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:

45

10110

01011

11010

11111

11

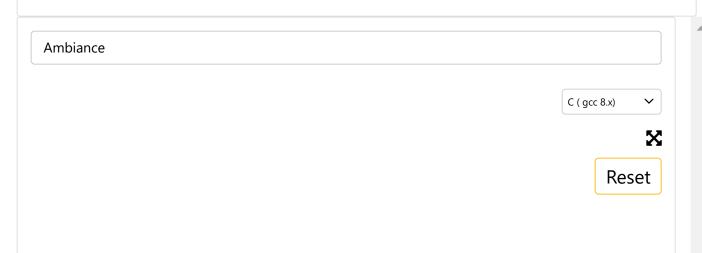
14

Output:

yes

https://www.skillrack.com/faces/candidate/codeprogram.xhtml Explanation: One of the possible paths from the source cell to the destination cell in the matrix is highlighted below. 10110 01011 1 **1** 0 **1** 0 1 1 1 1 1 **Example Input/Output 2:** Input: 3 3 101 011 101 0 2 20 Output: no

Max Execution Time Limit: 500 millisecs



```
#include<stdio.h>
 1
 2
    #include<stdlib.h>
 3
 4
    int R,C,sourceR,sourceC,destR,destC,found=0;
 5
 6
    void traverse(int matrix[R][C], int row,int col)
 7
    {
 8
        if(row>=0 && row<R && col>=0 && col<C)
9
10
            if(row == destR && col == destC)
11
             {
12
                 found = 1;
13
                 return;
14
            }
15
            if(matrix[row][col]==0 || matrix[row][col]==2)
16
            return;
17
18
19
            matrix[row][col] = 2;
20
21
22
        traverse(matrix,row,col+1);
        if(!found)
23
        traverse(matrix,row,col-1);
24
25
        if(!found)
        traverse(matrix,row-1,col);
26
27
        if(!found)
        traverse(matrix,row+1,col);
28
29
        }
30
    }
31
32
    int main()
33
    {
34
        scanf("%d%d", &R,&C);
        int matrix[R][C];
35
        for(int row=0;row<R;row++)</pre>
36
            for(int col=0;col<C;col++)</pre>
37
                 scanf("%d",&matrix[row][col]);
38
39
40
41
        scanf("%d%d%d%d",&sourceR,&sourceC,&destR,&destC);
42
43
        if(matrix[sourceR][sourceC]==1 && matrix[destR][destC]==1)
44
45
            traverse(matrix, sourceR, sourceC);
46
        }
47
        printf(found==1?"yes":"no");
48
49
    }
```

