

# 695. Max Area of Island

Medium 6529 154 Add to List Share

You are given an `m x n` binary matrix `grid`. An island is a group of `1`'s (representing land) connected **4-directionally** (horizontal or vertical.) You may assume all four edges of the grid are surrounded by water.

The **area** of an island is the number of cells with a value `1` in the island.

Return *the maximum **area** of an island in `grid`*. If there is no island, return `0`.

## Example 1:

0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0
0	1	1	0	1	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	0	1	0	1	0	0
0	1	0	0	1	1	0	0	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0

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

**Output:** 6

**Explanation:** The answer is not 11, because the island must be connected 4-directionally.

## Example 2:

**Input:** `grid = [[0,0,0,0,0,0,0,0]]`

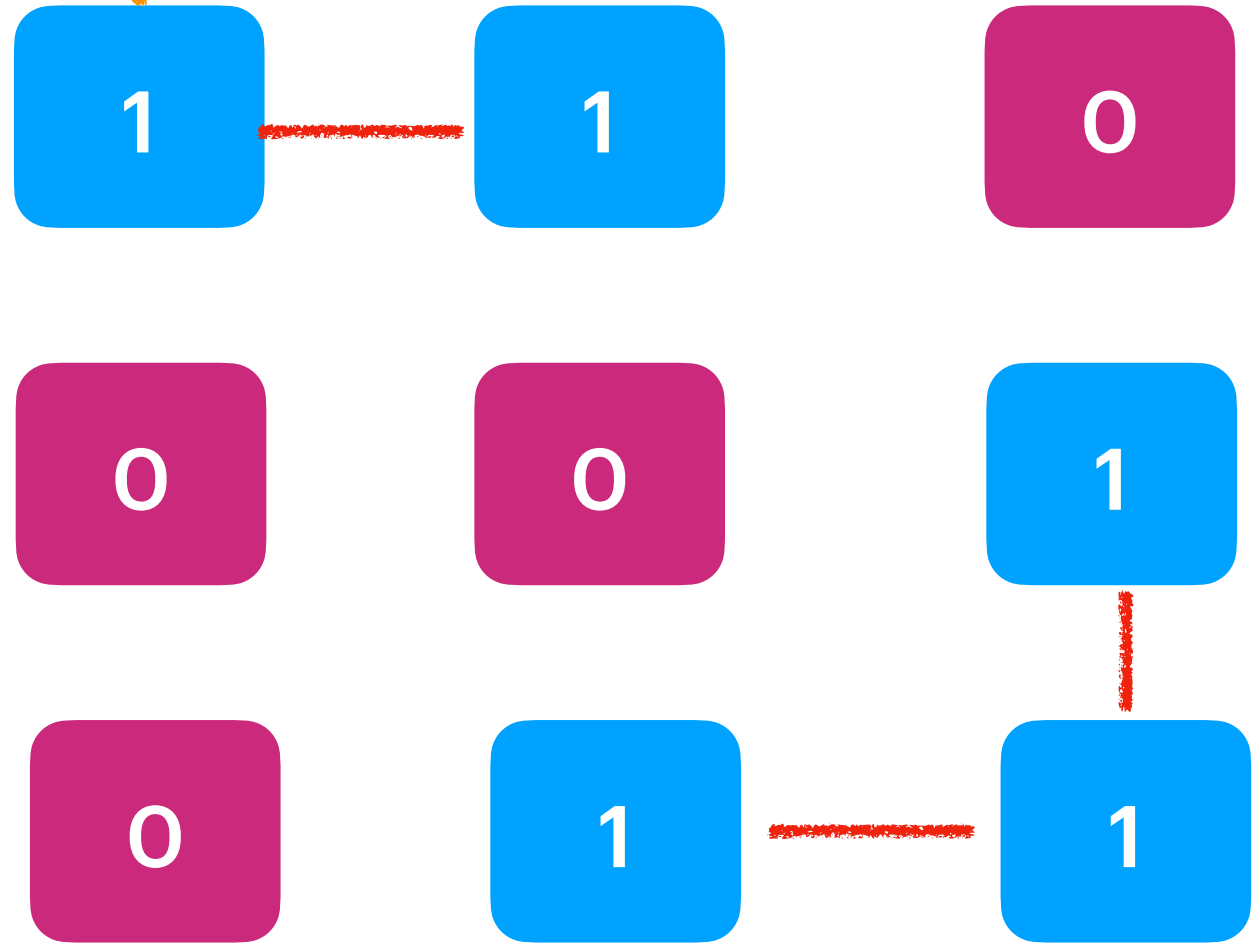
**Output:** 0

## Constraints:

- `m == grid.length`
- `n == grid[i].length`
- `1 <= m, n <= 50`
- `grid[i][j]` is either `0` or `1`.

