CMSC 142 (Last) Lab Exercise
Finding the Shortest Path: Binary Version
Deadline: May 23, 2025

INSTRUCTIONS:

1. **Submit in LMS a zip File containing the following:**
    a. python code files (version1.py, version2.py)
    b. short description of the algorithm
2. **SUBMISSION DETAILS**
    a. The implemented program code should be written in python
    b. A short report in pdf format. Provide a short explanation of your code/solution. Include references, if any.
    c. If AI was used, disclose the parts of your code where AI was used. List down the prompts.
    d. Please do not submit code that you copied from another group or from a source you found online (and slightly modified). Those who submit copied codes will be automatically given a score of zero, and the source of the copied codes will also be given deductions.

ACTIVITY GUIDE:

1. Read the problem specifications provided below.
2. There are two problems that you need to solve:
    a. Binary Version
    b. Weighted Version
    c. Believe me, you just have to change 2-3 lines of your code to make both of these work
3. Some test inputs are provided with the solution
    a. Binary Version
        i. Input Filename: input_bin
        ii. Solution Filename: solution_bin
    b. Weighted Version
        i. Input Filename: input_weighted
        ii. Solution Filename: solution_weighted

PROBLEM SPECIFICATIONS:

Shorty decided to walk around UPV to destress after a whole day of classes, lab defense, and exams. Since Shorty likes hiking, she trekked along the Nature Trail and found a shortcut leading to the University library. Upon arrival, she was enchanted by the smell of new books and went ahead to read some fiction.

Tired from hiking, Shorty fell asleep while reading and got transported into a new dimension. In this realm, the University library is a maze filled with evil spirits. To get back to the real

world, she must find the historical book containing the spell that would help transport her back.

However, Shorty is only given 5 minutes. The library is a grid of NxN rooms, and each room can either be:

- Open (1) safe to walk through
- Blocked (0): has evil spirits

Your mission: **Find the shortest (least-cost) path from the fiction area (where Shorty is currently situated) to the historical section where the book is located.**

Input:

- You will be reading a file
- First input is the number of N test cases / number of mazes
- This is followed by N chunks of data, containing the following in this format:
    - The size of the square grid X
    - This is followed by X number of rows, where each row is composed of 0s and 1s, separated by comma

Output:

- For each test case, output the shortest number of steps to reach the destination

Extra Output:

- Show a visualization of the maze and the shortest path taken by your algorithm

Example Input:

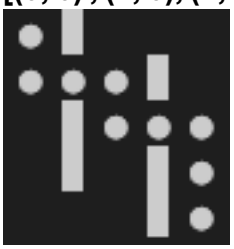
2
5
[1, 0, 1, 1, 1],
[1, 1, 1, 0, 1],
[1, 0, 1, 1, 1],
[1, 0, 1, 0, 1],
[1, 1, 1, 0, 1]
7
[1, 0, 1, 1, 1, 0, 0],
[1, 0, 1, 0, 1, 1, 1],
[1, 0, 1, 1, 1, 0, 1],
[1, 1, 1, 0, 1, 0, 1],
[1, 0, 0, 1, 1, 0, 1],
[0, 1, 0, 1, 1, 1, 1],
[0, 0, 0, 0, 0, 0, 1]

Output:

8
14

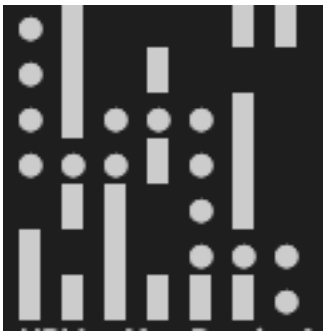
Explanation:

For the first test case, the shortest path is:
[(0, 0) , (1, 0), (1, 1), (1, 2), (2, 2), (2, 3), (2, 4), (3, 4), (4, 4)]



The cost is 8.
Note that we assume that the starting cost at the source cell is 0. Hence, we start counting the cost when we take the first step.

For the second test case, the shortest path is:
[(0, 0) , (1, 0), (2, 0), (3, 0), (3, 1), (3, 2), (2, 2), (2, 3), (2, 4), (3, 4), (4, 4), (5, 4), (5, 5), (5, 6), (6, 6)]



The cost is 14.

Finding the Shortest Path: Weighted Version

Now, suppose that the rooms have costs. The larger the cost, the stronger the evil spirit is in that room. Therefore, we must find the path with the least cost to get to the historical section, fast and safe from evil.

In this input, we have 2 test cases:
2
4
[1, 1, 1, 1],
[1, 1, 8, 1],
[4, 5, 1, 0],
[1, 1, 1, 1]
4
[1, 0, 0, 1],
[1, 5, 3, 1],
[1, 2, 1, 1],
[1, 0, 0, 1],

The output should be:
9
7

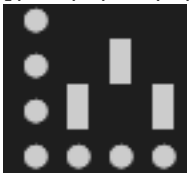
Explanation:

For the first test case, the shortest path is:
[(0, 0), (1, 0), (2, 0), (3, 0), (3, 1), (3, 2), (3, 3)]
The cost is 9 because:
(0, 0) – assumed to have 0 cost
(1, 0) - 1
(2, 0) - 1
(3, 0) - 4
(3, 1) - 1
(3, 2) - 1
(3, 3) - 1

Total: 9

For the second test case, the shortest path is:

[(0, 0), (1, 0), (2, 0), (2, 1), (2, 2), (2, 3), (3, 3)]



(0, 0) = 0
(1, 0)=1
(2, 0) = 1
(2, 1) = 2
(2, 2) = 1
(2, 3) = 1
(3, 3) = 1

Total: 7

You may use this (as a reference) for visualization. If you can find a better way to visualize, you may do so.

```
def print_maze(maze, path=None):
    path = path or []
    for r in range(len(maze)):
        row_str = ''
        for c in range(len(maze[0])):
            if (r, c) in path:
                row_str += '● '  # path
            elif maze[r][c] == 1:
                row_str += '  '  # open space
            else:
                row_str += '█ '  # wall
        print(row_str)
```