200. Number of Islands

Medium 14825 350 Add to List Share

Given an `m x 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:

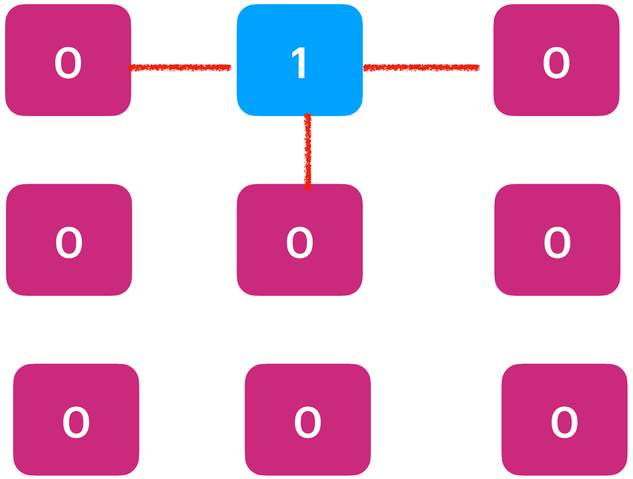
```
Input: grid = [
  ["1","1","1","1","0"],
  ["1","1","0","1","0"],
  ["1","1","0","0","0"],
  ["0","0","0","0","0"]
]
Output: 1
```

Example 2:

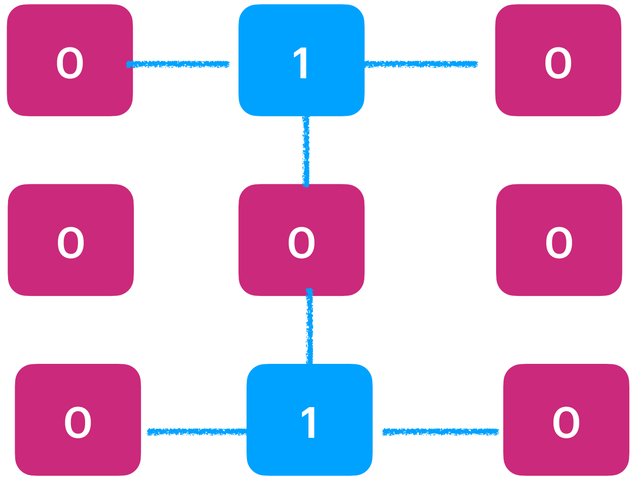
```
Input: grid = [
  ["1","1","0","0","0"],
  ["1","1","0","0","0"],
  ["0","0","1","0","0"],
  ["0","0","0","1","1"]
]
Output: 3
```

Constraints:

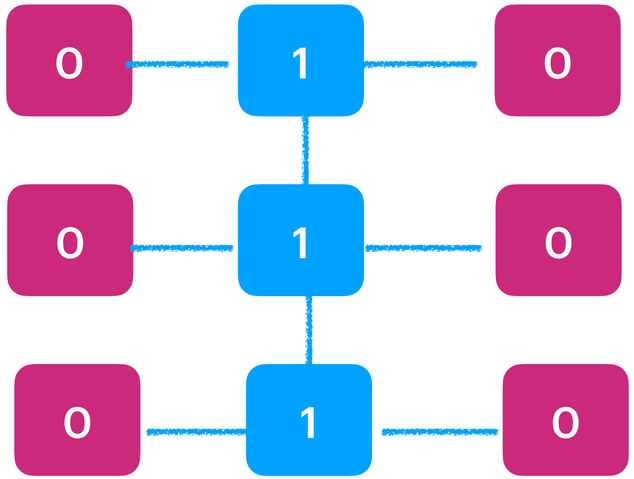
- `m == grid.length`
- `n == grid[i].length`
- `1 <= m, n <= 300`
- `grid[i][j]` is `'0'` or `'1'`.



