American University of Armenia, CSE
CS121 Data Structures B
Spring 2022

Homework Assignment 1

Due Date: Saturday, February 5 by 23:59 electronically on Moodle

Please solve the programming tasks using Java, following good coding practices (details are on Moodle). Don't forget to submit the corresponding assignment report.

1. (8 points) For the code below:

```
1  /** Returns the number of times second array stores
2  sum of sufix sums from first. */
3  public static int example(int[] first, int[] second) {
4      // assume equal-length arrays
5      int n = first.length, count = 0, total = 0;
6      for (int j=n-1; j >= 0; j--)              // loop from n-1 to 0
7          for (int k=n-1; k >= j ; k--)         // loop from n-1 to j
8              total += first[k];
9      for (int i=0; i < n; i++)                 // loop from 0 to n-1
10         if (second[i] == total) count++;
11     return count;
12 }
```

(a) Give and justify Big-O characterization, in terms of n.

(b) Implement an alternative method *fast_example* that performs the same task efficiently. What is the Big-O complexity?

2. (8 points) Consider a programming language that has exactly one variable, y, and two operations $++$ and $--$ that increase and decrease the value of y by 1 respectively. Each statement in this language includes exactly one operation and the variable y. The operation sign can be located at either side of the variable. Write a program, that inputs natural number n and n lines, each containing one statement, and calculates the final value of y after applying all the operations. The initial value of y is 0. Your program should use minimum number of primitive operations and use as little space as possible. You are not allowed to perform character comparison.

sample input	sample output
5 ++y y-- y++ --y --y	-1

3. (12 points) Let M be an $n \times n$ integer matrix, consisting of $n \times n - 1$ ones and a single zero. In one step, you can swap any two adjacent columns or rows of M. Write a program that inputs natural odd number n, and n lines, each containing n integers separated by a single space (representing M) and calculates the minimum number of steps needed to relocate the only zero to the center of M. The program should use minimum number of primitive operations. M should consists of $n \times n - 1$ 1s and a single 0.
What is a tight bound on the running time of this method? What is the worst case complexity? When do we encounter the worst case?

sample input	sample output
5 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	3

4. (19 points) Write an efficient **recursive** program that, given a natural number n, determines and outputs all the possible strings of length n, s.t. they can contain only the letters `e, f, g, h, i`, and three consecutive letters cannot be all vowels or all consonants.

5. (24 points) Write a **recursive** program that inputs a natural number n and prints a crossed rhombus of asterisks (the symbol `*`) with height n. Note that your program is not allowed to use arrays, strings or loops; it should directly generate the output in a recursive manner. You are allowed to have at most one additional recursive helper method.

For example, when $n = 6$, your program should print:

```
     *
    ***
   * * *
  *  *  *
 *   *   *
***********
 *   *   *
  *  *  *
   * * *
    ***
     *
```

6. (24 points) Consider a rectangular maze of size $n \times m$ and a player trying to find the exit in it. An available position is denoted with `` `.' ``, a wall with `` `X' ``, the exit position with `` `E' ``. The player can move up, down, left or right one position at every step but cannot leave the boundaries of the maze.

Write a **recursive** program that determines the minimal number of steps needed to exit the maze from the given initial position of the player. If the exit cannot be reached, the program should output the largest value of the Integer type. The input consists of the maze dimensions n and m and the coordinates of the initial position of the player (y and x coordinates counted from the top-right corner), followed by n lines of m characters each, describing the maze.

sample input	sample output
4 5 0 3 .X..X ..X.. .X..X X...E	4