COMP2119 Introduction to data structures and algorithms
Assignment 3 (Programming)

Due: April 7, 2022 5:00pm

Part 1 (Graph)

(Only Part 1 is mandatory and graded.)

Q 1. Consider a "*matrix map*" of cells arranged as an $m \times n$ matrix. Each cell in the map is referred to by a "*location*". Specifically, the location of the upper left corner cell of the map is (0,0), and the location of the lower right corner cell of the map is $(m-1, n-1)$. Each cell in the map has one of two values 0 and 1. A cell with value 0 means that it is an empty cell where a robot is allowed to move into. A cell with value 1 means that it is a wall where a robot can not move into. A robot is to navigate through the map starting at location $(0, 0)$ until it reaches location $(m-1, n-1)$. Robot movements are restricted by two rules: (1) The robot can move only within the map. (2) The robot can move up, down, left or right by one step each time to a neighbouring cell. Also, we cells $(0, 0)$ and $(m-1, n-1)$ are empty. Write programs to solve the following problems:

(a) [40%] Write a program that, given a matrix map, returns **the least number of steps** that the robot need to take to travel from cell $(0, 0)$ to cell $(m-1, n-1)$. Your program should return "-1" if there are no valid routes. Here are some examples:

- Example 1:

0

Input: grid=[[0]]
Output: 0

- Example 2:

0	1
1	0

Input: grid=[[0, 1], [1, 0]]
Output: -1

- Example 3:

0	0	1
0	1	0
0	0	0

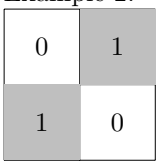
Input: grid=[[0, 0, 1], [0, 1, 0], [0, 0, 0]]
Output: 4

(b) [30%] Assuming that the robot has the ability to **eliminate 1 wall** (i.e., turning one cell from "1" to "0"), write a program that returns the least number of steps for the robot to travel from cell $(0, 0)$ to cell $(m-1, n-1)$. Some examples:

- Example 1:

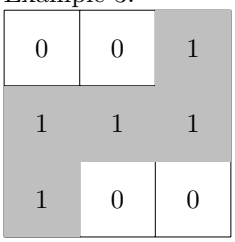
Input: grid=[[0]]
Output: 1

- Example 2:



Input: grid=[[0, 1], [1, 0]]
Output: 2

- Example 3:



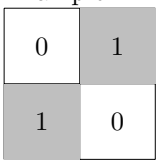
Input: grid=[[0, 0, 1], [1, 1, 1], [1, 0, 0]]
Output: 4

(c) [30%] Now, assume that the robot can eliminate k walls, where $k \geq 1$. Repeat part (b). Some examples:

- Example 1:

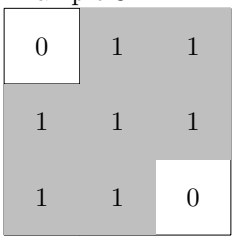
Input: grid=[[0]], $k = 1$
Output: 0

- Example 2:



Input: grid=[[0, 1], [1, 0]], $k = 1$
Output: 2

- Example 3:



Input: grid=[[0, 1, 1], [1, 1, 0], [1, 1, 0]], $k = 2$
Output: -1

Note:

1. You should complete the assignment in Python 3. You are allowed to use 'collections' package, 'heapq' package, 'set' and 'list' directly in your programs.

2. You should complete certain files provided in the **A3** folder (see below). You can only modify the functions **leastStepQ1a**, **leastStepQ1b**, **leastStepQ1c** and **theLeastPrice**.

```
A3
├── COMP2119_Graph.pdf ..... The introduction of this assignment is written in this document
├── Part1
│   ├── A3_P1_1a.py ..... The leastStepQ1a function in this file is for Q1(a) in part 1
│   ├── A3_P1_1b.py ..... The leastStepQ1b function in this file is for Q1(b)in part 1
│   ├── A3_P1_1c.py ..... The leastStepQ1c function in this file is for Q1(c)in part 1
│   └── utils.py ..... This file should not be modified
└── Part2
    ├── A3_P2_1.py ..... The theLeastPrice function in this file is for Q1 in part 2
    └── utils.py ..... This file should not be modified
```

3. Some test cases are provided but you are encouraged to design your own test cases. The autograding result will be shown on your terminal once you run '**python A3_P1_1a.py**', '**python A3_P1_1b.py**', '**python A3_P1_1c.py**'. Your score in the assignment will be evaluated by other (unrevealed) test cases, which are not used by the autograder. You should organize your submitted files in the following way. Please replace the **UID** by your university number and make sure your programs can run normally before you zip the folder **UID** as a **UID.zip** file.

```
UID
├── Part1
│   ├── A3_P1_1a.py ..... This is your code file for Q1(b)in part 1
│   ├── A3_P1_1b.py ..... This is your code file for Q1(b)in part 1
│   ├── A3_P1_1c.py ..... This is your code file for Q1(c)in part 1
│   └── utils.py ..... This file should not be modified
└── results
    ├── UID.pdf ..... Please give a brief description of your algorithm which would help the TA
    │   understand your algorithms of Q1 in part 1
    ├── UID_P1_1a.jpg ..... This is a screenshot of your autograding result of Q1(a)
    ├── UID_P1_1b.jpg ..... This is a screenshot of your autograding result of Q1(b)
    └── UID_P1_1c.jpg ..... This is a screenshot of your autograding result of Q1(c)
```

4. In the **leastStepQ1a** function and **leastStepQ1b** function, there will be one parameter, *grid*. In the **canReachQ1c** function, there will be two parameters, *grid* and k. The structure of the parameter *grid* is List[List[int]]. The data type of k is int. The output variable of these three functions are int.