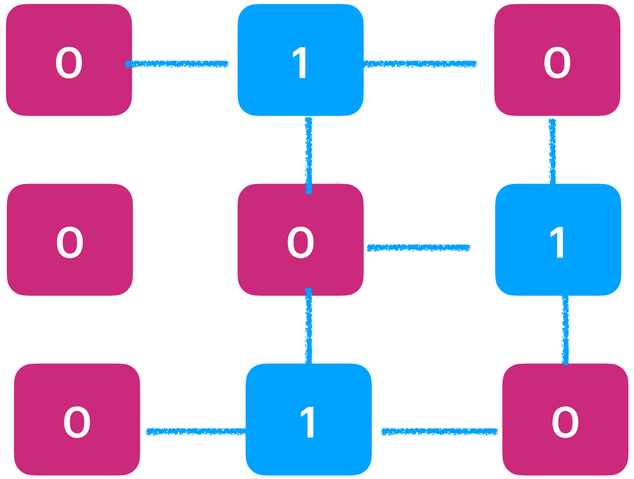
No.of Islands = 1



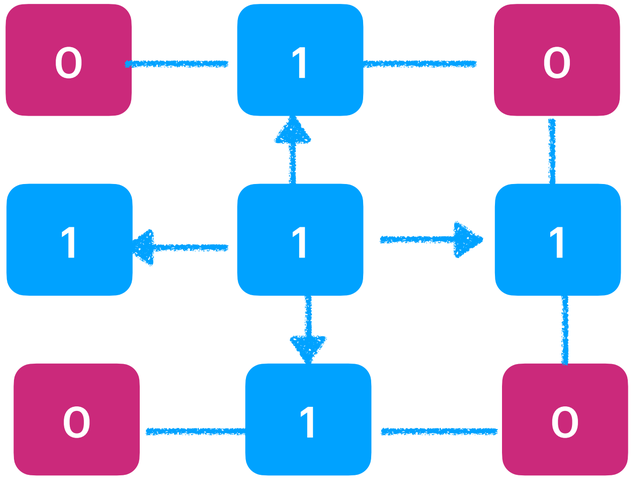
No.of Islands = 2



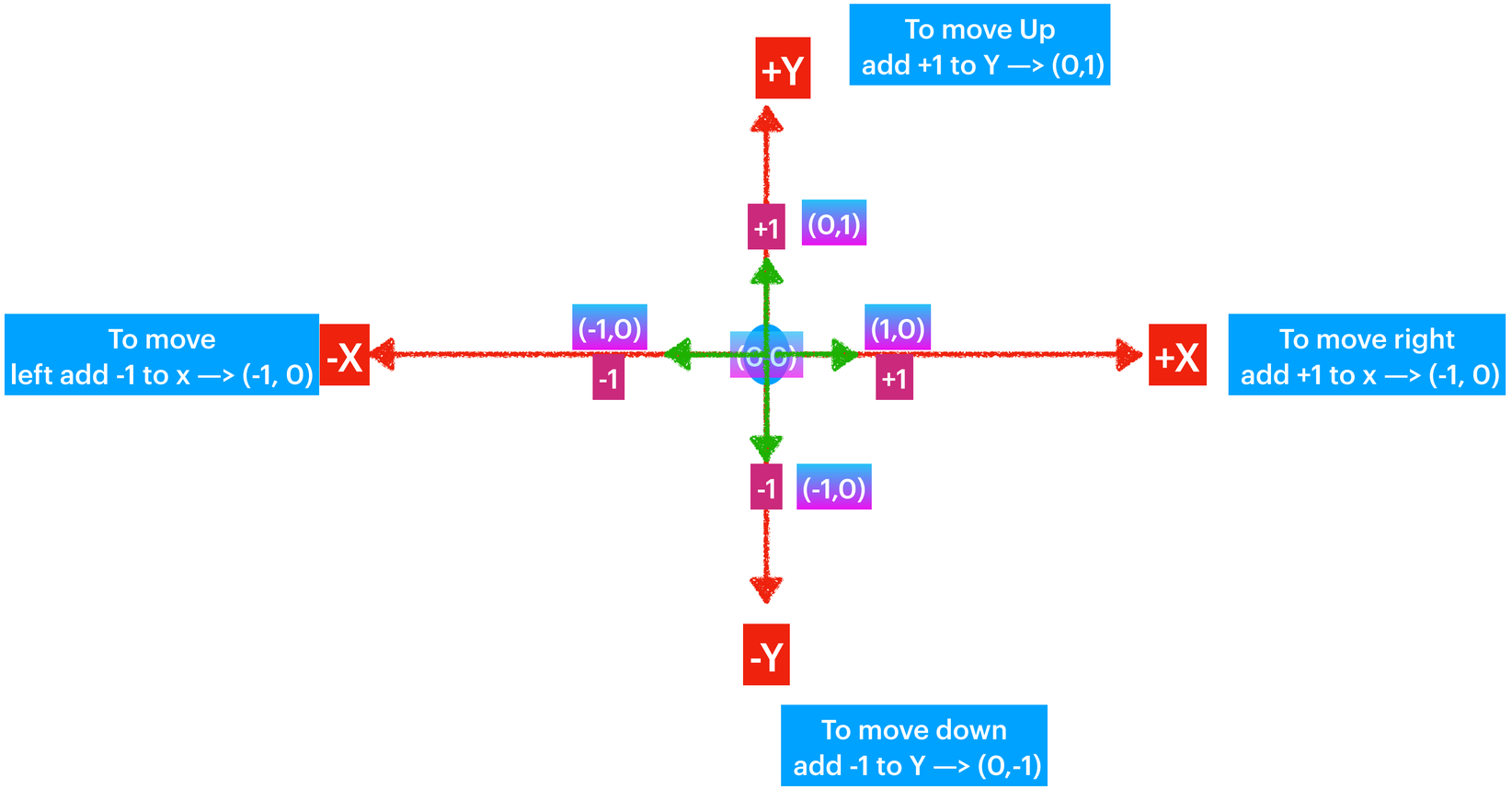
No.of Islands = 1



No.of Islands = 3



No.of Islands = 1


To move Up
add +1 to Y —> (0,1)
+Y
+1
(0,1)
To move
left add -1 to x —> (-1, 0)
-X
(-1,0)
-1
(C,C)
(1,0)
+1
+X
To move right
add +1 to x —> (-1, 0)
-1
(-1,0)
-Y
To move down
add -1 to Y —> (0,-1)

islandCount = 0

(0,0)	(0,1)	(0,2)
0	1	0
(1,0)	(1,1)	(1,2)
1	1	1
(2,0)	(2,1)	(2,2)
0	1	0

(0,0) Position value is
0 its not island so move to (0,1)

islandCount = 1

(0,0)	(0,1)	(0,2)
0	1	0
