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):

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

Explanation:

One of the possible paths from the source cell to the destination cell in the matrix is highlighted below.

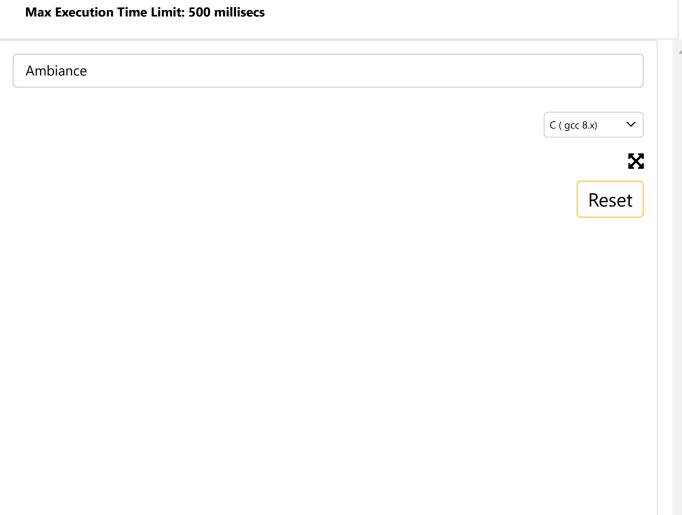
10110

01011

11010

11111

Example Input/Output 2: Input: 3 3 1 0 1 0 1 1 1 0 1 0 1	•
2 0 Output: no Max Execution Time Limit: 500 millisecs	imit: 500 millisecs



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

