

## 463. Island Perimeter

You are given a map in form of a two-dimensional integer grid where 1 represents land and 0 represents water.

Grid cells are connected horizontally/vertically (not diagonally). The grid is completely surrounded by water, and there is exactly one island (i.e., one or more connected land cells).

The island doesn't have "lakes" (water inside that isn't connected to the water around the island). One cell is a square with side length 1. The grid is rectangular, width and height don't exceed 100. Determine the perimeter of the island.

Example:

Input:

```
[[0,1,0,0],  
 [1,1,1,0],  
 [0,1,0,0],  
 [1,1,0,0]]
```

Output: 16

```

class Solution {

    int[] dx = {1, 0, -1, 0};
    int[] dy = {0, 1, 0, -1};

    // dfs
    // 对于一片陆地,
    // 若其上方没有陆地, 则贡献一条边;
    // 若其左方没有陆地, 则贡献一条边;
    // 若其下方没有陆地, 则贡献一条边;
    // 若其右方没有陆地, 则贡献一条边;

    public int islandPerimeter(int[][] grid) {
        int perimeter = 0;

        for (int i = 0; i < grid.length; i++) {
            for (int j = 0; j < grid[i].length; j++) {
                if (grid[i][j] == 1) {
                    for (int k = 0; k < 4; k++) {
                        if (valid(i + dx[k], j + dy[k], grid)) {
                            perimeter++;
                        }
                    }
                }
            }
        }

        return perimeter;
    }

    private boolean valid(int x, int y, int[][] grid) {
        if (x >= 0 && x < grid.length && y >= 0 && y < grid[x].length) {
            return grid[x][y] == 0;
        }

        return true;
    }
}

```