Description | Hints | Submissions | Discussions | Notes

Infected People

Difficulty	Time Limit	Memory	Score
[illegible]	1 sec	256000KB	100

80/80 XP | 30/30

Description

Your city is having people infected with a virus. The city in which you live is represented as a grid consisting of **n** rows and **m** columns. Cells containing **2** are the cells where the people infected with the virus are present and the cells having **1** are the cells having people not yet infected with the virus. There are certain empty cells which are represented by **0**. The infected people in a unit time can infect **all** their adjacent cells, i.e, if they are present at cell [i, j] they can infect cells [i−1, j], [i+1, j], [i, j−1] and [i, j+1]. The virus **cannot** pass through empty cells. Your task is to print the minimum time in which all the people are infected with the virus. If the virus cannot infect everyone, print **−1**.

Input Format

The first line contains two integers **n** and **m** — the number of rows and columns, respectively.

The following **n** lines contain **m** integers each, the j-th element in the i-th line is the number written in the j-th cell of the i-th row.

Output Format

Print the minimum time in which everyone can be infected or **−1** if everyone cannot be infected.

Constraints

Copy

```
1 <= n <= 1000
1 <= m <= 1000
```

Sample Input 1

Copy

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

C++14 | 00:00:00 | 12 px

```
{
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cin >> n >> m;
    vis.assign(n, vector<int>(m, 0));
    grid.assign(n, vector<int>(m, 0));
    closest_infected_person_distance.assign
    (n, vector<int>(m, 1e9));
    for (int i = 0; i < n; i++)
    {
        for(int j=0;j<m;j++){
            cin>>grid[i][j];
            if(grid[i][j]==2){
                infected_people.push_back
                (make_pair(i,j));
            }
            else if(grid[i][j]==1){
                not_infected_people.
                push_back(make_pair(i,j));
            }
        }
    }
    Multi_source_bfs(infected_people);
    int ans=INT_MIN;
```

Sample Tests | Manual Tests

Test Case 1 | Test Case 2

ACCEPTED

Input

Console | Run on Sample