UNSW ICPC Workshop T3W4 - Graph Theory
Hard Problem Set
Source: South Pacific Divisionals and Regionals (various years)

Discuss the problems in this document and try to solve them with your group. Make sure everyone is comfortable with the solution before moving on. Ask us if you need help, or want to check your solution.

The first problem is easier, and the other three are harder and roughly similar in difficulty, so feel free to do them in any order.

You can submit to all problems by finding the corresponding contest on this page:
https://prog4fun.csse.canterbury.ac.nz/course/view.php?id=2&section=3

Problem I
Iguana Date
Time limit: 2 seconds

Iggy the Iguana is back! This time, he brought his date, Ignia, with him to the corn maze. The corn maze is a square grid where some of the cells are blocked off with impassable corn plants and others are cleared out. Iggy and Ignia can only travel in and through cells that are cleared out. They can move to a cell in any of the four cardinal directions (north, south, east, and west).

The corn maze they chose for their date is special in that every cell that is cleared out can be reached from the start (the top-left of the maze) through exactly one direct path (without revisiting cells).

Iggy and Ignia start in the top-left cell. They stay in that cell for one minute, then move to an adjacent cell. Every minute Iggy and Ignia will move to an adjacent cell. When they reach the bottom-right cell, they stay there for one minute and then leave the maze and their date ends. Both of the iguanas are having a wonderful time and do not want the fun times to end. Therefore, they will choose a path through the maze (possibly revisiting cells) that makes their date last as long as possible. However, Iggy and Ignia do not want to get lost forever, so they will never enter a cell from the same direction more than once.

How long can Iggy and Ignia make their date last?

Input

The first line of input contains a single integer n ($2 \leq n \leq 100$), which is the length of each side of the square grid representing the maze.

The next n lines describe the grid. Each of these lines contains a string of length n containing only the characters '`.`' and '#'. A '`.`' represents a cell that is cleared out and which Iggy and Ignia can travel through, and a '#' represents a cell blocked off with corn plants.

Output

Display the length of Iggy and Ignia's date.

Sample Input 1

```
3
...
.#.
#..
```

Sample Output 1

```
7
```

Sample Input 2

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

Sample Output 2

```
15
```

Sample Input 3

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

Sample Output 3

```
13
```

Problem F
Fusion
Time limit: 15 seconds

What a time to be alive! Today, the very first *cold fusion* reactor has just been built and cleared for operation. Of course, when the reactor is started for the first time, nothing happens.

Bewildered physicists pore over their calculations looking for errors. Countless double-checks and triple-checks are made before a trio of engineers discover the problem: two atoms in the reactor have become α-stuck!

The reactor can be thought of as a grid of cells with R rows and C columns. Each cell is either empty, contains a deuterium atom (two of which are α-stuck) or is blocked off by a control rod to prevent runaway reactions. Each cell is adjacent to the four cells to its left, right, top and bottom (cells on the edge of the reactor are adjacent to fewer than four cells).

You can issue fusion instructions to the reactor by specifying two atoms to be fused. Be wary; two atoms can be fused only if they are in adjacent cells or there is a path of adjacent empty cells from one atom to the other. When two atoms are fused, they will produce helium which floats away, effectively removing the two original atoms from the reactor (leaving the cells empty).

In order to fix the problem in the reactor, you will need to fuse the α-stuck atoms with each other. What is the fewest number of instructions you need to give to the reactor in order to fuse the α-stuck atoms?

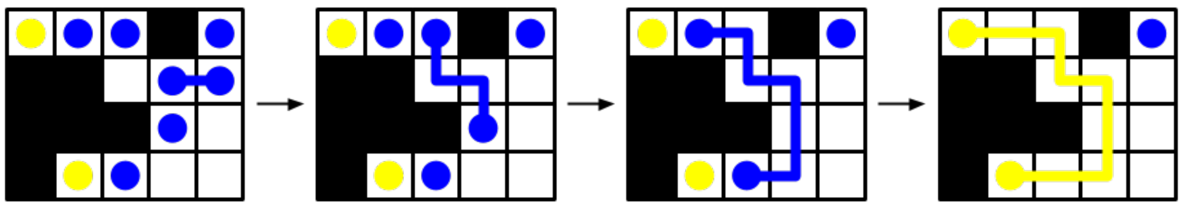


Figure F.1: This is the reactor for Sample Input 1. Black squares denote blocked off cells, blue circles denote deuterium atoms and the yellow circles denote the two α-stuck atoms. One optimal sequence of fusions is shown.

Will you and your fellow engineers overcome the final barrier to *cold fusion*?

Input

The first line of input contains two integers R ($1 \leq R \leq 1000$), which is the number of rows in the reactor, and C ($1 \leq C \leq 1000$), which is the number of columns in the reactor. The next R lines describe the reactor. Each of these lines is a string with exactly C characters.