Start BFS(row:0, col:0)  
because grid[row][col] = 1

MaxSize = 0



Size = 0

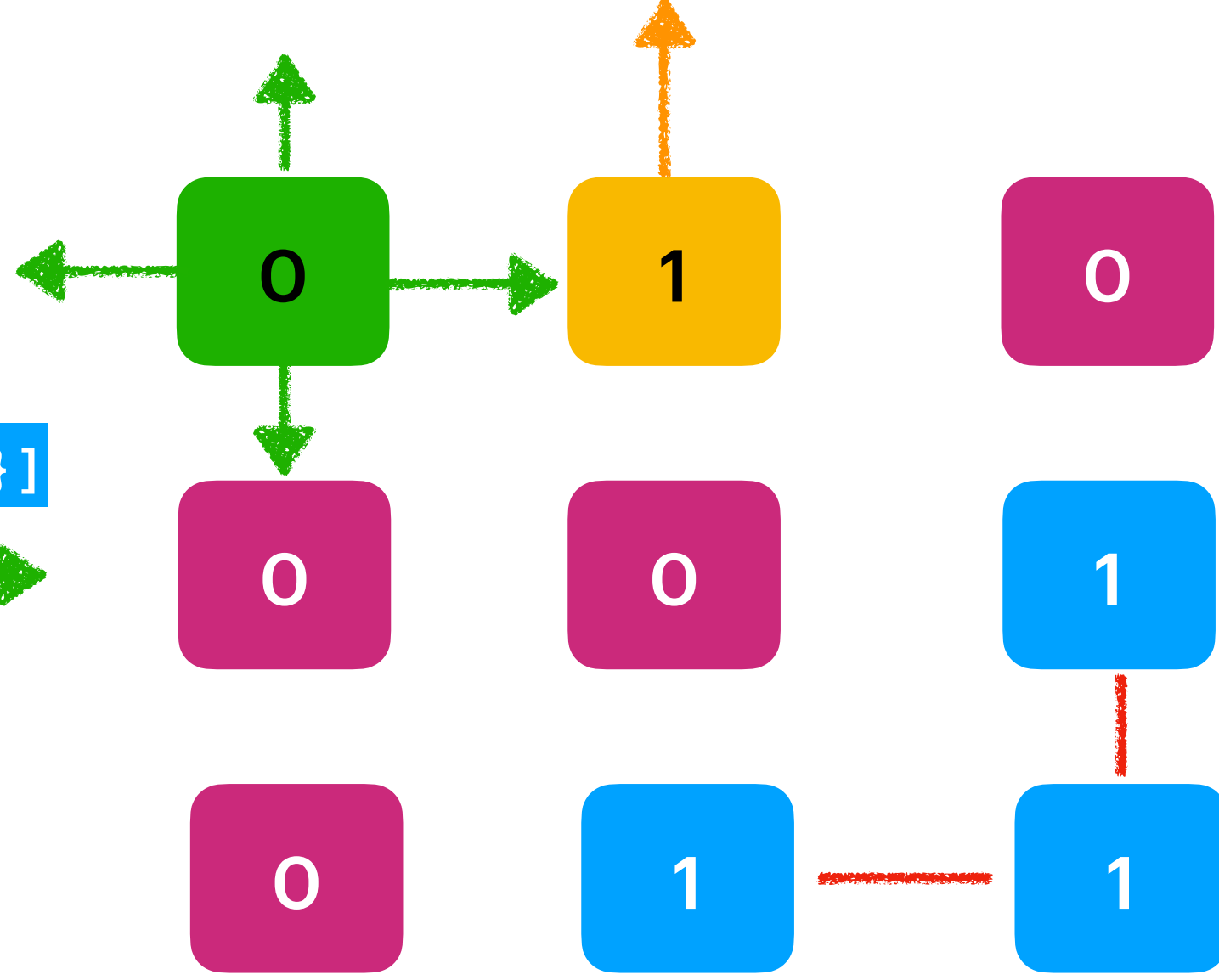
Input  
Queue[{0,0}]



Connected Island Goes to Queue:

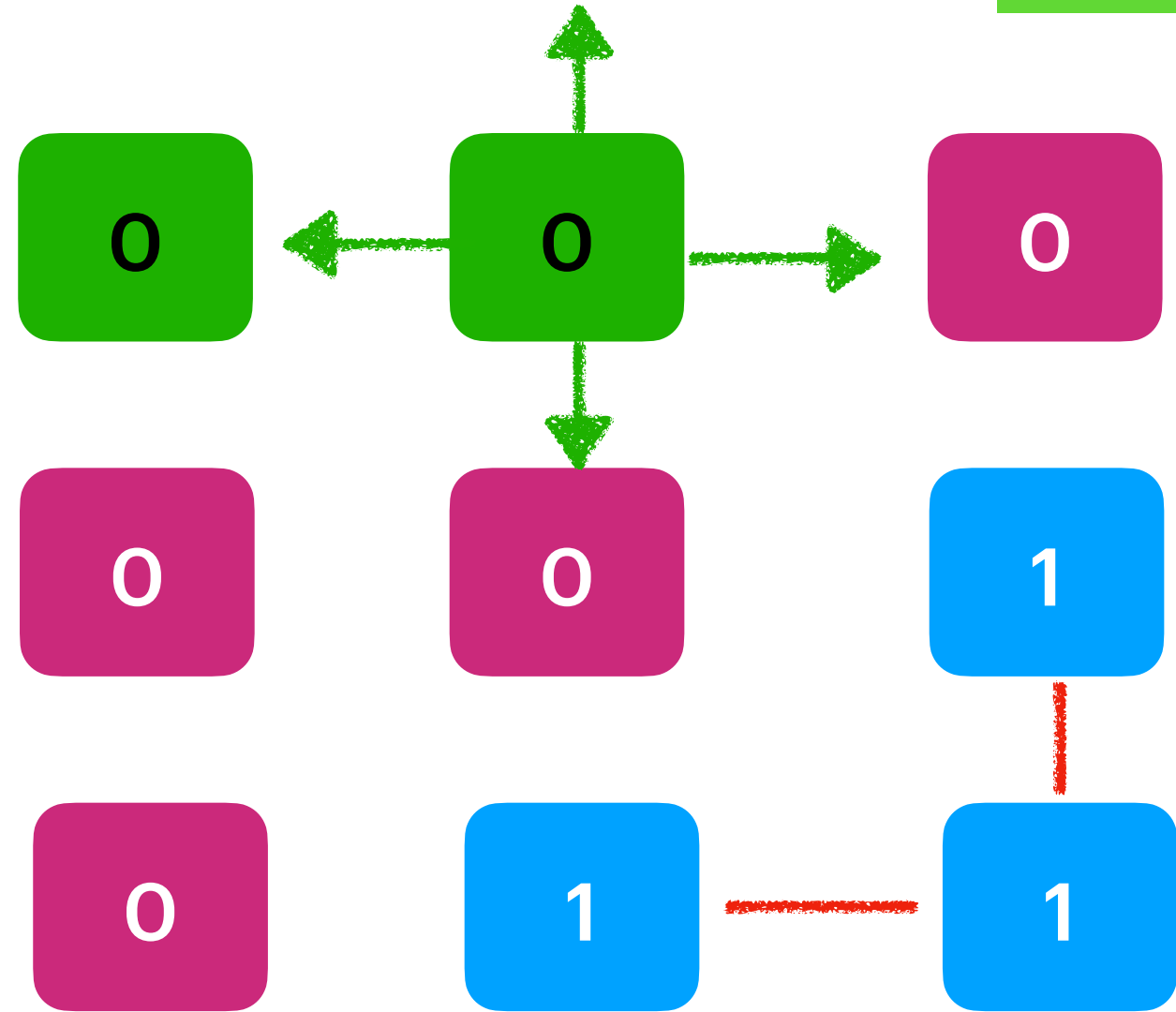
Queue[{1,1}]

