1020. Number of Enclaves

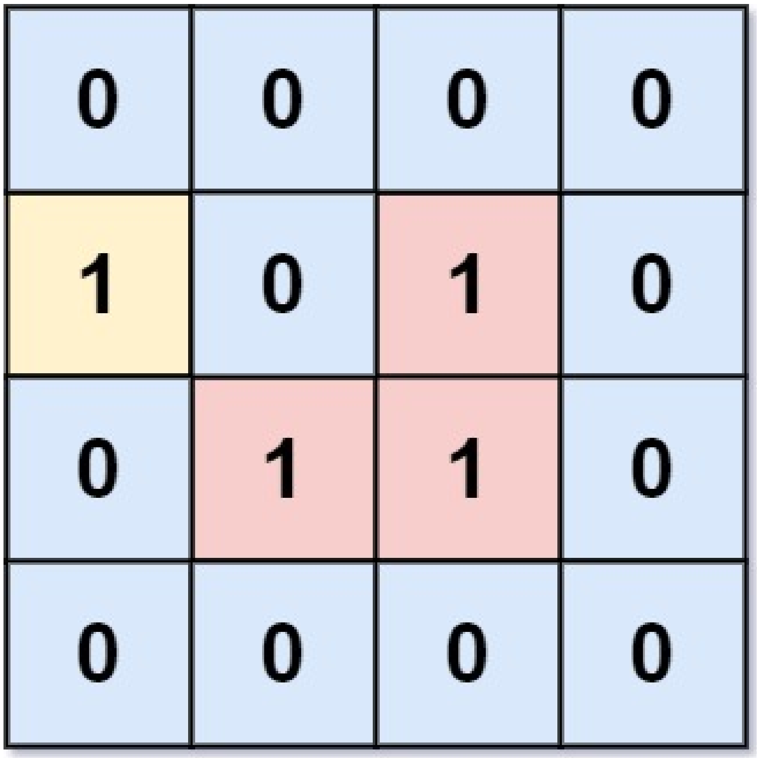
Created	@November 29, 2022 3:49 PM
Difficulty	Medium
LC Url	https://leetcode.com/problems/number-of-enclaves/
Importance	
Tag	DFS Island
Reference	https://labuladong.github.io/algo/4/31/107/

You are given an `m x n` binary matrix `grid`, where `0` represents a sea cell and `1` represents a land cell.

A **move** consists of walking from one land cell to another adjacent (**4-directionally**) land cell or walking off the boundary of the `grid`.

Return *the number of land cells in* `grid` *for which we cannot walk off the boundary of the grid in any number of* ***moves***.

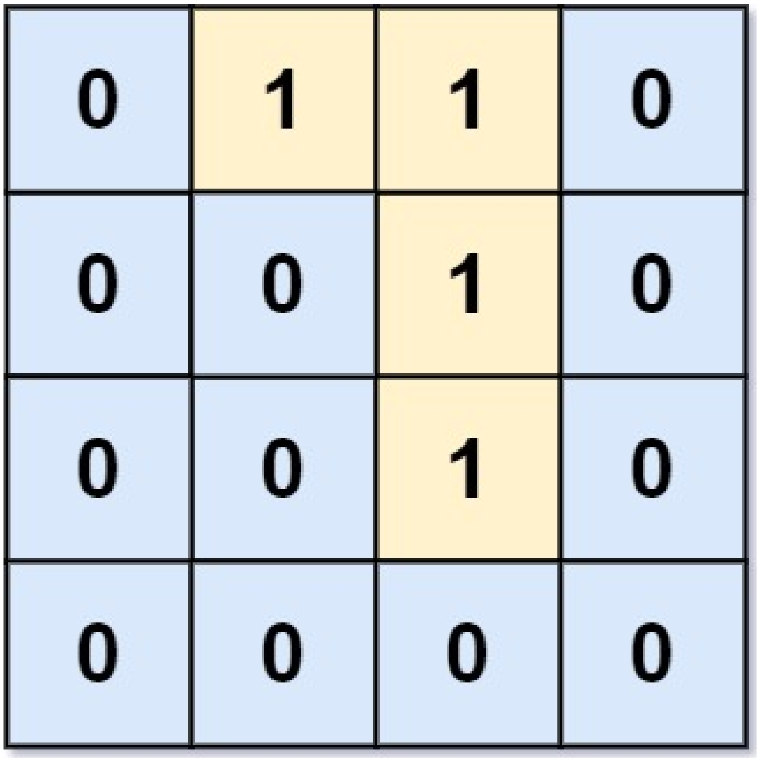
Example 1:



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

```
Input: grid = [[0,0,0,0],[1,0,1,0],[0,1,1,0],[0,0,0,0]]
Output: 3
Explanation: There are three 1s that are enclosed by 0s, and one 1 that is not enclosed be
cause its on the boundary.
```

Example 2:



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

```
Input: grid = [[0,1,1,0],[0,0,1,0],[0,0,1,0],[0,0,0,0]]
Output: 0
Explanation: All 1s are either on the boundary or can reach the boundary.
```

Constraints:

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

Solution

这道岛屿题目的解法稍微改改就可以解决力扣第 1020 题「飞地的数量」，这题不让你求封闭岛屿的数量，而是求封闭岛屿的面积总和。

其实思路都是一样的，先把靠边的陆地淹掉，然后去数剩下的陆地数量就行了，注意第 1020 题中 `1` 代表陆地，`0` 代表海水：

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

    def numEnclaves(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)

        # 数一数剩下的陆地
        for i in range(m):
            for j in range(n):
                if grid[i][j] == 1:
                    res += 1

        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] == 0:
            return

        # 将 (i, j) 变成海水
        grid[i][j] = 0

        # 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
```

1254. Number of Closed Islands