PRACTICE COMPETE JOBS LEADERBOARD Search swatantragoswam1

All Contests > tp14-3d > Rat in a Maze - Mul jumps

Rat in a Maze - Mul jumps

locked

Problem Submissions Leaderboard

A Maze is given as N * N binary matrix of blocks where source block is the upper left most block i.e., maze[0][0] and destination block is lower rightmost block i.e maze[N-1][N-1] A rat starts from source and has to reach destination. The rat can move only in two directions: forward and down. In the maze matrix, 0 means the block is dead end and non-zero number means the block can be used in the path from source to destination. The non-zero value of mat[i][j] indicates number of maximum jumps rat can make from cell mat[i][j].

If There is path in both Direction than rat move to down First & if rat can move to two differrent cell after direction selection he tries to move near by cell first

In this variation, Rat is allowed to jump multiple steps at a time instead of 1. There exists a single path, find it.

If its impossible to reach destination , print -1.

Input Format

- Line 1 contains N , size of the matrix.
- Next N lines contain N integers.

Constraints

- N <= 10
- matrix elements <= N

Output Format

- Print a N * N binary matrix where 1 denotes the cells used to reach destination, 0 otherwise.

Sample Input 0

```
4
2 1 0 0
3 0 0 1
0 1 0 1
0 0 0 1
```

Sample Output 0

```
 1  0  0  0
 1  0  0  1
 0  0  0  1
 0  0  0  1
```

Explanation 0

Rat started with M[0][0] and can jump upto 2 steps right/down. Let's try in horizontal direction - M[0][1] won't lead to solution and M[0][2] is 0 which is dead end. So, backtrack and try in down direction. Rat jump down to M[1][0] which eventually leads to solution.

Submissions: 14

Max Score: 10

Rate This Challenge:

☆ ☆ ☆ ☆ ☆

More

Current Buffer (saved locally, editable)

C++

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
#include<bits/stdc++.h>
using namespace std;
static int maze[10][10];
static int path[10][10];
bool dfs(int n,int x,int y)
{
    if(x>=n||y>=n) return false;

    path[x][y]=1;
    if(x==n-1&& y==n-1)
        return true;
    int step=maze[x][y];
    for(int d=1;d<=step;d++)
    {
        if(dfs(n,x,y+d)) return true;

        if(dfs(n,x+d,y)) return true;
    }
    path[x][y]=0;
    return false;
}
```

```
void findPath(int n)
{
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
            path[i][j]=0;
    }
    if(!dfs(n,0,0))
    {
        cout<<"-1\n";
        return ;
    }
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            cout<<" "<<path[i][j]<<" ";
        }
        cout<<"\n";
    }
}

int main() {

        int n;
    cin>>n;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            cin>>maze[i][j];

    findPath(n);

    return 0;
}
```

Line: 60 Col: 2

Upload Code as File □ Test against custom input

Run Code Submit Code

Testcase 0 ✓

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
4
2 1 0 0
3 0 0 1
0 1 0 1
0 0 0 1
```

Your Output (stdout)

```
 1   0   0   0
 1   0   0   1
 0   0   0   1
 0   0   0   1
```

Expected Output

```
 1   0   0   0
 1   0   0   1
 0   0   0   1
 0   0   0   1
```