Output



Size = 1

Input  
Queue[{1,1}]



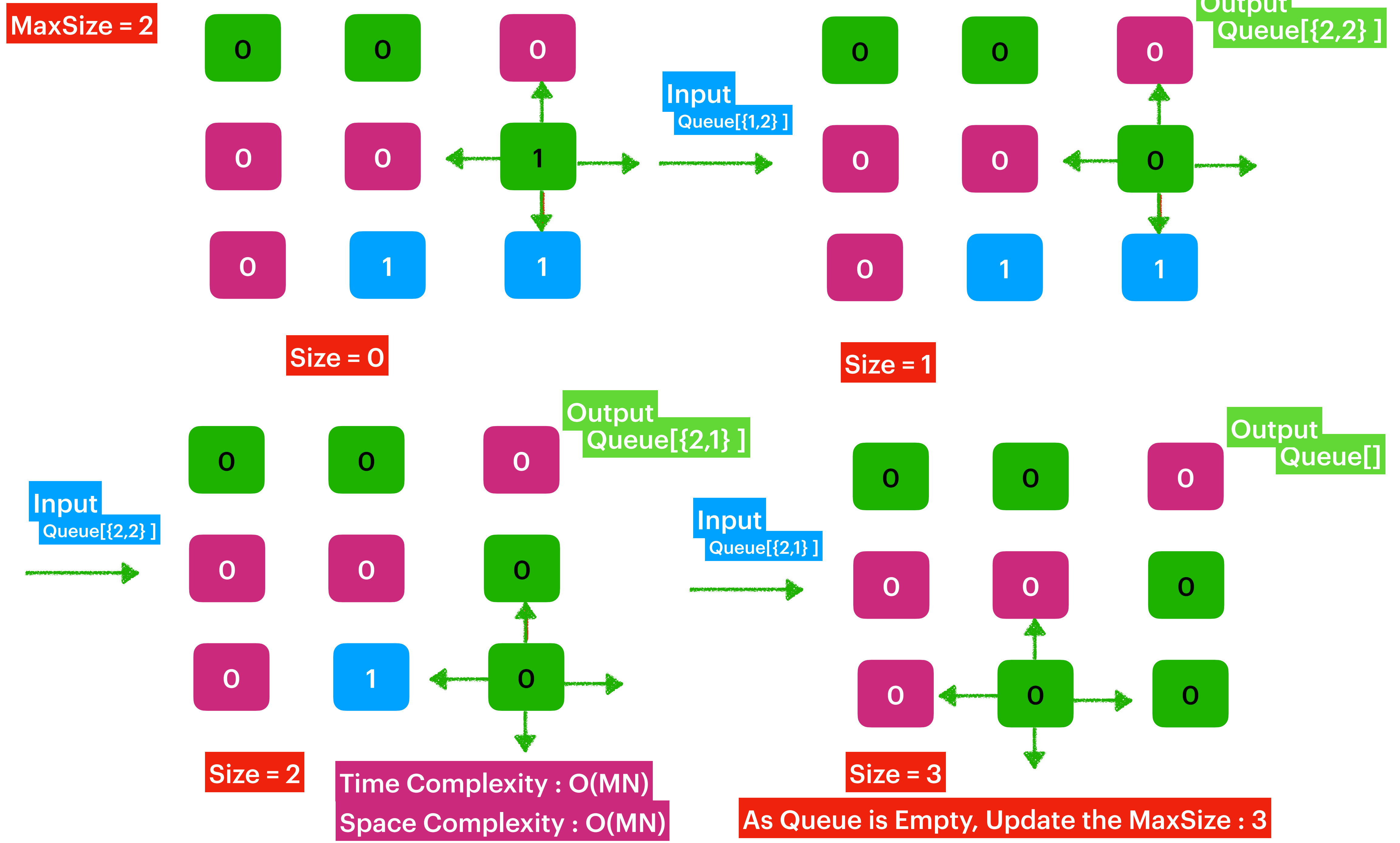
Size = 2

Queue[1]

Output

In the Given Input We found 2 Islands  
Length : 2 & Length :3  
So expected out is Max Length which 3

Now as the Queue Empty, update the MaxSize with CurrentSize:  
MaxSize = Math.max(MaxSize, size) = 2



## 827. Making A Large Island

Hard

👍 2464

💬 54

♡ Add to List

🔖 Share

You are given an  $n \times n$  binary matrix `grid`. You are allowed to change **at most one** `0` to be `1`.

Return *the size of the largest **island** in `grid` after applying this operation.*

An **island** is a 4-directionally connected group of `1` s.