(1,0)	(1,1)	(1,2)
1	1	1
(2,0)	(2,1)	(2,2)
0	1	0

(0,1) Position has value
1 So we need unity the connected Islands using BFS/DFS

Algorithm :

1. Traverse the Matrix,When ever we find cell value as 1. Increment islandCounter,
2. Initiate the BFS/DFS travel for the Cell.
3. Then after repeat the above two steps for the next cells.

Refer the BFS traversal Below which united all the connected islands.
We used 0 as a flag. [Updated connected islands value 1 to 0]

Output of
BFS Traversal for the Cell(0,1)

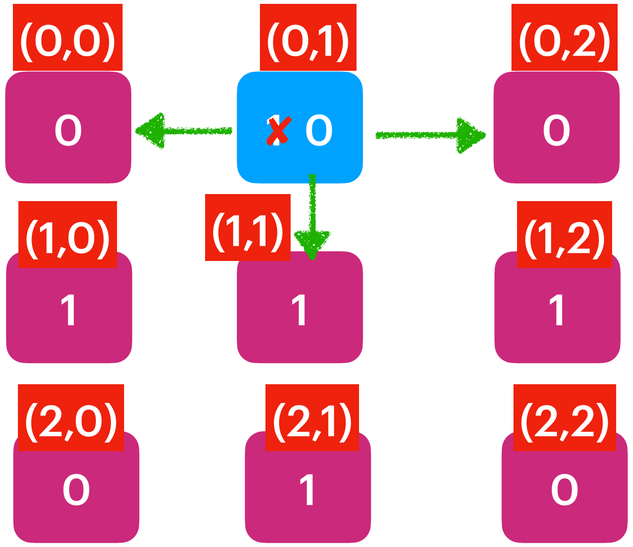
(0,0)	(0,1)	(0,2)
0	0	0
(1,0)	(1,1)	(1,2)
0	0	0
(2,0)	(2,1)	(2,2)
0	0	0

