Programming Assignment III
Shortest Path in a Maze
Due Date: 2024/10/09   (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 cell.

x : obstacle cell.

s : start cell.

t : goal cell.

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

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

The input 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. Each non-empty cell is represented by c and k, where c is the column number of the cell, and k is the type of the cell. The list is terminated by a 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,

$$
\begin{array}{ccccc}
0 & x & 2 & 3 & 4 \\
0 & s & 1 & x & 5 \\
0 & 0 & x & 0 & 6 \\
0 & x & 0 & x & t \\
\end{array}
$$

Notes

The format of the report of the assignment should be close to a technical research report, and include at least the following items:

1. Title and Author.
   This section should include assignment number, *your name*, *student number* and *email address* on the *first* part of the *first* page of your report.

2. Description of the Problem.
   A "formal" description of the problem in this assignment.
   In addition to the basic requirements specified in the assignment, highlight additional functions or features that you have implemented.
   Do not copy assignment instructions directly into this section.

3. Main results.
   This section should include at least the following items.

   (a) The design of your program.
       If the design or part of the design was obtained by reference to other sources, discussions, etc., appropriate citations should be given.

   (b) The data structures used in the program to improve the efficiency of the program.
       These data structures should be implemented by you and appear in the first part of your program.

    (c) List of your program with comments.

        i. If your program is very long, list only the main parts of the program here and the entire program in appendix.

        ii. Additional comments can be added manually to explain the design of the program.

    (d) Outputs of the compilation and the executions of your program. This section should use screen dumps whenever possible.

4. The performance evaluation of your program.

    (a) List the execution time of your program with various input sizes, such as $n = 100, 200, \cdots, 1000$.

    (b) The maximum input size that your program can run in a reasonable amount of time, such as 1 minute, 5 minutes, or 10 minutes.

5. Conclusions

    (a) Summarize what you did and the interesting things you learned from this assignment.

    (b) Describe the difficulties you encountered during program development and how you overcame them. (This is strong evidence that you did the work yourself.)

Additional notes:

1. Turn in your report on or before due day.

2. The output of the program execution should indicate the correctness of your program. In other words, a set of comprehensive (but not necessarily exhaustive) annotated test data for the problem should be provided to show that your program is indeed correct. This can be done by carefully selecting a set of test data.

3. Print or write the report on A4 paper and staple it to the upper left corner.