COMP 3711 – Design and Analysis of Algorithms
2022 Spring Semester – Written Assignment # 4
Distributed: April 3, 2022
Due: April 17, 2022, 11:59 PM

Your solution should contain

(i) your name, (ii) your student ID #, and (iii) your email address at the top of its first page.

Some Notes:

- Please write clearly and briefly. In particular, your solutions should be written or printed on clean white paper with no watermarks, i.e., student society paper is not allowed.
- Please follow the guidelines on doing your own work and avoiding plagiarism as described in the class policies. **You must acknowledge individuals who assisted you, or sources where you found solutions.** Failure to do so will be considered plagiarism.
- Please make a copy of your assignment before submitting it. If we can¡¯t find your submission, we will ask you to resubmit the copy.
- Submit a SOFTCOPY of your assignment to Canvas by the deadline. The softcopy should be one PDF file (no word or jpegs permitted, nor multiple files). If your submission is a scan of a handwritten solution, make sure that it is of high enough resolution to be easily read. At least 300dpi and possible denser.

Problem 1 [25 pts] **Number of Contiguous Sub-arrays with sum k**

Give an $O(n)$ algorithm for finding the number of contiguous sub-arrays of an array $A[1...n]$ with sum equal to k. Note that a contiguous sub-array includes all the elements between two indices i and j. For example, if $A = [2, 4, 6, 8, 10]$ the sub-array $[4, 6, 8]$ is contiguous, while $[4, 8]$ is not.

For example, let $A = [1, 1, 1, 2, 1]$ and $k = 3$, then there are 3 contiguous sub-arrays that got a sum equal to 3, e.g. $[1, 1, 1]$, $[1, 2]$, and $[2, 1]$.

Problem 2 [25 pts] **Distinct subsequences**

Given two strings $A[1...n]$ and $B[1...m]$, such that $n >= m$, propose an algorithm to find the number of distinct subsequences that can be formed from A such that they are identical to B.

For example, let A = caaat and B = cat, then the answer is 3, since the subsequences ca__t, c_a_t, and c__at are identical to B.

Problem 3 [25 pts] **Unique Paths**

You are given a binary matrix of size $m \times n$, where 1s indicate objects and 0s indicate empty cells. The matrix is like a maze, meaning that we can move to empty cells (0s) but cannot move on the objects (1s). Assuming that the starting position is at $(1, 1)$ top-left corner, the target position is at (m, n) bottom-right corner and you can move only right $(x, y + 1)$ or down $(x + 1, y)$, propose an algorithm that finds all the unique paths from the start to the end cell.

For example, given the matrix $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ the number of unique paths are 2. Either go right twice ($\rightarrow$) and then down twice ($\downarrow$) or go down twice ($\downarrow$) and then right twice ($\rightarrow$).

Problem 4 [25 pts] **Change Permutations**

Given an integer W and a set of coins C, propose an algorithm to find the number of ways the coins can sum to W assuming you have an infinite number of each one of the coins.

For example, let $C = [1, 2, 4]$ and $W = 4$, the result is 4, because there are 4 different ways such the sum of the coins is equal to $W = 4$: $\{1, 1, 1, 1\}$, $\{1, 1, 2\}$, $\{2, 2\}$, and $\{4\}$.