Time Complexity : O(MN)
Each cell can be visited at max twice
.
Space Complexity : O(MN) becaue of BFS Queue

Input

Queue : [(0,1)]

BFS Traversal



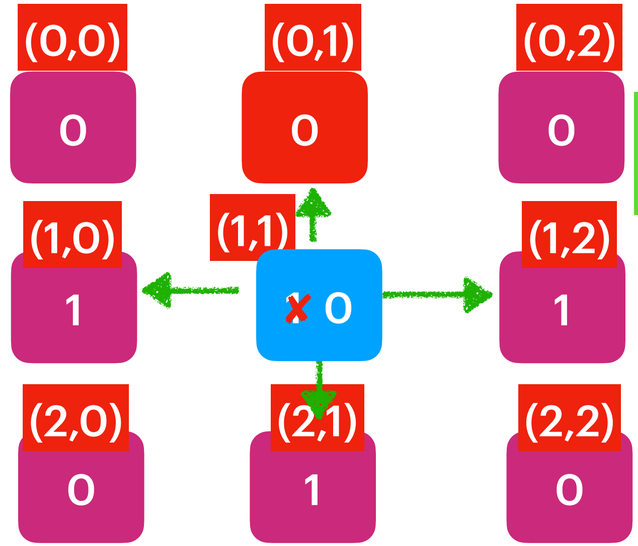
Output

Queue : [(1,1)]

1. As the queue is not empty:
2. Pop the cell from Queue:
Cell (0,1)
3. Make Cell(0,1) value as 0
—> Take Four Directions from Cell(0,1)
We identified that bottom direction
Cell(1,1) has value ‘1’
—-> Add (1,1) to Queue.

Output

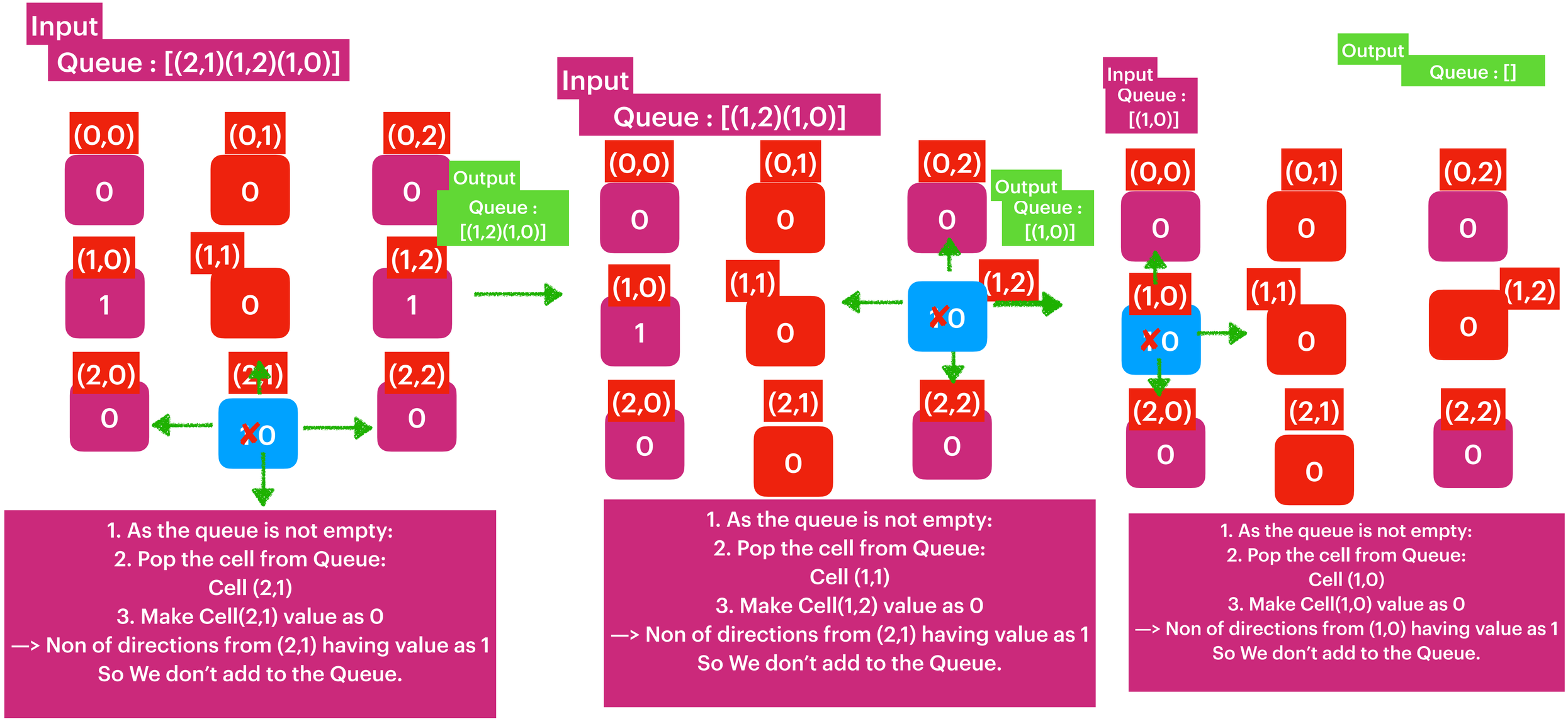
Queue : [(1,1)]



