

COP 3503 Homework #4: Maze Magic

Filename: maze.java

Time Limit: 4 seconds (per input case)

Standard Input, Standard Output

Your friend Sastry is hopelessly lost in a maze, and you would like to help him get out. Luckily, his cell phone is fully charged and you can call him and give him directions to follow. He's spent so much time lost in the maze, he insists that you get him out as fast as possible.

The maze can be modeled by a two dimensional grid with r rows and c columns. Sastry's location is labeled with the character '*' and the one location that allows for escape out of the maze is labeled with the character '\$'. In most circumstances, Sastry can move to the left or right by one square on a single row, or move one square up or down in a single column. However, there are some exceptions to this rule. Some squares in the grid are forbidden. These are marked with the character '!'. Other squares are teleportation squares. Each of these are marked with a capital letter. If Sastry is on a square with a letter, say 'A', he can teleport, in one move, to all other squares labeled with the letter 'A'. Same goes for all of the other capital letters. *Note that from a teleportation square, one can always choose not to use the teleportation feature and can still move left, right, up or down by one square. Thus, one can travel from a square labeled 'A' to an adjacent square labeled 'B', or an adjacent square labeled '!'.*

The Problem:

Given the size and contents of the maze, figure out the fewest number of moves Sastry needs to make to get out of the maze.

The Input:

The first line of input contains two space separated integers, r ($2 \leq r \leq 1000$) and c ($2 \leq c \leq 1000$), representing the number of rows and number of columns in the grid, respectively.

The following r lines contain c characters each. The i^{th} line of these lines contains the contents of the i^{th} row of the grid, from left to right.

It is guaranteed that exactly one of the grid characters will be '*' and exactly one of the grid characters will be '\$'. All grid characters that represent regular squares will be labeled with the character '.'. All forbidden squares will be represented with the grid character '!'. All other squares will be capital letters, representing various teleportation squares. If a letter appears in the grid, then it will appear in at least two separate grid squares.

Partial Credit Input Restrictions:

In some of the input cases (enough to allow a maximum score of 70%), no teleportation squares will exist.

In some of the input cases (enough to allow a maximum score of 90%), no letter will appear in the grid more than 10 times.

In the last set of input cases worth 10% of the grade (to raise your grade from 90% to 100% max), there are no restrictions on the number of times an individual letter appears in the grid (beyond the size of the grid and the other grid square requirements).

The Output:

If Sastry can get out of the maze, output a single integer representing the fewest number of moves it will take him to get out. If he can't get out, output -1.

Sample Input

Sample Output

3 4 .\$!* !!!.	8
4 2 .. *! !\$..	-1
6 6 C..\$B. !!!!!!!CB .*.AAA C.B.CB	4