Introduction

Ever wondered how computers play games? Want to make that ?
In this challenge train model to solve the game **Tile-Slider**, a replica of game Match the Tiles.

Game Description

The Game board is a grid with some obstacles that cannot be passed through. There are some source tiles (colored squares with holes in the center) and corresponding destination positions (denoted by the colored circles). Our target is to move all source tiles to their respective destination positions. Moves allowed are up, down, left, right (by swiping in the original game, by printing out a letter U, L, D or R in here). When you play one move, say you choose to swipe down, then all source tiles will move as far down as they can i.e will stop only when there is no way down or they have hit some obstacle or another tile (and that tile cannot go any more down too). Note that only the source tiles will move not the destination tiles.

Dataset

Each test file contains a grid . You need to output a set of moves for each test case. Every source should reach its respective destination tile with the output set of moves. Lesser the number of moves required to reach destination better the score is.

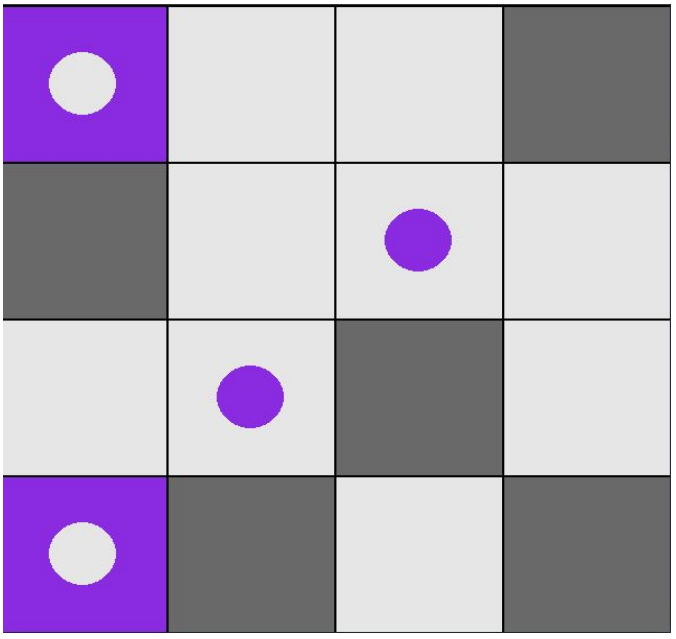
Input format:

First line contains two integers N, M where N represents the number of rows in the grid and M number of columns in the grid. Following N lines contain M characters each where each character is either . or # where # signifies obstacles which can not be passed through and . free space. Following that will be an integer S representing the number of source tiles. Then below S lines each will be of format <row_source> <column_source> <row_destination> <column_destination>. It is guaranteed that in all test files any source/destination position does not collide with obstacles. 0 based indexing is followed.

Example:

```
4 4
...#
#...
..#.
.#.#
2
0 0 1 2
3 0 2 1
```

A winning set of moves for this grid:
URD



Files

Dataset comprises 2000 files having input format as mentioned above. These are of varying grid sizes.
Download dataset here

Submission

Submit a csv file containing headers[filename, moves]. This file should contain solutions for all 2000 puzzles. Each row of file will contain a test filename (under filename column) and a string containing a set of moves (under moves column).
eg. sample_1.txt UDDLRLU

Evaluation Criteria

The primary metric is the number of puzzles you solve.
If two teams solve the same number of puzzles, the ties will be broken on the basis of a lower total number of moves to solve all the puzzles combined.

Simulator

We offer a python library to visualize this problem in PyGame, as well as helper functions which will allow you to start off with getting the valid moves, maintaining the state, etc. easily.