

1254. Number of Closed Islands

🕒 Created	@November 29, 2022 3:25 PM
📌 Difficulty	Medium
🔗 LC Url	https://leetcode.com/problems/number-of-closed-islands/
📌 Importance	
🏷️ Tag	DFS Island
🔗 Reference	

Given a 2D `grid` consists of `0`s (land) and `1`s (water). An *island* is a maximal 4-directionally connected group of `0`s and a *closed island* is an island **totally** (all left, top, right, bottom) surrounded by `1`s.

Return the number of *closed islands*.

Example 1:

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

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

Output: 2

Explanation:

Islands in gray are closed because they are completely surrounded by water (group of 1s).

Example 2:

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

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

Example 3:

Input: grid = [[1,1,1,1,1,1,1],
[1,0,0,0,0,0,1],
[1,0,1,1,1,0,1],
[1,0,1,0,1,0,1],
[1,0,1,1,1,0,1],
[1,0,0,0,0,0,1],
[1,1,1,1,1,1,1]]
Output: 2

Constraints:

- `1 <= grid.length, grid[0].length <= 100`
- `0 <= grid[i][j] <= 1`

Solution

```

class Solution:
    directions = [(0, 1), (0, -1), (1, 0), (-1, 0)]

    def closedIsland(self, grid: List[List[int]]) -> int:
        res = 0
        m = len(grid)
        if m == 0:
            return res
        n = len(grid[0])

        for i in range(m):
            # 把靠左边的岛屿淹掉
            self.dfs(grid, i, 0)
            # 把靠右边的岛屿淹掉
            self.dfs(grid, i, n - 1)

        for j in range(n):
            # 把靠上边的岛屿淹掉
            self.dfs(grid, 0, j)
            # 把靠下边的岛屿淹掉
            self.dfs(grid, m - 1, j)

        # 遍历 grid, 剩下的岛屿都是封闭岛屿
        for i in range(m):
            for j in range(n):
                if grid[i][j] == 0:
                    res += 1
                    self.dfs(grid, i, j)

        return res

    def dfs(self, grid, i, j):
        """
        从 (i, j) 开始, 将与之相邻的陆地都变成海水
        """
        if not self.is_valid(grid, i, j):
            return

        # 已经是海水了
        if grid[i][j] == 1:
            return

        # 将 (i, j) 变成海水
        grid[i][j] = 1

        # 淹没上下左右的陆地
        for direction in self.directions:
            cur_i, cur_j = i + direction[0], j + direction[1]
            self.dfs(grid, cur_i, cur_j)

    def is_valid(self, grid, i, j):
        """

```

```
Check whether (i, j) is in the domain
"""
m, n = len(grid), len(grid[0])
if 0 <= i < m and 0 <= j < n:
    return True
return False
```

1020. Number of Enclaves