

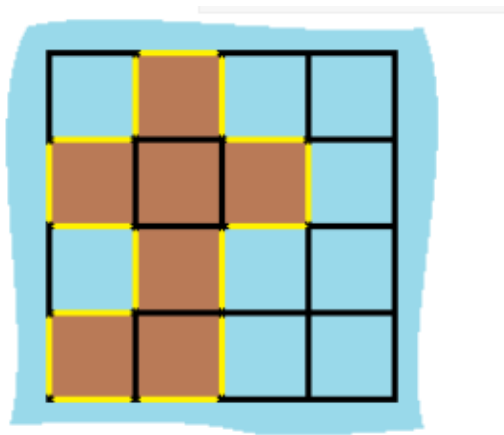
岛屿的周长

给定一个 $\text{row} \times \text{col}$ 的二维网格地图 grid ，其中： $\text{grid}[i][j] = 1$ 表示陆地， $\text{grid}[i][j] = 0$ 表示水域。

网格中的格子 **水平**和**垂直** 方向相连（对角线方向不相连）。整个网格被水完全包围，但其中恰好有一个岛屿（或者说，一个或多个表示陆地的格子相连组成的岛屿）。

岛屿中没有“湖”（“湖”指水域在岛屿内部且不和岛屿周围的水相连）。格子是边长为 1 的正方形。网格为长方形，且宽度和高度均不超过 100 。计算这个岛屿的周长。

示例 1：



输入： $\text{grid} = [[0,1,0,0],[1,1,1,0],[0,1,0,0],[1,1,0,0]]$

输出：16

解释：它的周长是上面图片中的 16 个黄色的边

示例 2：

输入： $\text{grid} = [[1]]$

输出：4

示例 3：

输入： $\text{grid} = [[1,0]]$

输出：4

```
class Solution {
public:
    int circle=0;
    int row,col;
    int dx[4] = {1, 0, -1, 0}, dy[4] = {0, 1, 0, -1};
    void dfs(vector<vector<int>>& grid,int x,int y)
    {
        //沿岛屿陆地搜索边界
        if(x>=row||y>=col||x<0||y<0||grid[x][y]==0)//出界或遇水++
        {
            circle++;
            return;
        }
        if(grid[x][y]!=1) return;
        grid[x][y]++; //记录搜索过位置
        for(int i=0;i<4;i++)
        {
            dfs(grid,x+dx[i],y+dy[i]);
        }
    }
    int islandPerimeter(vector<vector<int>>& grid) {
        row=grid.size();
        col=grid[0].size();
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<col;j++)
            {
                if(grid[i][j]==1)
                {
                    dfs(grid,i,j);
                    break;
                }
            }
        }
        return circle;
    }
};
```