Maze Solver
==========

In a given maze, your goal is to find a path from 'S' to 'F'. You can move only up, down, left and right directions. The path cannot pass through a position that is visited before.

Writer a program that outputs the path.

PROBLEM NAME: maze

INPUT FORMAT:

* Line 1: Has two integers; the rows 3 <= N <= 100, and the columns 3 <= M <= 100 of the maze.

* Line 2..N+1: The maze composed of '#' and '.'. '#' means a wall where as '.' is empty spot. 'S' indicates the starting position and 'F' is the ending position.

SAMPLE INPUT:

```
5 7
#######
#F#...#
#.#.#.#
#...#S#
#######
```

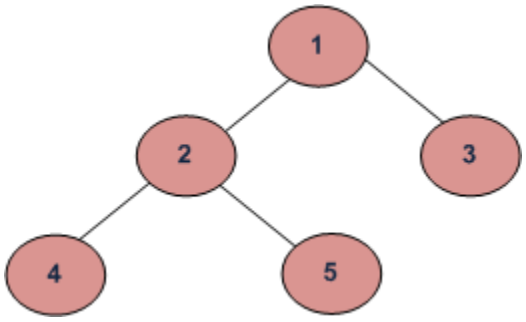
OUTPUT FORMAT:

The output should be a path from 'S' to 'F' as a string. The path is composed of characters 'U' (up), 'D' (down), 'L' (left), 'R' (right). There may be many right paths; just output any of them. It doesn't have to be the shortest path either. The length of the path has to be less than 10,000.

SAMPLE OUTPUT:

```
UULLDDLLUU
```

What I am struggling with: Right now, when my code does a DFS and it comes across a dead end, it jumps immediately back to the latest unvisited node. However, the problem requires me to keep track of the path taken so I need to either add a backtracking route to my path string or I need to erase the route that takes me to a dead end and go down a different path. I don't know how to write the code for that.




In the above example, if I had to go from Node 1 to Node 3, my code is currently printing out the path as 12453, however it should be printing out 12425213 or 13 (in this case it would erase the path that takes it to the dead ends).