CSE 551 HOMEWORK 4
COMPLETE BY 10/20/21

(1) Define $T(0) = T(1) = 1$ and

$$T(n) = \sum_{i=1}^{n-1} T(i) \cdot T(i-1)$$

  (a) What is the runtime of the recurrence relation above if intermediate results are not stored?
  (b) If the value of $T(i)$ is memoized, what happens to the runtime?
  (c) Can you devise an evaluation technique with linear runtime?

(2) A fast food joint sells chicken nuggets in packs of varying sizes. We want to know, given packs with sizes $[n_1, n_2, \ldots n_k]$, if we can order N chicken nuggets.
  (a) Devise a brute force solution and a dynamic programming solution.
  (b) Derive and compare the complexities of the brute force solution and the dynamic programming (DP) solution. What aspect of DP changes the complexity?

(3) A palindrome is a string that is the same whether read left-to-right or right-to-left. Devise a dynamic programming algorithm to find the *minimum* number of characters to delete from an input string to make it a palindrome.

For example, the word "apple" has a minimum number of characters to be removed to make it a palindrome is 3 (leaving "pp").

(4) There are N stairs and Alice can climb one or two stairs at a time. How many different ways can Alice climb the stairs such that she starts at the bottom and ends at the top? The algorithm does not need to explicitly list the ways. First devise a top-down (recursive) approach to solve this problem. What is its runtime complexity if intermediate results are not stored and not reused? How does memoizing intermediate results change the complexity? Next devise a bottom-up DP approach to solve this problem. What is its runtime complexity?

(5) At a fair, there is a game with n balloons that are placed adjacent to each other in a line. In this game, you must shoot all the balloons. Each balloon $i \in 1, \ldots, n$ has an award $v(i)$. If you shoot ith balloon you will get an award of $v(i-1)v(i)v(i+1)$ unless the left or right balloon is missing (which is the case for the balloons on the ends of the line). In the case that left (or right) neighbor is missing, let $v(i-1) = 0$ (or $v(i+1) = 0$). When balloon i is shot, the balloons to the right of balloon i decrease their position by 1 and their awards remain the same. You have to shoot all balloons in some order so that you score a maximum total number of points.
  (a) Does the strategy "For the remaining balloons at each step, shoot the balloon with highest value" work?
  (b) Does the strategy "For the remaining balloons at each step, shoot the balloon with highest award at that point in time" work?
  (c) Devise a dynamic programming method to determine an optimal order in which to shoot the balloons. Make the algorithm as efficient as possible.

Date: October 2021.