Data Structures HW1

Logistics

Due date: 3/28 (Sun) 23:59

Submission

- Via LMS (no email submission)
- **Three files** (please make sure you upload three files)

1. Report (no specific template, MS word or 한글, in English or in Korean)
2. output.txt (answer file)
3. Zip file of your program (**Compress it** to make a single file.)

Note: No restrictions on programming languages and platforms.

Report

Describe your idea for solving the problem, explain the code in detail, and draw your conclusion. Your report must include the following:

1. Program description (comments on important code lines)
2. How to run (so that I can test your program)

Evaluation Policy (10 pts in total)

Score (10pts) = Report (4pts) + Accuracy (6pts: 2pts for Task1, 4pts for Task2)

Accuracy: Full Mark x # right answers / # total cases

Penalties

1. Unable to build code → Accuracy = 0
2. Not using recursion → Score /= 2;
3. **Plagiarism → Score = -5 (will affect your overall grade)**
4. Late Submission → Report = 2
5. Wrong output format and missing files (in case you forget to submit output.txt, or...) → Report = 3

Problem Description: Maze Runner

Using a recursive method, write a program to check whether there exists a path for a given maze. Mazes are provided in the “input.txt” file which is written in the following format:

```
2
3 3
1 0 0
1 1 0
1 0 1
4 3
1 0 1
1 1 1
0 1 1
1 0 1
```

First, the number on the first line (i.e., 2) represents the number of mazes to be processed. The two numbers in the second line (i.e., 3 3) show the numbers of rows and columns for the first maze, respectively. So, the first maze size is 3 x 3, which is given in the next three lines. The same goes for the second maze.

The mazes are written with a binary value, 0 or 1, and you can only move to a point of 1. For all mazes, the starting point is set to the top left point and the end point to the bottom right point. It is guaranteed that these two points are 1. There are two conditions on your moves.

First, **you cannot move left, that is, you can only move right or up or down**.

Second, **you cannot go to a point that you have already visited. Once you have visited a point, then you cannot revisit it.**

There are two tasks for this homework.

Task1 (mandatory): Find a path for each maze, and write the result in “output.txt”.

Since the path can only be found in the second case in the example above, your program should write the binary answer for each case in “ouput.txt” as follows:

```
0

1
```

Task2 (bonus): For the same data, find the number of unique paths for each case, and write the result in “output.txt”. Limit the maximum number of paths for each case to 1,000,000: print 1000000 if the number of found paths exceeds 1,000,000. In the example above, you can find two paths for the second maze, so "output.txt" will be as follows:

```
0

2
```

As you know, the answer of Task2 can cover that of Task1. **So, if you solve Task2, then you do not need to write code for Task1.**

*****Important Note*****

When writing the output file, **write only one answer per line as above**. Since a program will be used to evaluate, the score result may be weird if the format is not correct. Below are some output files in a wrong format:

Case 1. Unnecessary information

```
12345678 Alice Dept. of Mobile Systems Engineering

#1 0

#2 2
```

Case 2. Multiple answers in a line

```
0 2
```

Case 3. Undefined

```
0

2 0
```