

## Two-Terminal Maze Routing Problem

### Problem Descriptions:

There are a source and a destination in a rectangular area. And you need find a path from the source to the destination. The area is divided into cells: The source is located in a certain cell, and so is the destination. You can move from a cell to one of its neighboring cells in four directions (i.e., up, down, right and left) in a single step. It consumes a cost (e.g., energy and money) to pass through a cell. Especially, some cells are forbidden to pass.

Suppose you are initially at the source. What you need to do is to find a path with a minimum cost. This problem is useful in PathFinding procedure to move a character in a video game, or to connect terminals in a VLSI circuit.

1	1	1	1	2	1	2	1
1	2	3	2	2	1	3	2
1	2	5	1	4	D	3	4
1	2	3	1	1	3	4	-1
1	2	2	1	1	1	5	-1
1	-1	-1	2	2	2	2	-1
1	-1	-1	2	2	3	4	3
S	1	4	3	1	1	3	1

1	1	1	1	2	1	2	1
1	2	3	2	2	1	3	2
1	2	5	1	4	D	3	4
1	2	3	1	1	3	4	-1
1	2	2	1	1	1	5	-1
1	-1	-1	2	2	2	2	-1
1	-1	-1	2	2	3	4	3
A	1	4	3	1	1	3	1

An example is shown on the figures above. The left figure shows cells with costs to pass through, where there are blue cells as the source (S) and the destination (D) and red cells as the blockages (and we use -1 to represent  $+\infty$ ). The right figure shows a possible path in yellow.

### Input:

A  $10000 \times 10000$  matrix, each element in matrix represent the source, the destination, a normal cell, or a blockage.

- Each normal cell has a value representing the cost (ranging from 1 to 5). A cell with value -1 is forbidden to pass through.
- S represents the source, and D represents the destination.

The matrix is given row by row, and each element will be separated by one space. An example is shown below:

```
1 1 1 1 2 1 2 1
1 2 3 2 2 1 3 2
1 2 5 1 4 D 3 4
1 2 3 1 1 3 4 -1
1 2 2 1 1 1 5 -1
1 -1 -1 2 2 2 2 -1
```

```
1 -1 -1 2 2 3 4 3
S 1 4 3 1 1 3 1
```

**Output:**

- If you cannot find a path: output -1.
- Otherwise output the minimum cost of your path. Then output the position of each cell in your path from source to drain in one line. The position of a cell should be represented like "(row, column)" and each position is separated by one space.

**Requirement:**

- Download benchmark from <https://disk.pku.edu.cn/#/link/9F3F13C117557FC6DE99F27DB08DB8D4?gns=6DACEE73D90348F983B454E5C4F5D2F9%2FC5F082E4D78746C0AB88F092EC94FCA2> to this directory and read the matrix from \*.txt file in directory “./benchmark”.
- Output all results to result.txt in directory “./”, each line should contain the case name and its output like:

```
case1 -1
case2 15 (0, 0) (0, 1) (2, 0) ... (6, 8)
case3 27
...
```