### Example 1:

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

**Output:** 3

**Explanation:** Change one `0` to `1` and connect two `1`s, then we get an island with area = 3.

### Example 2:

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

**Output:** 4

**Explanation:** Change the `0` to `1` and make the island bigger, only one island with area = 4.

### Example 3:

**Input:** `grid = [[1,1],[1,1]]`

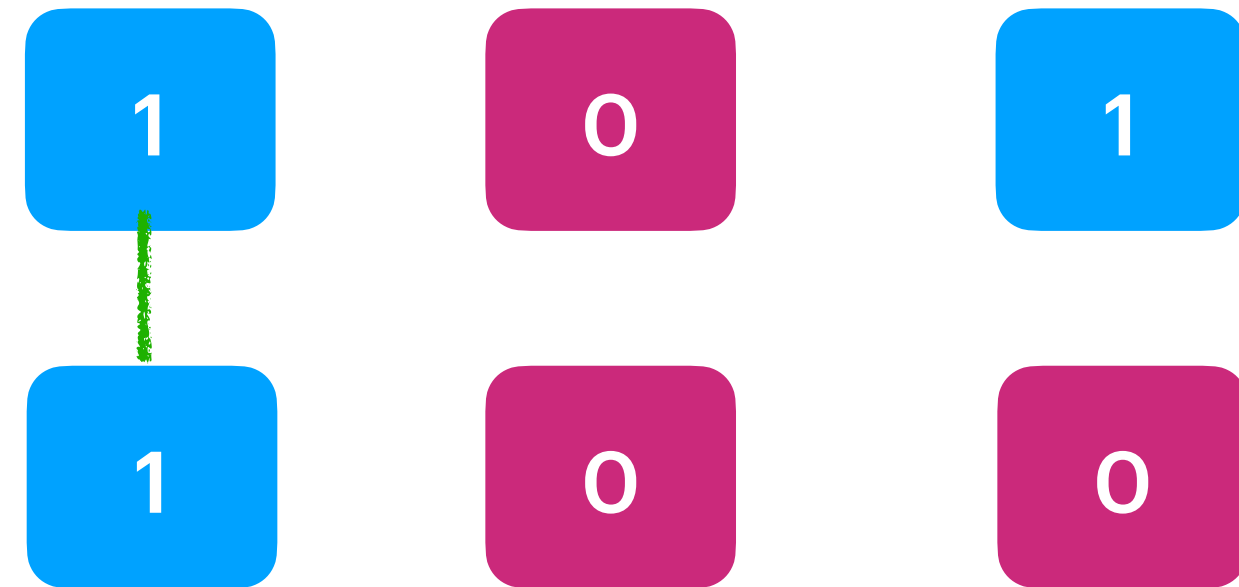
**Output:** 4

**Explanation:** Can't change any `0` to `1`, only one island with area = 4.

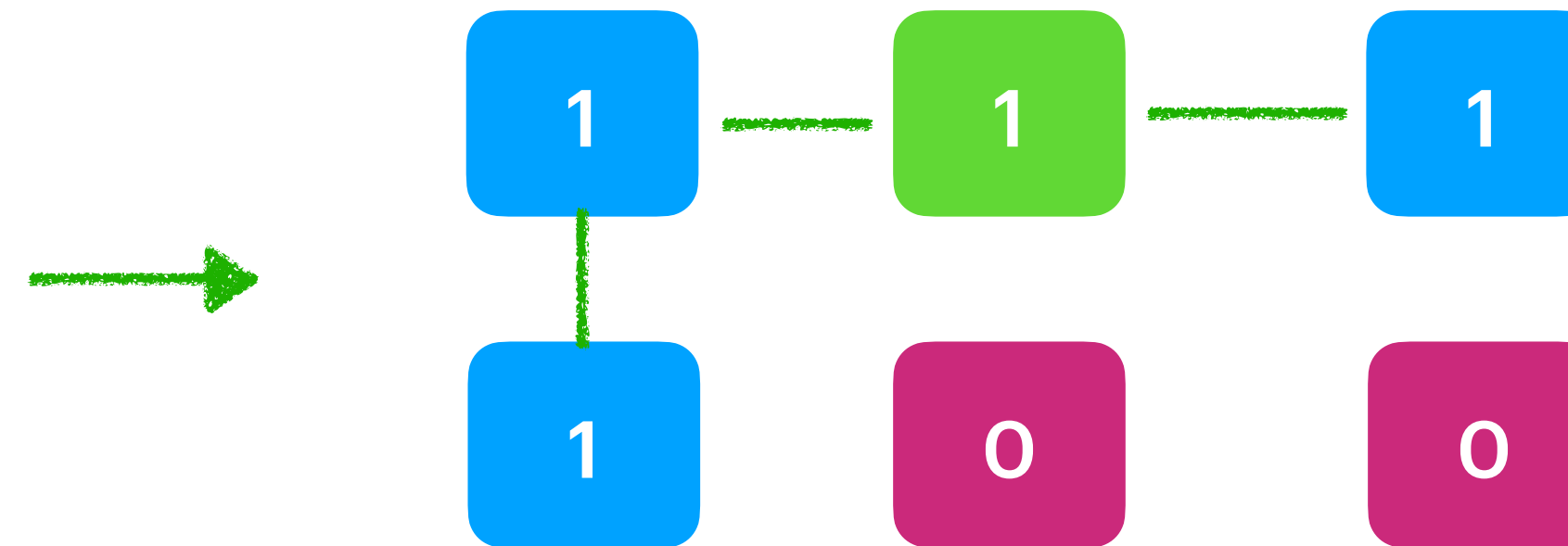
### Constraints:

- `n == grid.length`
- `n == grid[i].length`
- `1 <= n <= 500`
- `grid[i][j]` is either `0` or `1`.

