

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，并且它是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 1:

输入:

```
11110
11010
11000
00000
```

输出: 1

#### 示例 2:

输入:

```
11000
11000
00100
00011
```

输出: 3

思路，用遍历去标记，先找岛，找到岛后，把他的土地全给消除，在找其他的岛屿，遍历的思路用迷宫探索的，四周遍历即可（坑比，01问题给用字符串传入）

```
class Solution {
public:
    void dfs(vector<vector<char>>& grid,int i,int j,int &n,int &m){
        if(i>0&&grid[i-1][j]=='1'){
            grid[i-1][j]='0';
            dfs(grid,i-1,j,n,m);
        }
        if(i<n-1&&grid[i+1][j]=='1'){
            grid[i+1][j]='0';
            dfs(grid,i+1,j,n,m);
        }
        if(j>0&&grid[i][j-1]=='1'){
            grid[i][j-1]='0';
            dfs(grid,i,j-1,n,m);
        }
        if(j<m-1&&grid[i][j+1]=='1'){
            grid[i][j+1]='0';
            dfs(grid,i,j+1,n,m);
        }
    }
    int numIslands(vector<vector<char>>& grid) {
```

```
int num=0;
int n=grid.size();
if(n==0){
    return 0;
}
int m=grid[0].size();
for(int i=0;i<n;i++){
    for(int j=0;j<m;j++){
        if(grid[i][j]=='1'){
            num++;
            grid[i][j]='0';
            dfs(grid,i,j,n,m);
        }
    }
}
return num;
}
};
```