- `'.'` denotes an empty cell.
- `'#'` denotes a blocked cell.
- `'O'` denotes a cell containing a deuterium atom.
- `'A'` denotes a cell containing an α-stuck atom.

There are exactly two `'A'`s in the input.

Output

Display the minimum number of fusion instructions needed to fuse the α-stuck atoms. If it is not possible to fuse the α-stuck atoms together, display `-1` instead.

Sample Input 1

```
4 5
AOO#O
##.OO
###O.
#AO..
```

Sample Output 1

```
4
```

Sample Input 2

```
2 5
..O..
A###A
```

Sample Output 2

```
-1
```

Problem K
Karaoke Meetup
Time limit: 8 seconds

The judges of the South Pacific Programming Contest are planning their next secret karaoke meetup and are looking for a place to hold it. Last time, they asked Timothy to pick the location, but of course, he just picked somewhere really close to his house, and far from everyone else's! This year, you would like to pick a meeting location that is fair.

All the judges live in the same city. The city consists of various locations in which the meeting could be held and roads that connect pairs of locations. The city is built such that for each pair of locations, there is exactly one path between them. Each road has a length and can be used to travel in either direction. You consider a meeting point fair if the distances from each of the judges' houses are similar. For each location, its *fairness score* is the ratio $\frac{A}{B}$, where A is the minimum distance from the location to any judges' house and B is the largest distance. What is the maximum fairness score for all vertices?

Input

The first line contains two integers n ($2 \leq n \leq 200\,000$), which is the number of locations in the city, and k ($2 \leq k \leq n$), which is the number of judges.

The next k lines describe the location of the judges' houses. Each of these lines contains a single integer ℓ ($1 \leq \ell \leq n$), which is the location of this judge's house. No two judges live at the same location.

The next $n - 1$ lines describe the roads in the city. Each of these lines contains three integers u ($1 \leq u \leq n$), v ($1 \leq v \leq n$), and w ($1 \leq w \leq 10^9$), which denotes a road between locations u and v with a length of w.

Output

Display the maximum fairness score as a reduced fraction.

Sample Input 1
```
3 2
2
3
1 2 1
1 3 1
```

Sample Output 1
```
1/1
```

Sample Input 2
```
2 2
1
2
1 2 10
```

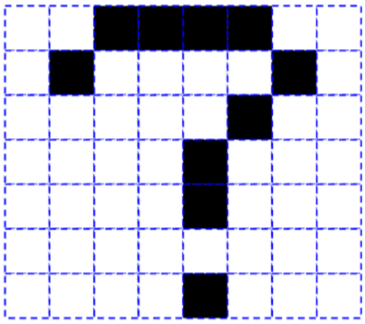
Sample Output 2
```
0/1
```

Sample Input 3
```
5 3
5
2
4
3 1 1
1 4 1
1 2 1
3 5 1
```

Sample Output 3
```
1/2
```

Sample Input 4	Sample Output 4
10 4 1 5 8 9 3 4 5 3 2 20 4 9 5 2 8 6 6 2 3 5 7 7 10 2 4 4 1 17 7 2 5	5/16

Problem F
Flip Tool
Time limit: 15 seconds



Orso is using some simple software to create 2D images. Unfortunately, the software has stopped working correctly. Only the flip tool is working. Orso has already drawn some pixels black and some pixels white (every pixel is either black or white). He wants to get back to a state where every pixel is the same colour.

The way the flip tool works is by first having the user select a pixel. After this pixel is selected, the flip tool finds the largest 8-connected region of pixels all of the same colour containing the selected pixel, then flips the colour of each pixel in that region (from white to black or from black to white). In other words, the flip tool starts at the selected pixel, then selects each adjacent pixel that shares an edge or corner with it, and which has the same colour as it, then it selects each of their neighbours of the same colour, and so on. All the pixels selected this way have their colour flipped.

What is the fewest number of flip tool applications required to get to a state where every pixel is the same colour?

Input

The first line of input contains two integers R ($1 \le R \le 1\,000$) and C ($1 \le C \le 1\,000$), which are the number of rows and columns of pixels in the image respectively.

The next R lines describe the image. Each of these lines contains a string of length C containing only the characters '`.`' and '`x`'. A '`.`' represents a white pixel and a '`x`' represents a black pixel.

Output

Display the minimum number of flip tool operations required.

Sample Input 1

```
6 6
.xx...
xxx...
....xx
......
xxxxxx
......
```

Sample Output 1

```
2
```

Sample Input 2

```
2 8
..xx.xx.
x..xxx.x
```

Sample Output 2

```
2
```

Sample Input 3	Sample Output 3
7 8 ..XXXX.. .X....X. .....X.. ....X... ....X... ....X...	1