CSC373 – Problem Set 1

Remember to write your **full name** and **student number** prominently on your submission. To avoid suspicions of plagiarism: at the beginning of your submission, **clearly state any resources (people, print, electronic) outside of your group, the course notes, and the course staff, that you consulted**.

Remember that you are required to submit your problem sets as both LaTeX `.tex` source files and `.pdf` files. There is a 10% penalty on the assignment for failing to submit both the `.tex` and `.pdf`.

Due Sep 26, 2020, 22:00; required files: ps1.pdf, ps1.tex

Answer each question completely, always justifying your claims and reasoning. Your solution will be graded not only on correctness, but also on clarity. Answers that are technically correct that are hard to understand will not receive full marks. Mark values for each question are contained in the [square brackets].

You may work in groups of up to THREE to complete these questions.

Please see the course information sheet for the late submission policy.

1. [**15 points**]

   The pandemic unleashed due to the SARS-CoV-2 virus (also referred to as Covid-19) has resulted in mandatory social distancing requirements.

   As the University plans to reopen its classrooms to be used for in-person lectures again, it needs to ensure that the physical distancing guidelines are followed. As a result, the university needs to reconsider the layout of its classrooms.

   Assume that each student seat is exactly 1 feet wide. As per the distancing requirement, every two seated students must have at least 6 feet of distance between them. You are intrigued, and want to count the resulting number of possible student seating arrangements.

   For some $n \geq 0$, consider a row with exactly $n+1$ seats. You are given an array of n non-negative integers `A[1...n]`, where `A[i]` gives the distance in feet between seat i and seat $i+1$.

   Your goal is to determine the number of possible valid student seating arrangements.

   e.g. Given `A = [2, 3]`, specifying the distances between 3 seats, there are a total of 5 valid seating arrangements:

   - 1 arrangement with no students `␣␣␣`,
   - 3 possible arrangements with 1 student `X␣␣`, `␣X␣`, `␣␣X`,
   - and one arrangement with 2 students `X␣X`.

   Note that the last arrangement is valid since the empty seat in the middle is 1 feet wide, giving that the distance between the two students is exactly 6 feet.

   Systematically design a Dynamic Programming based algorithm for the above problem with worst case running-time complexity of $O(n)$ by answering the following questions:

   (a) [**1 point**] Clearly and precisely specify in English the problem you wish to solve recursively.

   (b) [**3 point**] Give a clear and precise recursive formula / algorithm for solving the problem. Identify and specify the base cases.

   (c) [**1 point**] Identify all the subproblems that your recursive algorithm needs to solve.

   (d) [**1 point**] What is the memoization data structure you will use?

   (e) [**2 point**] Find a good bottom-up evaluation order of the memoization data-structure (non-recursive) such that before solving a subproblem instance, all necessary subproblems have been solved.

   (f) [**5 point**] Write down the final dynamic-programming algorithm (non-recursive).

   (g) [**2 point**] Analyze its time and space complexity.