



DISCO Labyrinth Challenge

DISCO is currently looking to build a huge labyrinth on some premises represented by an $n \times m$ grid of unit cells. Out of those, some cells are *blocked* by pillars, while the remaining cells are called *free*. It is also possible that there are no pillars. In order to transform the premises into a maze, some free cells will have to be filled up with concrete. In a valid maze, the following property has to be satisfied: for every pair (a, b) of free cells, there must be exactly one *simple path* between a and b consisting only of moves between cells sharing a side (i.e. up, down, left, right).

To maximize the fun, DISCO wants the labyrinth to have as many dead ends as possible. A dead end is a free cell with exactly one free neighbor cell.

You are given the map of the premises as input. You will have to aid DISCO in building a valid labyrinth with as many dead ends as you can.



Implementation Details

You are given 10 input files, each describing a scenario for how the premises may look. For each input file you should submit an output file with a valid labyrinth that can be constructed on the DISCO premises. Your score will reflect how many dead ends the labyrinth has.

Input Format

An input file describes the grid representing the premises and gives a parameter k , used for scoring. The format is as follows:

- Line 1: $n m k$
- Line $1 + i$ (for $1 \leq i \leq n$): row i of the grid, which is a length m string, consisting of the following characters:
 - ‘.’: a free cell,
 - ‘#’: a pillar.

Output Format

Line i (for $1 \leq i \leq n$) describes row i of the labyrinth (the premises, after some free cells have been filled). It is a length m string, consisting of the following characters (without any whitespace):

- ‘.’: a free cell,
- ‘#’: a pillar,
- ‘X’: a cell filled with concrete.

Scoring

An output file is *valid* if the following two conditions are met:

1. The output premises match the input ones, with the only exception that some (potentially 0) ‘.’ characters in the input have been changed to ‘X’ (cells which got filled up with concrete).
2. The output should be a valid labyrinth, as described above.

If your output is not valid, the score for that test case will be 0. Otherwise, the score will be $10 \cdot \min(1, \ell/k)$ points, rounded down to two digits after the decimal point, where ℓ is the number of dead ends in the output labyrinth and k is the number provided in the input. **It is guaranteed that for each test case there exists a solution that scores 10 points.**

Caveats. If your solution is valid but still scores 0 points according to the formula, the verdict you will see in the submission interface will be “Wrong Answer.” If your solution gets a positive score that is less than 10 on a test, then you will see the verdict “Partially Correct.”

Example

Consider the following input:

```
4 5 5
....#
.#...#
...#.
....#
```

Below are two possible valid output labyrinths:

```
.X.X#      .X.X#
.#...#     .#...#
...#X      ...#X
XX...#     .X...#
```

They have 4 and 5 dead ends, respectively, so they earn $10 \cdot 4/5 = 8$ and $10 \cdot 5/5 = 10$ points. The dead ends are marked with ‘o’ below:

```
oXoX#      oXoX#
.#.o#      .#.o#
...#X      ...#X
XX.o#      oX.o#
```

Note that the following three outputs would not be valid:

```
.XXX#      ...X#      XXXX#
.#XX#      .#.X#      X#XX#
...#       ...#X      ..X#X
XX...#     XXXX#      ..XX#
```

For the left one, there is no path between the rightmost two cells. For the two other outputs, for every pair of distinct free cells there are exactly two distinct simple paths between them.

Registration Instructions

You will have to go to <http://82.77.33.94:8860/> and hit ‘Register’.

- If you are competing alone: put your DISCO quiz nickname in the “First Name” field. Complete your email in the “email” field with your ETH email.
- If you are competing in a team (of at most 3), do the same, but write all nicknames in the “First Name” field, separated by commas: `nick1, nick2, ...`. For the “email” field, select a team “captain” and use their ETH email.

The interface requires filling in the “Last name” field, but we will not use it, so just complete with a “-”. You are free to choose whatever username you want. For the password, we strongly recommend using a unique password for this challenge (e.g. use a trusted random password generator). We only store your passwords hashed.

Submission Instructions

Go to the link above and log in. Inside, check out the “Statement” and “Submissions” tabs under “Labyrinth.” In the former you will find this task description together with the 10 inputs you will have to solve. In the latter you can upload your solutions and get real-time feedback for how well they are doing. You can submit as many times as you want, but please be mindful of the server load.

The total score of your best attempt at each test is displayed at the top of the page in “Submissions.” This will be used to determine the challenge winners. Note that for each test case we only look at your best attempt for that test (which might come in different submissions for different tests).

You have the option to either upload your answers individually or as a zip file. Please also upload the source code you used to generate your outputs in a file called “`code.txt`” file (see the interface). If you generated any outputs by hand, feel free to not upload any “`code.txt`” file” You can use PYTHON, or any other programming language you are comfortable with. If you change your code between submissions, please upload the modified code again with your next submission, so that each submitted file comes with the code that generated it. You don’t need to include instructions for compiling/running your code. Your code will not be run, but may be used for detecting fraud.

Finally, if you have any questions about the task or the challenge in general, use the “Communication” tab, where you can ask us questions. Please note that there is no provision for asking how to solve the problem, that is your challenge after all, so reserve the functionality for clarifications about the task description and any critical technical problems you might encounter.

Deadline

The challenge ends at 23:59, 19 Oct 2022. Submissions after this time will not be accepted.