694. Number of Distinct Islands

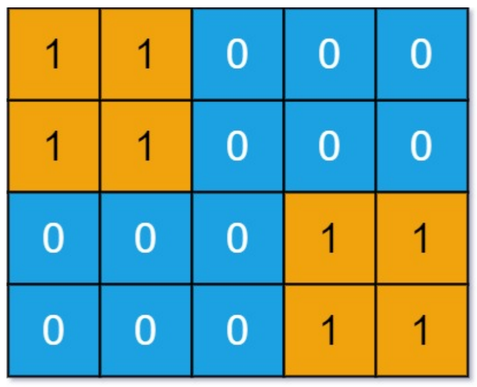
Medium 1527 87 Add to List Share

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.

An island is considered to be the same as another if and only if one island can be translated (and not rotated or reflected) to equal the other.

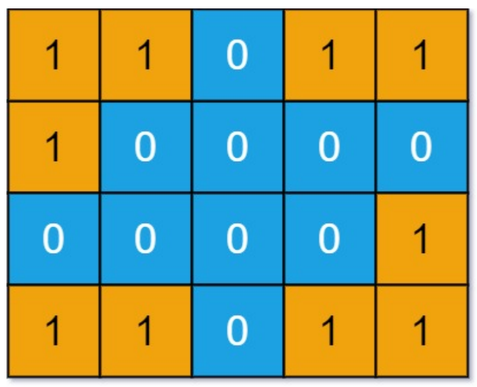
Return *the number of **distinct** islands*.

Example 1:




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

Example 2:




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

Constraints:

- `m == grid.length`
- `n == grid[i].length`
- `1 <= m, n <= 50`
- `grid[i][j]` is either `0` or `1`.

Accepted 107,466 | Submissions 181,835

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Amazon | 7 Facebook | 5 Bloomberg | 3 Uber | 2 ByteDance | 2

Related Topics

Hash Table Depth-First Search Breadth-First Search Union Find Hash Function

Similar Questions

Number of Islands	Medium
Number of Distinct Islands II	Hard
Count Sub Islands	Medium

Java Autocomplete

```java
class Solution {

    private int[][] grid;
    private boolean[][] seen;
    private Set<Pair<Integer, Integer>> currentIsland;
    private int currRowOrigin;
    private int currColOrigin;

    private void dfs(int row, int col) {
        if (row < 0 || row >= grid.length || col < 0 || col >= grid[0].length) return;
        if (grid[row][col] == 0 || seen[row][col]) return;
        seen[row][col] = true;
        currentIsland.add(new Pair<>(row - currRowOrigin, col - currColOrigin));
        dfs(row + 1, col);
        dfs(row - 1, col);
        dfs(row, col + 1);
        dfs(row, col - 1);
    }

    public int numDistinctIslands(int[][] grid) {
        this.grid = grid;
        this.seen = new boolean[grid.length][grid[0].length];
        Set<Set<Pair<Integer, Integer>>> islands = new HashSet<>();
        for (int row = 0; row < grid.length; row++) {
            for (int col = 0; col < grid[0].length; col++) {
                this.currentIsland = new HashSet<>();
                this.currRowOrigin = row;
                this.currColOrigin = col;
                dfs(row, col);
                if (!currentIsland.isEmpty()) islands.add(currentIsland);
            }
        }
        return islands.size();
    }
}
```