Assignment 2 – Stack & Queue

TA: Max (max.t7401@gmail.com)

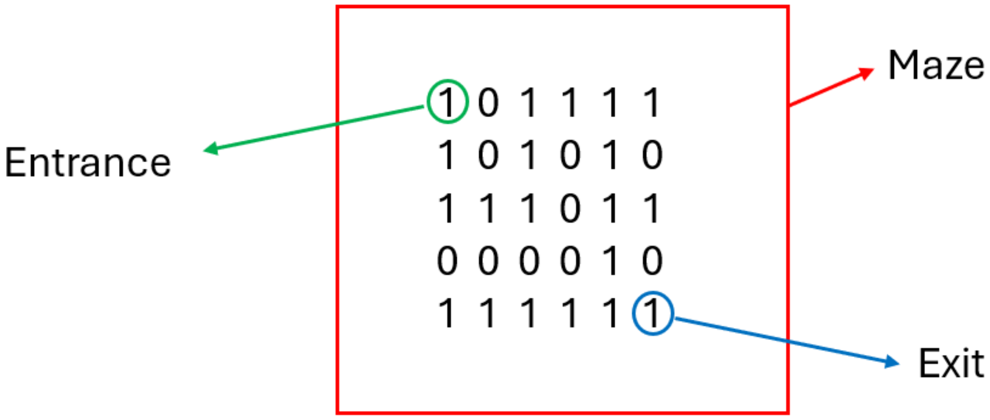
Deadline:11:59 p.m., October 23, 2023

1. Maze Game

This is a maze game in which you must implement a queue to find the shortest path out of the maze, using a breadth-frist search. The maze is represented as a matrix composed of 0s and 1s. As shown in the Figure 1 below, the entrance of the maze is fixed at the first element of the matrix, and the exit is fixed at the last element. During movement, an element with a value of 1 represents a walkable path, while an element with a value of 0 is considered an impassable wall.

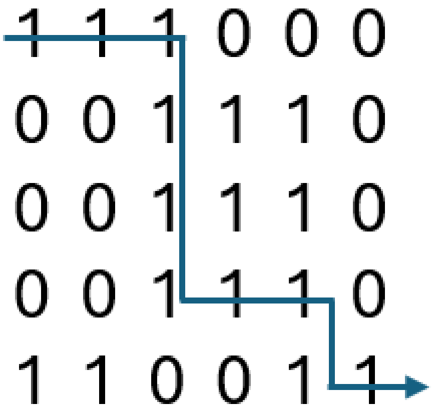
Unlike the DFS (Depth-First Search) approach introduced in the class, BFS (Breadth-First Search) on a maze explores the maze level by level. The visiting order is to take some vertex as the starting point, then visit all vertices adjacent to the start, and then visit the vertices adjacent to those vertices. That is, gradually moving outward layer by layer. This continues until all connected vertices have been visited.

Hint: When multiple shortest paths exist (shown as Figure 2), your BFS search must follow the fixed direction priority in the order of Up → Down → Left → Right.



```
1 0 1 1 1 1
1 0 1 0 1 0
1 1 1 0 1 1
0 0 0 0 1 0
1 1 1 1 1 1
```

Figure 1: A maze example



```
1 1 1 0 0 0
0 0 1 1 1 0
0 0 1 1 1 0
0 0 1 1 1 0
1 1 0 0 1 1
```

Figure 2: Shortest path in search order Up → Down → Left → Right.

Input:

