



链滴

# ARTS 011

作者: [lucianolixin](#)

原文链接: <https://ld246.com/article/1573525782680>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



ARTS 是由左耳朵耗子陈皓在极客时间专栏《左耳听风》中发起的一个每周学习打卡计划。

Algorithm: 至少做一个 LeetCode 的算法题。主要为了编程训练和学习。

Review : 阅读并点评至少一篇英文技术文章。主要为了学习英文, 如果你英文不行, 很难成为技术手。

Tip: 学习至少一个技术技巧。主要是为了总结和归纳你日常工作中所遇到的知识点。

Share: 分享一篇有观点和思考的技术文章。主要为了输出你的影响力, 能够输出你的价值观。

## Algorithm

给定一个由 '1' (陆地) 和 '0' (水) 组成的二维网格, 计算岛屿的数量。一个岛被水包围, 并且它通过水平方向或垂直方向上相邻的陆地连接而成的。你可以假设网格的四个边均被水包围。

示例

**输入:**

11110

11010

11000

00000

**输出:** 1

来源: 力扣 (LeetCode)

链接: <https://leetcode-cn.com/problems/number-of-islands>

```
func numIslands(grid [][]byte) int {  
  
    count := 0  
    for y, rows := range grid {  
        for x, element := range rows {  
            if element == 49 {  
                count++  
                bfs(y, x, grid)  
            }  
        }  
    }  
    return count  
}
```

```
func bfs(y int, x int, grid [][]byte) {  
  
    if y+1 != len(grid) {  
        if grid[y+1][x] == 49 {  
            grid[y+1][x] = 48  
            bfs(y+1,x, grid)  
        }  
    }  
  
    if y-1 >= 0 {  
        if grid[y-1][x] == 49 {  
            grid[y-1][x] = 48  
            bfs(y-1,x, grid)  
        }  
    }  
  
    if x+1 != len(grid[y]) {  
        if grid[y][x+1] == 49{  
            grid[y][x+1] = 48  
            bfs(y,x+1, grid)  
        }  
    }  
  
    if x-1 >= 0 {  
        if grid[y][x-1] == 49{  
            grid[y][x-1] = 48  
            bfs(y,x-1, grid)  
        }  
    }  
  
    return  
}
```