

Description

Solution

Discuss (999+)

Submissions

C#

200. Number of Islands

Medium 11796 301 Add to List Share

Given an $m \times n$ 2D binary grid `grid` which represents a map of '1' s (land) and '0' s (water), return *the number of islands*.

An **island** is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

Example 1:

Input: `grid = [`
`["1","1","1","1","0"],`
`["1","1","0","1","0"],`
`["1","1","0","0","0"],`
`["0","0","0","0","0"]`
`]`

Output: 1

Example 2:

Input: `grid = [`
`["1","1","0","0","0"],`
`["1","1","0","0","0"],`
`["0","0","1","0","0"],`
`["0","0","0","1","1"]`
`]`

Output: 3

Constraints:

- `m == grid.length`
- `n == grid[i].length`
- `1 <= m, n <= 300`
- `grid[i][j]` is '0' or '1' .

```

1 public class Solution {
2     public int NumIslands
3     [] grid) {
4         if (grid == nul
5         grid.Length == 0) {
6             return 0;
7         }
8         var nr = grid.L
9         var nc = grid[0
10
11
12         var noOfIslands
13         for(var r =0; r
14             r++)
15             {
16                 for(var c=0
17                     c++)
18                     {
19                         if(grid
20                         '1')
21                         {
22                             ++noOfIslands;
23                             gri
24                             '0';
25
26                             Que
27                             neighbors = new Queue<i
28                             neighbors.Enqueue(r *
29                             (neighbors.Count > 0) {
30                             neighbors.Dequeue();
31                             = id / nc;
32                             = id % nc;
33                             - 1 >= 0 && grid[row-1]
34                             '1') {
35                             neighbors.Enqueue((row
36                             + col));
37                             grid[row-1][col] = '0'
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Your previous code was restored from y

Console

Problems

Pick One

Prev

6/30

Next

Run Code

Submit