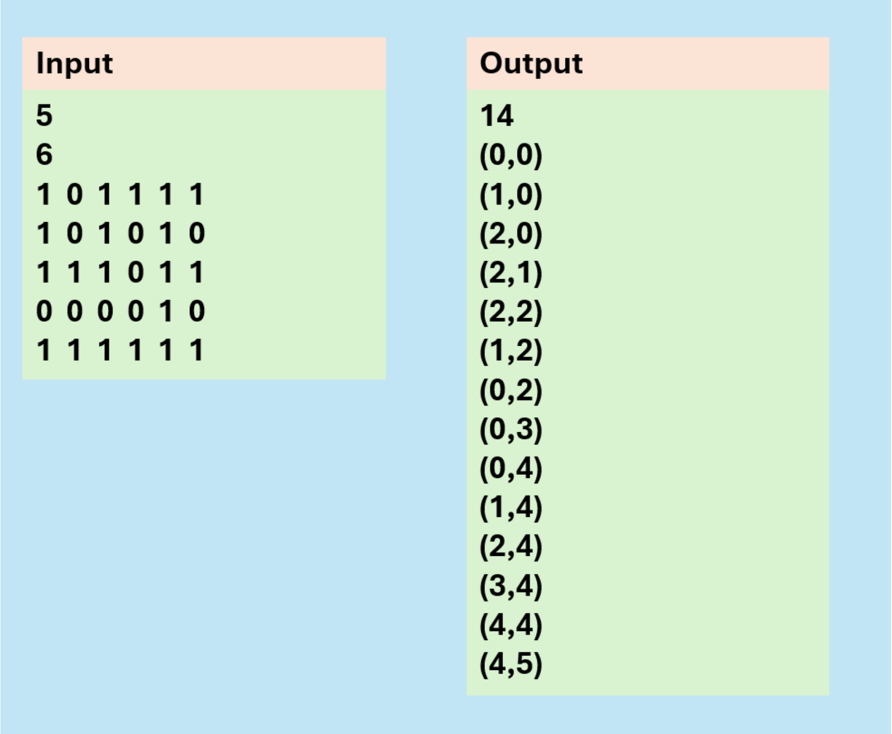
- In the test data, the first line of input represents the number of rows in the maze, and the second line represents the number of columns.
- The following lines (equal to the number of rows) describe the layout of the maze matrix. It consists of the following characters:
  - “0” represents the wall.
  - “1” represents the path.

Output:

- If the maze has a path from the entrance to the exit, you must find the shortest path among all possible paths and output the result using the format as shown in Figure 2.
  - The first line shows the number of steps in the shortest path.
  - The second line shows the fixed coordinates of the entrance (0,0).
  - The last line shows the fixed coordinates of the exit. That is,the exit of the cordiates is (number of rows − 1,number of columns − 1).
  - From the third line up to the line before the last, list all the coordinates in the path.
- If no path exists to leave the maze, output "Cannot reach the exit".

Time Limit: 1 second

Sample test case:



Input

```
5
6
1 0 1 1 1 1
1 0 1 0 1 0
1 1 1 0 1 1
0 0 0 0 1 0
1 1 1 1 1 1
```

Output

```
14
(0,0)
(1,0)
(2,0)
(2,1)
(2,2)
(1,2)
(0,2)
(0,3)
(0,4)
(1,4)
(2,4)
(3,4)
(4,4)
(4,5)
```

Figure 2: A sample test case with its input and output

2. Hanoi Tower

The Tower of Hanoi is a classic puzzle consisting of three rods and a number of disks of different sizes, which can slide onto any rod. The puzzle starts with all the disks stacked in ascending order of size on one rod, with the largest disk at the bottom and the smallest at the top. The objective is to move the entire stack to another rod, following these rules below:

1. Only one disk can be moved at a time.
2. Each move consists of taking the top disk from one rod and placing it on top of another rod.
3. No disk may be placed on top of a smaller disk.

You must implement a stack for each rod to solve the problem. As shown in Figure 3 below, all the disks are initially stacked on rod A. The disks are numbered sequentially from 1 to ***d***, where a larger number indicates a larger disk.You must move all the disks to rod C using the minimum number of moves, which is $2^n - 1$. In the input data, you will receive a number ***k***, which indicates that you must output the first k moves and the state of the rods after completing these k moves.

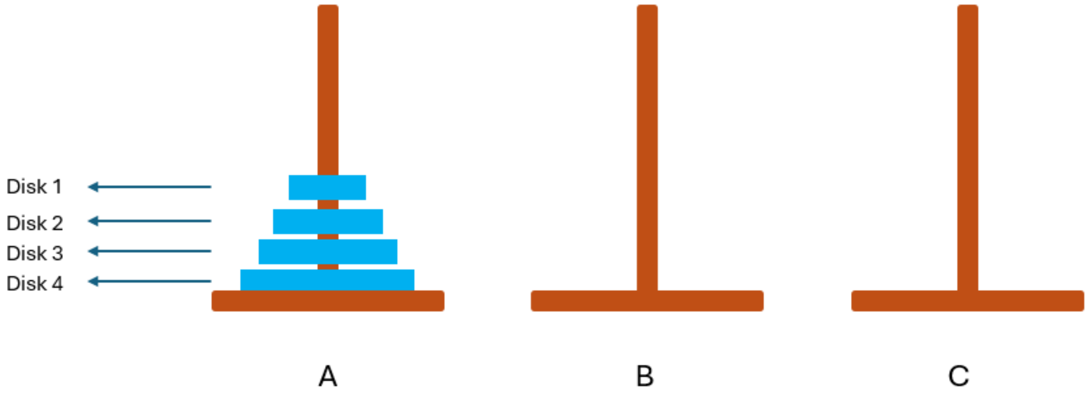


Figure 3: 2. Hanoi Towers

Input:

