Writing Assignment 4 CMPSC 465 Fall 2020 (Instructor: Mingfu Shao) Due: Nov. 1, 11:59pm

INSTRUCTIONS:

1. Include the name and PSU access ID of every member in your group in your solution.
2. Submit your solution to Gradescope. Make sure only one of your group members submits. After submitting, make sure to add your group members.
3. Your always need to explain the running time of your algorithm.

Problem 1 (10 points).

Given array A with n distinct integers, and k, $1 \leq k \leq n$, design an algorithm to find the longest increasing subsequence of A that includes $A[k]$. Your algorithm should run in $O(n^2)$ time.

Problem 2 (10 points).

Let $A = a_1 a_2 \cdots a_n$ and $B = b_1 b_2 \cdots b_m$ be two strings. Design a dynamic programming algorithm to compute the *longest common subsequence* between A and B, i.e., to find the *largest* k and indices $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ and $1 \leq j_1 < j_2 < \cdots < j_k \leq m$ such that $A[i_1] = B[j_1], A[i_2] = B[j_2], \cdots, A[i_k] = B[j_k]$. Your algorithm should run in $O(mn)$ time.

Problem 3 (10 points).

You are taking an online quiz where the problems are numbered $i, 1 \leq i \leq n$, and have to be solved in the given order. Each problem i can be solved to obtain x_i points. Some problems can take a lot of time to solve; in fact, after solving problem i you won't be able to solve the next p_i problems. Given x_i and p_i for all problems $1 \leq i \leq n$, design an algorithm to choose a set of problems that maximizes the total points. Your algorithm should run in $O(n)$ time. (*Hint: think examining the problems in a reverse order.*)

Problem 4 (10 points).

Kim is going to order a burger. She was given a sequence of n ingredients, and she was only allowed to add a continuous subset of these ingredients, and each ingredient can be added at most once. Kim assigns a value v_i to the i-th ingredient to measure how she likes it (v_i could be negative, and a negative value means she does not like this ingredient). Design an algorithm to help Kim to order a burger she likes most (i.e., the sum of the values of the ingredients she would pick is maximized). Your solution should output the maximum sum of chosen value.

For example, the ingredients = {Lettuce, Tomatoes, Grilled Onions, Grilled Mushrooms, Green Peppers}, and corresponding values are $\{-8, 5, -1, 3, -9\}$, then the answer is 7, i.e., with {Tomatoes, Grilled Onions, Grilled Mushrooms} being added.

Problem 5 (10 points).

In a Penn State Nittany Lions football match, we have organized huge trumpets to support our team playing on the field. We want to keep the level of noise constant throughout the match. The trumpets are operated by compressed gas. However, if you blow the trumpet for 2 seconds without stopping it will break. So when the trumpet makes noise, everything is okay, but in a pause of the trumpet,we need to chant "Penn State!".

Before the match, we decide on a chanting pattern. The pattern is a sequence of 0 and 1 which is interpreted in the following way: If the pattern shows a 1, the trumpet is blown. If it shows a 0, we chant "Penn State!". To ensure that the trumpet will not break, the pattern is not allowed to have two consecutive "1"s in it.

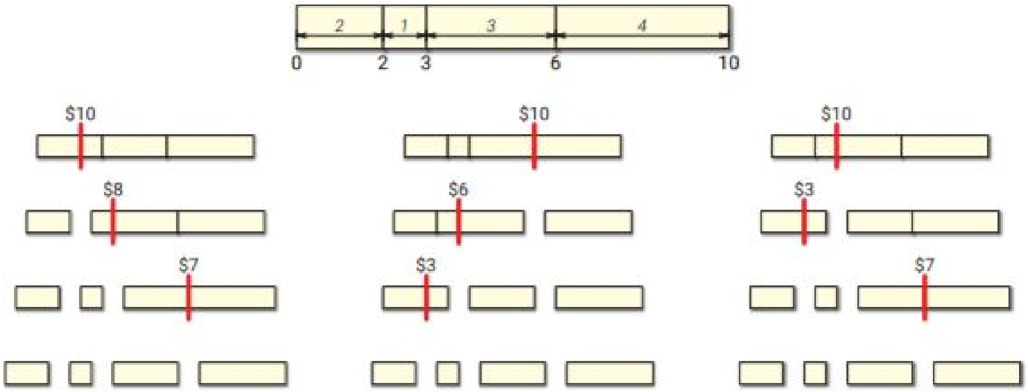
Given a positive integer n, determine the number of different chanting patterns of length n, i.e., calculate the number of n-bit sequences that contain no adjacent “1”s. For example, for $n = 3$ the answer is 5: sequences 000, 001, 010, 100, 101 are valid, while 011, 110, 111 are not. Your algorithm should run in $O(n)$ time.

Bonus Problem (10 points).

Suppose you have a long wood plank and a few markers on it indicating where you need to cut the plank. The price of each cut depends on the length of the plank. For example, cut a 10 meter long plank is \$10. However, cut a 5 meter long plank is only \$5 and cut a 1.5 meter long plank is only \$1.5.

Here is an example where the plank is 10 meters long and markers are on position 2, 3 and 6:

1. Making the cuts in order from left to right costs \$10 + \$8 + \$7 = \$25
2. Making the cuts in order from right to left costs \$10 + \$6 + \$3 = \$19
3. Making the center cut first \$10 + \$3 + \$7 = \$20



Describe and analyze an efficient dynamic programming algorithm that returns the minimum cost to make all marked cuts. The input to your algorithm is a number k, indicating the length of the wood plank, and a sorted array of positive numbers indicating the position of the markers.