

Programming Assignment III

Shortest Path in a Maze

Due Date: 2025/10/08 (40 points)

1 Description of the Assignment

Design and implement efficient data structure and algorithm for finding shortest path from the [start](#) cell s to the [destination](#) cell t in a maze of dimension $m \times n$.

2 Input

The input data is a description of a maze M of dimension $m \times n$. Each cell of M can be one of the following types:

0 : empty.

x : obstacle.

s : start.

t : destination.

Note that there is only one s and one t .

The input data contains a description of [non-zero](#) elements of the maze. For example, let the maze be

$$\begin{pmatrix} 0 & x & 0 & 0 & 0 \\ 0 & s & 0 & x & 0 \\ 0 & 0 & x & 0 & 0 \\ 0 & x & 0 & x & t \end{pmatrix}$$

Then the input data for the above maze is:

```
4 5
2 x 0
2 s 4 x 0
3 x 0
2 x 4 x 5 t 0
```

The first line of the input is the dimension of the maze. In the above example, the dimension of the maze is 4×5 .

For each row of the maze, a list of non-empty cells is given for that row. That is, each non-empty cell is represented by c and k , where c is the cell's column number and k is the cell's type. The list ends with 0. For example, the last row (row 4)

2 x 4 x 5 t 0

shows that $M[4][2] = x$ $M[4][4] = x$ and $M[4][5] = t$. Note that rows are numbered from 1 to m , and columns are numbered from 1 to n .

Design an efficient algorithm to find a shortest path from s to t . The path can only go through empty cells.

3 Output

Print out the path from s to t . In the above maze, the path is

$(2, 2)(2, 3)(1, 3)(1, 4)(1, 5)(2, 5)(3, 5)(4, 5)$

The path can also be shown in the rectangular matrix, by using a color which is different from the others. For example,

0	x	2	3	4
0	s	1	x	5
0	0	x	0	6
0	x	0	x	t

Notes

The format of each assignment report should be close to a technical research report and must include at least the following sections:

1. **Title and Author**

On the [first part of the first page](#), clearly include:

- Assignment number
- Your name
- Student number
- Email address

2. **Description of the Problem**

Provide a clear and formal description of the problem of this assignment.

In addition to the basic requirements given in the assignment, highlight any extra functions or features you have implemented. [Do not simply copy the assignment instructions into this section.](#)

3. **Main Results**

This section should include at least the following:

- (a) [Program Design](#).
Explain the overall design of your program. If any part of the design was obtained from references, discussions with other people, or other sources, proper citations must be provided.
- (b) [Data Structures](#).
Describe the data structures you implemented to improve efficiency. These should be your own implementations and appear in the first part of your program.
- (c) [Program Listing with Comments](#).
 - i. If the program is long, include only the main parts in the report body, and place the complete program in the appendix.
 - ii. Add explanatory comments where appropriate to clarify your design.
- (d) [Program Outputs](#).
Include compilation results and execution outputs. Whenever possible, provide [screen dumps](#).

4. Performance Evaluation

- (a) Report execution times of your program with various input sizes, such as $n = 100, 200, \dots, 1000$.
- (b) Indicate the maximum input size your program can handle within a reasonable time limit (e.g., 1, 5, or 10 minutes).

5. Conclusions

- (a) Summarize what you accomplished and any interesting insights gained from this assignment.
- (b) Describe the challenges you encountered during development and how you overcame them. (This provides strong evidence that you completed the work independently.)

Additional Notes

1. Submit your report on or before the due date.
2. The program output should clearly demonstrate correctness. That is, provide a set of [comprehensive](#) (but not necessarily exhaustive) annotated test cases to show that your program works correctly.
3. Print the report on A4 paper and staple it in the [upper-left](#) corner.