200. Number of Islands

Created	@November 28, 2022 11:27 PM
Difficulty	Medium
LC Url	https://leetcode.com/problems/number-of-islands/
Importance	
Tag	DFS Island
Reference	https://labuladong.github.io/algo/4/31/107/

Given an `m x n` 2D binary grid `grid` which represents a map of `'1'`s (land) and `'0'`s (water), return *the number of islands*.

An **island** is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

Example 1:

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

Example 2:

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

Constraints:

- `m == grid.length`
- `n == grid[i].length`
- `1 <= m, n <= 300`
- `grid[i][j]` is `'0'` or `'1'`.

Solution

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

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

        # 遍历grid
        for i in range(m):
            for j in range(n):
                if grid[i][j] == '1':
                    # 每发现一个岛屿，岛屿总数加1
                    res += 1
                    # 用DFS将岛屿标记
                    self.dfs(grid, i, j)

        return res

    def dfs(self, grid, i, j):
        m = len(grid)
        n = len(grid[0])

        # 判断是否在区域内
        if i < 0 or j < 0 or i >= m or j >= n:
            return

        # 如果这个格子不是岛屿，直接返回
        if grid[i][j] != '1':
            return

        # 将格子标记为“已遍历过”
        grid[i][j] = '2'
        # 访问上下左右四个相邻节点
        for direction in self.directions:
```

```
            cur_i, cur_j = i + direction[0], j + direction[1]
            self.dfs(grid, cur_i, cur_j)
```

力扣

https://leetcode.cn/problems/number-of-islands/solution/dao-yu-lei-wen-ti-de-tong-yong-jie-fa-dfs-bian-li-/