamic programming algorithm for solving the Change Making problem. What is its running time?

   (b) You now have the additional constraint that there is only one coin per denomination. Does your previous algorithm still work? If not, design a new one.

4. **Rod Cutting.** Given a rod of length n inches and an array of prices that includes prices of all pieces of integer length less than n. Give a dynamic programming algorithm to determine the maximum value obtainable by cutting up the rod and selling the pieces.