Max Length = 2

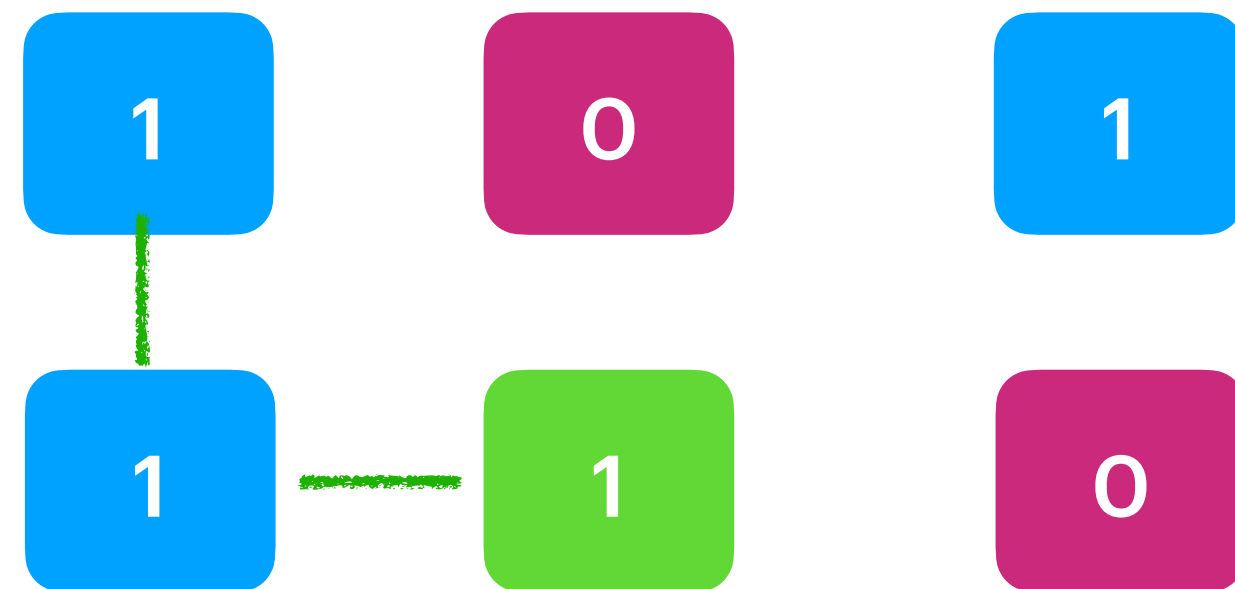


Making cell[0,1] to 1 make Island Length = 4

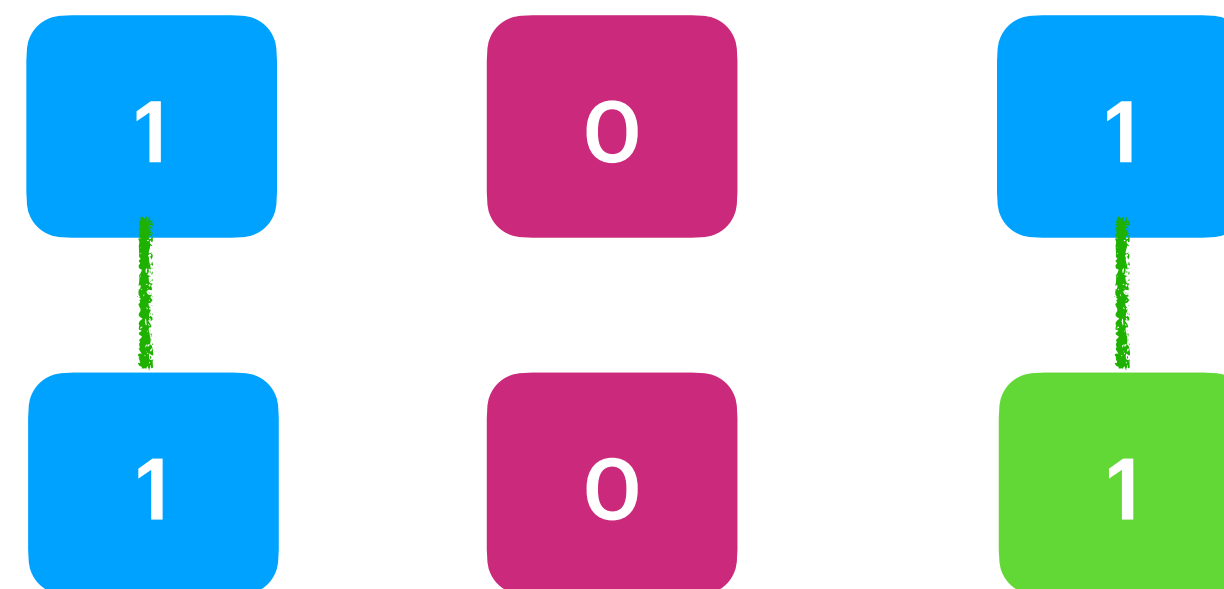


You are given an n x n binary matrix grid.  
You are allowed to change at most one 0 to be 1

Making cel1[1,1] to 1 make Island Length = 3



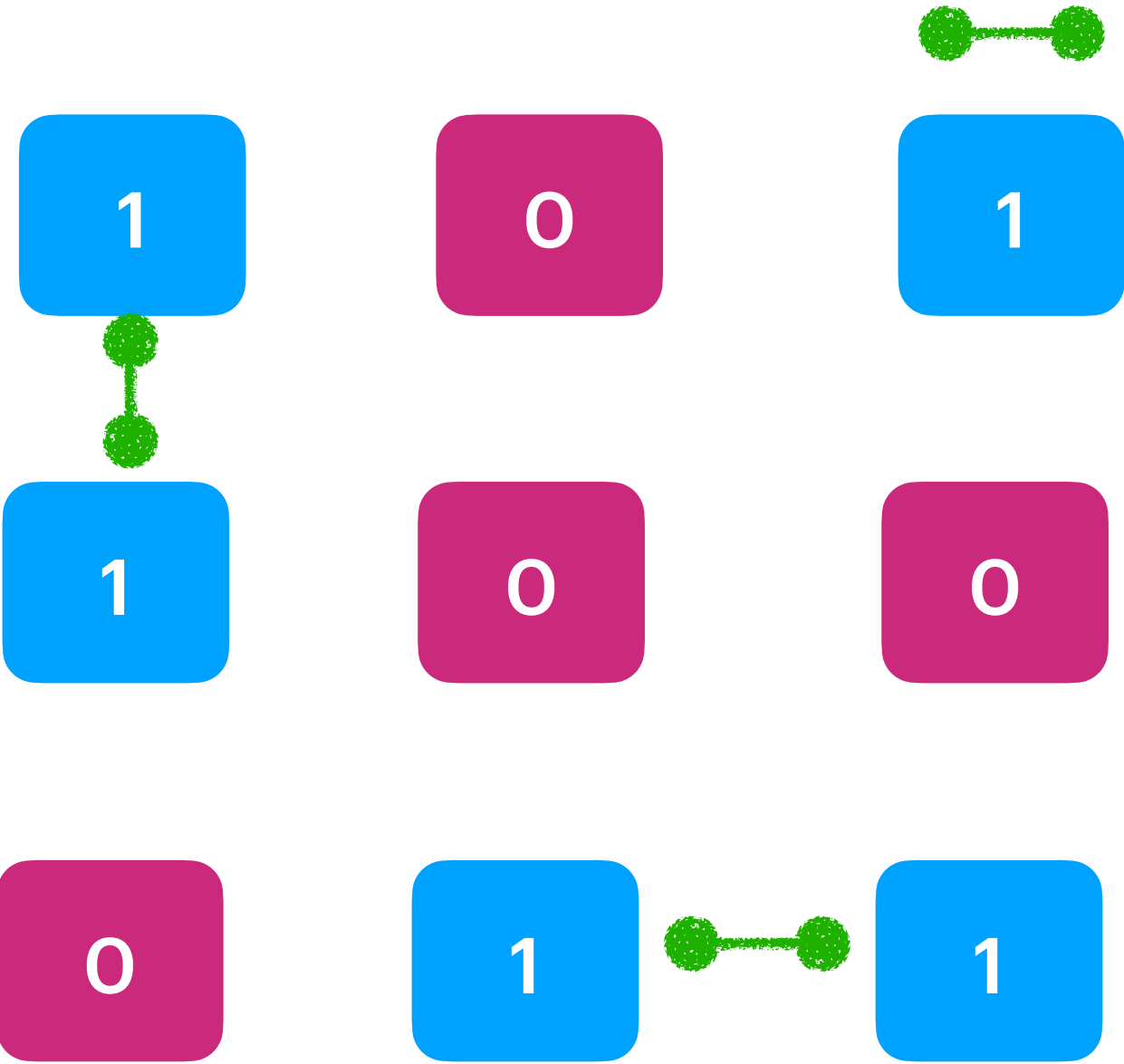
Making cel1[2,3] to 1 make Island Length = 2



