Problem 1254: Number of Closed Islands

Problem Information

Difficulty: Medium
Acceptance Rate: 66.94%
Paid Only: No
Tags: Array, Depth-First Search, Breadth-First Search, Union Find, Matrix

Problem Description

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

Return the number of _closed islands_.

Example 1:

![](https://assets.leetcode.com/uploads/2019/10/31/sample_3_1610.png)

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:

![](https://assets.leetcode.com/uploads/2019/10/31/sample_4_1610.png)

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`

Code Snippets

C++:

```
class Solution {
public:
int closedIsland(vector<vector<int>>& grid) {

}
};
```

Java:

```
class Solution {
public int closedIsland(int[][] grid) {

}
}
```

Python3:

```
class Solution:
def closedIsland(self, grid: List[List[int]]) -> int:
```