Homework 6

CS6033 Design and Analysis of Algorithms I
Fall 2024
(Sec. B, Prof. Yi-Jen Chiang)

Due: Wed. 12/4 by 1pm
(submit online on NYU Brightspace; one submission per group)
Maximum Score: 105 points

Note: This assignment has 2 pages.

1. (25 points)
Consider the following variation of the rod-cutting problem in the textbook: for each possible integer length $i > 0$ (whose price is $p_i > 0$), we can have **at most one piece** of length i in the resulting cutting, i.e., for each i, the length i can be used **at most once** in the rod cutting. (Other parts are the same: there is a rod of length n where $n > 0$ is an integer, and we want to cut it into pieces of integer lengths such that the total price is maximized.) Note that in the original problem, each length i could be used for an unrestricted number of times, but now no repetition is allowed. Design and analyze a dynamic programming algorithm to solve this new variation of the rod-cutting problem in $O(n^2)$ worst-case time.
(**Hint:** For each i, view the length-i piece as an individual item, and enhance the cost function $r(\cdot)$ so that you can express it to encode the extra information about such individual items.)

2. (25 points)
A subsequence is **palindromic** if it is the same whether read left to right or right to left. For example, the sequence

$$A, C, G, T, G, T, C, A, T, C, G$$

has a palindromic subsequence A, C, G, C, A (on the other hand, A, C, T is *not* palindromic). Given an input sequence $X = x_1 x_2 \cdots x_n$ of length n, your task is to find a longest palindromic subsequence of the input X. Design and analyze a dynamic programming algorithm to carry out this task in $O(n^2)$ worst-case time.

3. (25 points)
Given an undirected graph $G = (V, E)$, a **vertex cover** of G is a subset $S \subseteq V$ of vertices such that for any edge $e \in E$, at least one endpoint of e is in S. Suppose now G is a **tree** (**not** necessarily binary) rooted at some vertex r, and each vertex v has a weight $w(v) > 0$. For any vertex set $S \subseteq V$, the weight of S, $W(S)$, is defined as $W(S) = \sum_{v \in S} w(v)$. Design and analyze a dynamic programming algorithm to find a vertex cover S of tree G with the **minimum weight**, i.e., whose weight $W(S)$ is minimized. Your algorithm should run in $O(V)$ worst-case time.

4. (30 points)
You are given a rectangular piece of cloth with dimensions $X \times Y$, where X and Y are positive

integers. For each product $i \in \{1, 2, \cdots, n\}$, a rectangle of cloth of dimension $a_i \times b_i$ is needed and the final selling price of the product is c_i, where a_i and b_i are positive integers and $c_i > 0$. You have a machine that can **cut** any rectangular piece of cloth **into two pieces either horizontally or vertically.**

Design and analyze a dynamic programming algorithm to find the best return on the $X \times Y$ piece of cloth, i.e., a strategy of cutting the cloth so that the products made from the resulting pieces give the maximum sum of selling prices. For each product i, you are free to make as many copies as you wish, or none if desired. Your algorithm should run in $O(XYn)$ worst-case time.

(**Hint:** Let $P(x, y)$ be the maximum total selling price for a rectangular piece of cloth of dimension $x \times y$. Derive a recursive solution for $P(x, y)$. We may find the following notation useful in expressing the recursive solution for $P(x, y)$. Define $[\cdot]$ as below: [statement S] $= 1$ if statement S is true, and [statement S] $= 0$ else.)