Expected Output  
Max Length → 4

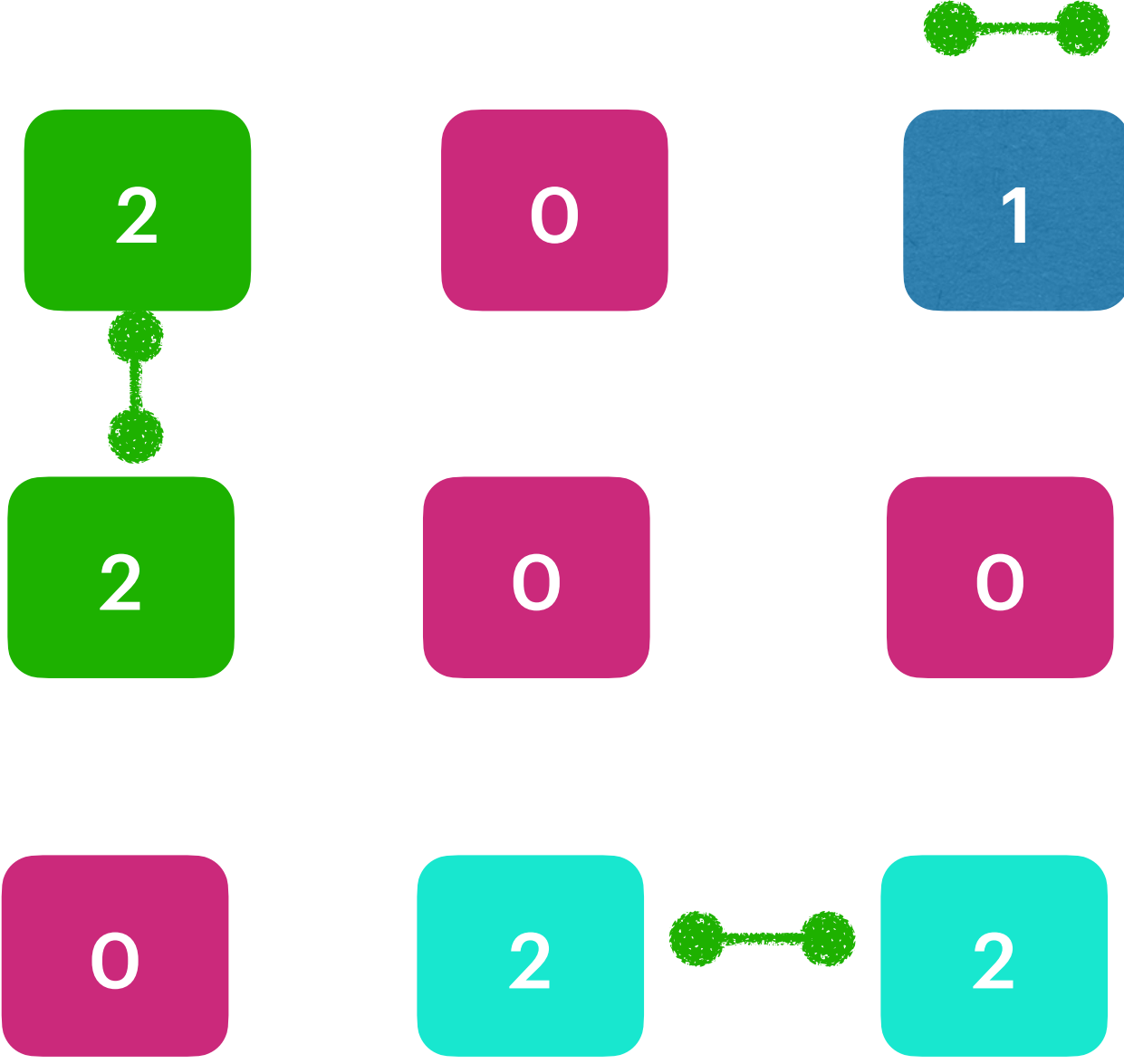


Input



Colour the Distinct Islands.  
Update its Length.