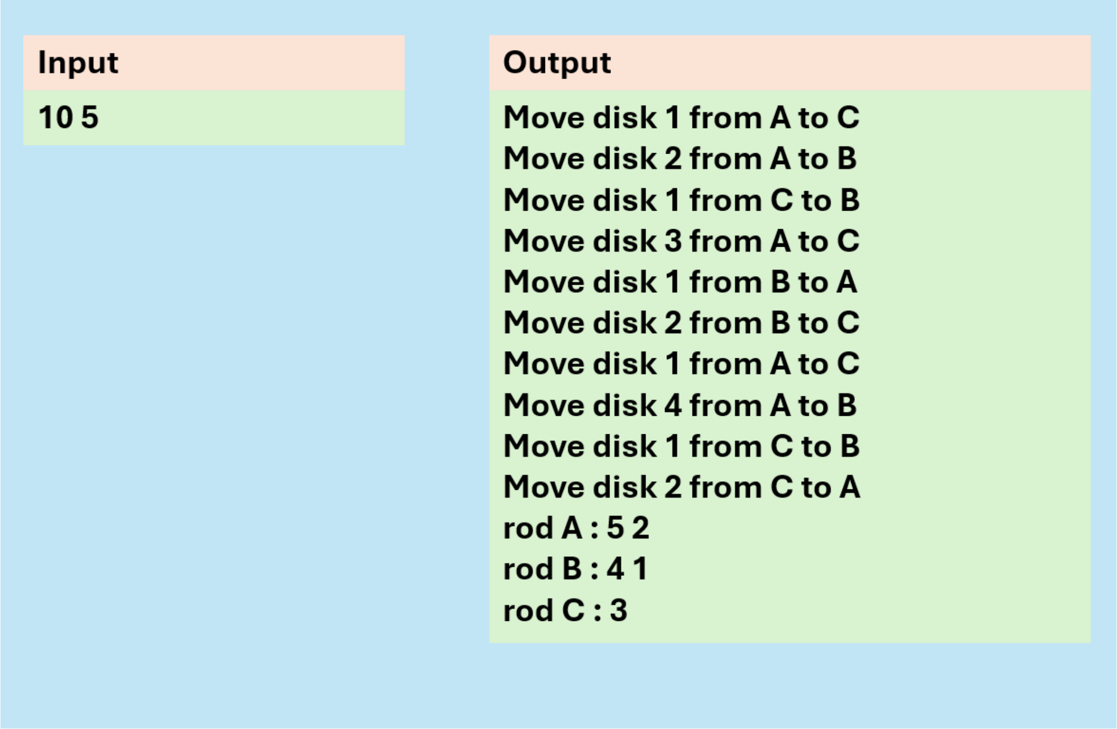
- The first line contains two integers, first number is ***k*** (1$\leq k \leq 2^d - 1$), second number is ***d*** (1$\leq d \leq 100$).

Output: (Figure 4)

- For the first *k* lines, you must output each move in the format: "**Move disk (disk number) from (rod ID) to (rod ID)**".
- For the last three lines, you must output the state of the three rods after performing the k moves, in order. The output format is as follows:
  "rod (rod ID) : (disk number) (disk number) ….".
  The disk numbers should be output in descending order, from largest to smallest.

Time Limit: 1 second

Sample test case:



Input

```
10 5
```

Output

```
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
rod A : 5 2
rod B : 4 1
rod C : 3
```

Figure 4: A sample test case with an input and the output

3. Readme, comments, and coding style

An indicator of good source code is readability. To keep source code maintainable and readable, you should add comments to your source code where reasonable. A consistent coding style also helps a lot when tracing the source code. For this assignment, please also compose a readme file in *.txt format and name it “README.txt”. This file should contain a brief explanation of how to use your program. Please remember to have your source code comments and readme file in English.

4. Submission

To submit your files electronically, log in to the DomJudge website through the following URL: http://domjudge.csie.io:12345

Press the submit button and choose the homework questions you want to submit. After submitting your code, DomJudge will give you a result to tell you whether your code is correct or not. Please note that your code will be evaluated by different sets of test cases. Please make sure your code can work correctly based on the description above. Additionally, you must compress your code and the README file into a zip file and upload it to **Ecourse2**. Otherwise, you will get zero points.

ATTENTION: Do NOT copy others’ work or you will get a zero.

5. Grade Policies

The TA(s) will mark and give points according to the following grading policies:
50 % Problem1(Maze Game)

45 % Problem2(Hanoi tower)

5% Readme, comments, and coding style.

The readme file should include your name, class ID, a brief description of the code, and other issues students think will be helpful for the TAs to understand their homework.