

# Rotten Oranges

\*\*\*\*\*

You are given a 2D array of integers of size  $N * M$ . Each of the cell contains either of these 3 integers: 0, 1, 2. The integer 2 denotes a rotten orange, 1 denotes a fresh orange and 0 denotes an empty cell.

Each rotten orange can rot fresh oranges in 4 adjacent cells in 1 unit of time. The 4 cells are left, right, top and bottom cell.

For a given matrix, find the minimum units of time in which all oranges become rot.

Return -1, if it is not possible.

## Input Format:

The first line of input contains 2 space separated integers, N and M, that denotes size of given 2D array.

The following N lines contains M space separated integers, that denotes the value of cells of given 2D array.

## Constraints:

Value of N and M lies in the range: [1, 10000].

Value of each cell in 2D array can be either 0, 1 or 2.

## Output Format:

Print the required integer, as described in the problem description.

## Sample Input 1:

```
3 5
2 1 0 2 1
1 0 1 2 1
1 0 0 2 1
```

## Sample Output 1:

```
2
```

## Explanation:

In the first unit of time, fresh oranges at coordinates: (0, 1), (0, 4), (1, 0), (1, 2), (1, 4), (2, 4).

In the second unit of time, fresh orange at coordinate: (2, 0) gets rot. Hence, in 2 units of time, all the fresh oranges become rot.

## Sample Input 2:

```
3 5
2 1 0 2 1
1 0 1 0 1
1 0 0 0 2
```

## Sample Output 2:

```
-1
```

## Explanation:

It is impossible to rot the fresh orange at (1, 2).