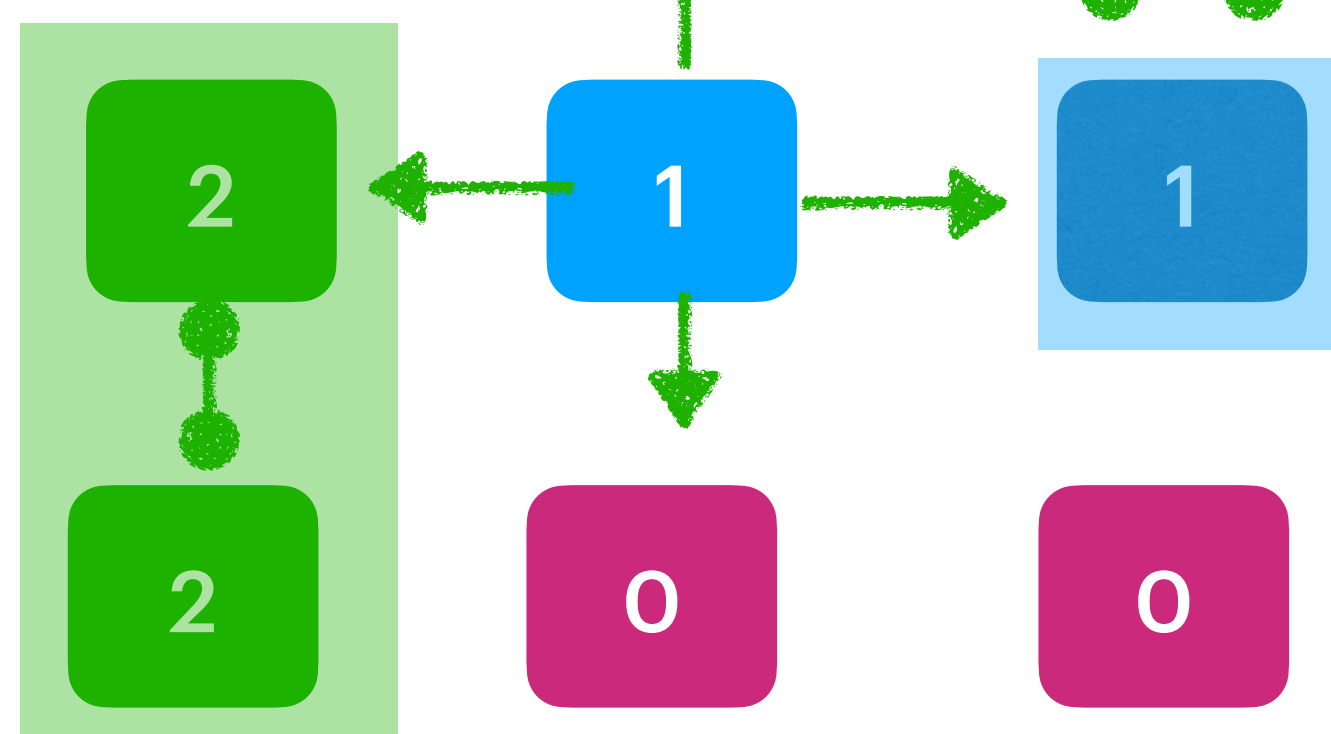
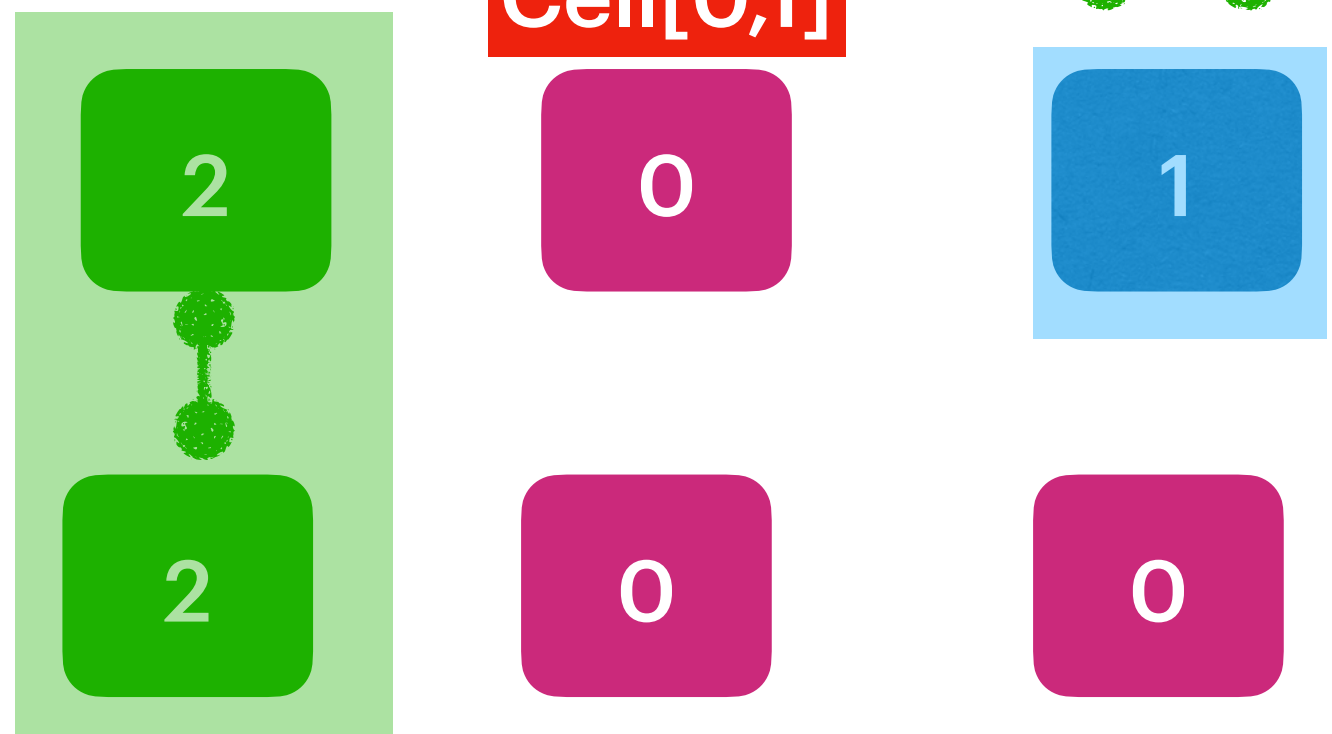
Output