Output

Queue : [(2,1)(1,2)(1,0)]

1. As the queue is not empty:
2. Pop the cell from Queue:
Cell (1,1)
3. Make Cell(1,1) value as 0
—> Take Four Directions from Cell(1,1)
We identified that bottom, left and right cells has 1 Add them to Queue
(2,1)(1,2)(1,0)



As Queue is Empty We stop BFS Traversal

695. Max Area of Island

Medium 6529 154 Add to List Share

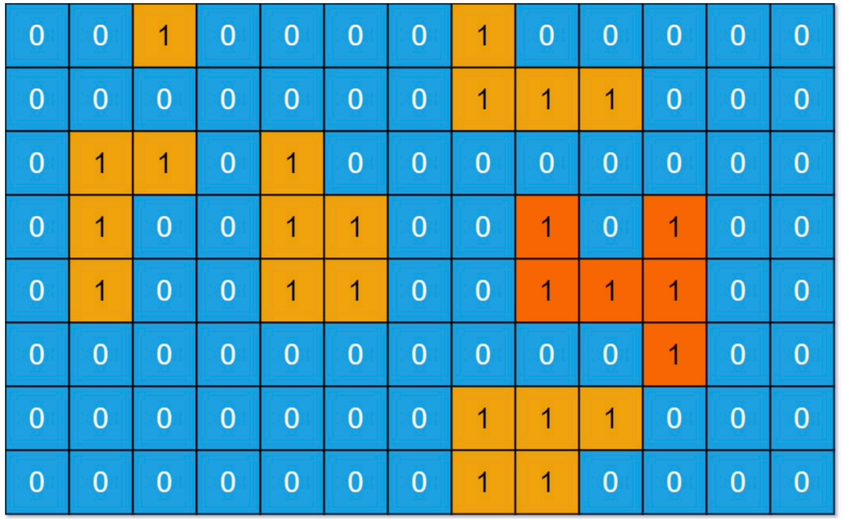
You are given an `m x n` binary matrix `grid`. An island is a group of `1`'s (representing land) connected **4-directionally** (horizontal or vertical.) You may assume all four edges of the grid are surrounded by water.

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`.

For Exercise

Example 1:




```
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],
[0,0,0,0,0,0,0,0,0,0,1,0,0],[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`.