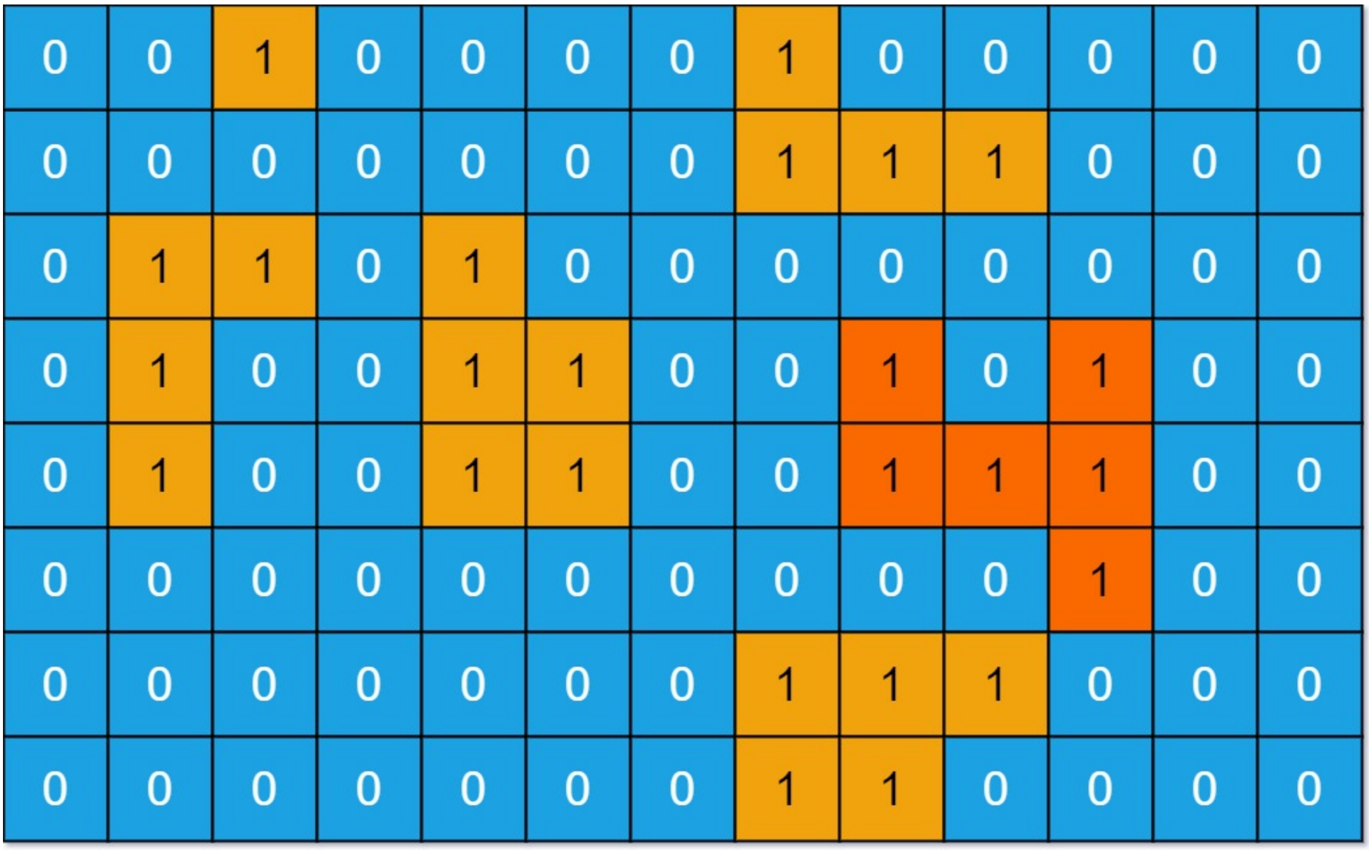
695. Max Area of Island

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

```
class Solution:
    def maxAreaOfIsland(self, grid: List[List[int]]) -> int:
        visited = [[False]*len(grid[0]) for _ in range(len(grid))]
        ans = 0
        for i in range(len(grid)):
            for j in range(len(grid[0])):
                if grid[i][j]==1 and visited[i][j]==False:
                    res = self.calculateArea(grid,visited,i,j)
                    ans = max(res,ans)
        return ans



    def calculateArea(self,grid,visited,i,j):
        if i<0 or j<0 or i>=len(grid) or j>=len(grid[0]) or visited[i][j]==True or grid[i][j]==0:
            return 0

        visited[i][j] = True

        up = self.calculateArea(grid,visited,i-1,j)
        right = self.calculateArea(grid,visited,i,j+1)
        down = self.calculateArea(grid,visited,i+1,j)
        left = self.calculateArea(grid,visited,i,j-1)

        return (up+right+down+left+grid[i][j])
```