Graph Theory - S001 (E017)

Solved Challenges 0/2

Back To Challenges List

Path Exists from Source to Destination Cell

ID:11090 Solved By 655 Users

The program must accept a matrix of size **R*C** and the indices of two cells (Source and Destination) in the matrix as the input. The matrix contains only **1's** and **0's**. The cell value **1** indicates the presence of a path. The cell value **0** indicates the presence of a stone (i.e., no path). The movement from one cell to another can be in the **left**, **right**, **bottom** and **top** directions. The program must print **yes** if there is a path from the given source cell to the destination cell. Else the program must print **no** as the output.

Boundary Condition(s):
2 <= R, C <= 50

Input Format:
The first line contains R and C separated by a space.
The next R lines, each containing C integers separated by a space.
The $(R+2)^{nd}$ line contains two integers representing the indices of the source cell.
The $(R+3)^{rd}$ line contains two integers representing the indices of the destination cell.

Output Format:
The first line contains yes or no.

Example Input/Output 1:
Input:
4 5
1 0 1 1 0
0 1 0 1 1
1 1 0 1 0
1 1 1 1 1
1 1
1 4

Output:
yes

Explanation:
One of the possible paths from the source cell to the destination cell in the matrix is highlighted below.
1 0 1 1 0
0 **1** 0 **1** **1**
1 **1** 0 **1** 0
1 **1** **1** **1** 1

Example Input/Output 2:
Input:
3 3
1 0 1
0 1 1
1 0 1
0 2
2 0

Output:
no

Max Execution Time Limit: 500 millisecs

Ambiance

C (gcc 8.x)

Reset

```
#include<stdio.h>
#include<stdlib.h>

int R,C,found=0;

void traverse(int matrix[R][C], int row,int col)
{
    if(row>=0 && row<R && col>=0 && col<C)
    {
        if(matrix[row][col]==0 || matrix[row][col]==2)
        return;

        matrix[row][col] = 2;
    }

    if(!found)
    traverse(matrix,row,col+1);
    if(!found)
    traverse(matrix,row,col-1);
    if(!found)
    traverse(matrix,row-1,col);
    if(!found)
    traverse(matrix,row+1,col);
}

int main()
{
    scanf("%d%d", &R,&C);
    int matrix[R][C];
    for(int row=0;row<R;row++)
        for(int col=0;col<C;col++)
            scanf("%d",&matrix[row][col]);

    int sourceR,sourceC,destR,destC;
    scanf("%d%d%d%d",&sourceR,&sourceC,&destR,&destC);

    if(sourceR==1 && sourceC==1 && destR==1 && destC==1)
    {
        traverse(matrix,sourceR,sourceC);
    }

    if(found==1)
    printf("YES");
    else
    printf("NO");
}
```

Code did not pass the execution

Your Program Output:

NO

Save Run

Run with a custom test case (Input/Output)