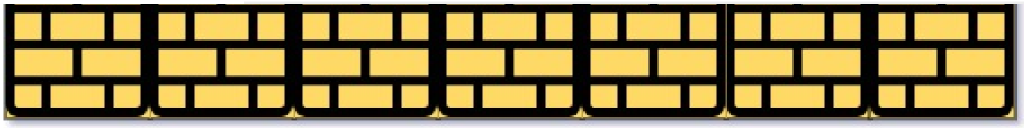
Description Solution Discuss (561) Submissions *i* C++ Aut



```
Input: maze = [[0,0,1,0,0],[0,0,0,0,0],[0,0,0,1,0],
[1,1,0,1,1],[0,0,0,0,0]], start = [0,4], destination = [3,2]
Output: false
Explanation: There is no way for the ball to stop at the
destination. Notice that you can pass through the destination
but you cannot stop there.
```

Example 3:

```
Input: maze = [[0,0,0,0,0],[1,1,0,0,1],[0,0,0,0,0],
[0,1,0,0,1],[0,1,0,0,0]], start = [4,3], destination = [0,1]
Output: false
```

Constraints:

- `m == maze.length`
- `n == maze[i].length`
- `1 <= m, n <= 100`
- `maze[i][j]` is `0` or `1`.
- `start.length == 2`
- `destination.length == 2`
- $0 <= start_{row}, destination_{row} <= m$
- $0 <= start_{col}, destination_{col} <= n$
- Both the ball and the destination exist in an empty space, and they will not be in the same position initially.
- The maze contains **at least 2 empty spaces**.

Accepted 133,731 Submissions 241,177

Seen this question in a real interview before? Yes No

Companies *i*

Related Topics

Similar Questions

```
class Solutic
    vector<ve
public:
    bool isVc
y, int n, int
vector<vector
    {
        if(x
x >= n || y >
[y] == 1) ret
        retur
    }
    bool
canWeReach(ve
>& maze,int x
y,vector<int>
destination,i
    {
        maze[


        if(x
destination[0
destination[1
            r


        for(i
        {
            i
newy = y;

 while(isVali
[0], newy+dir
maze))
            {

dirs[i][0];

dirs[i][1];
            }


            i
[newy] != 2 &
canWeReach(ma
newy,destinat

        }


        retur
    }
```

Your previous code was re

Problems Pick One ‹ Prev /149 Next › Contribute *i* ▶ Run Code