We Found 3 distinct Islands

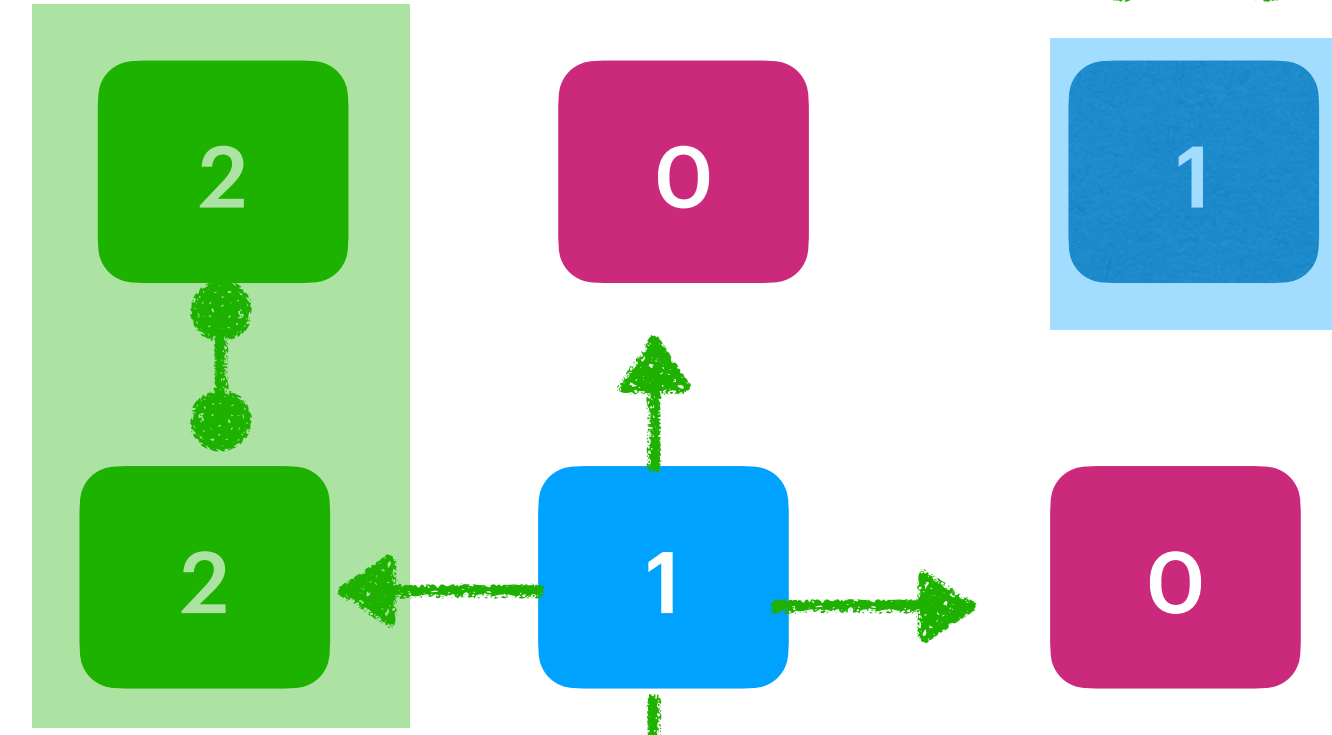
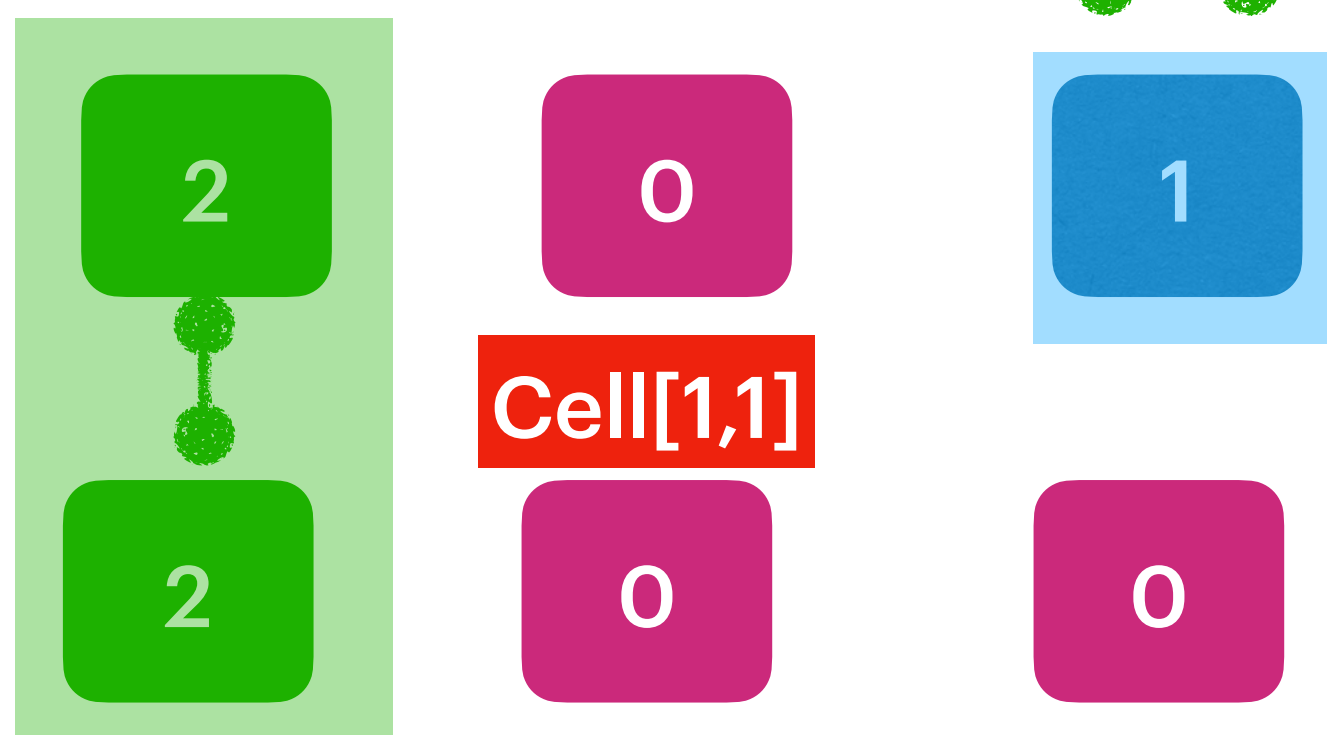
The Above use case is Equivalent to identify the size of Each Island :  $O(MN)$

Cell[0,1]



Making Cell [0,1] as Island Length = 4

Cell[1,1]



Making Cell [1,1] as Island Length = 5

