

需要打开多少监控器

题目描述:

某长方形停车场，每个车位上方都有对应监控器，当且仅当在当前车位或者前后左右四个方向任意一个车位范围停车时，监控器才需要打开；

给出某一时刻停车场的停车分布，请统计最少需要打开多少个监控器；

输入描述:

第一行输入 m, n ， n 表示长宽，满足 $1 < m, n \leq 20$ ；后面输入 m 行，每行有 n 个 0 或 1 的整数，整数间使用一个空格隔开，表示该行已停车情况，其中 0 表示空位，1 表示已停；

输出描述:

最少需要打开监控器的数量；

示例 1

输入:

```
3 3
0 0 0
0 1 0
0 0 0
```

输出:

```
5
```

说明:

```
const rl = require("readline").createInterface({ input: process.stdin });
var iter = rl[Symbol.asyncIterator]();
const readline = async () => (await iter.next()).value;
```

```
void async function () {
    // Write your code here
    var firstLine = await readline();
    var tokens = firstLine.split(' ');
    var m = parseInt(tokens[0]);
    var n = parseInt(tokens[1]);
    var mat = new Array(m);
    for (var k = 0; k < m; k++) {
        mat[k] = new Array(n);
    }
    var ans = 0;
    for(var i = 0; i < m; i++){
        lines = (await readline()).split(" ");
        for(var j = 0; j < lines.length; j++){
            mat[i][j] = parseInt(lines[j]);
        }
    }
}
```

```

    var result = monitor(mat, m, n);
    console.log(result);
}()

function monitor(mat, m, n) {
    var direction = [-1, 0, 1, 0, -1];
    var ans = 0;
    var monitor_mat = new Array(m);
    for (var l = 0; l < m; l++) {
        monitor_mat[l] = new Array(n);
    }

    for(var i = 0; i < m; i++) {
        for(var j = 0; j < n; j++) {
            if(mat[i][j] == 1) {
                if(monitor_mat[i][j] != 1){
                    monitor_mat[i][j] = 1;
                    ans += 1;
                }
                continue;
            }else {
                for(var k = 0; k < 4; k++) {
                    var x = i + direction[k];
                    var y = j + direction[k+1];
                    if( x >= 0 && y >= 0 && x < m && y < m && mat[x][y] == 1) {
                        if(monitor_mat[i][j] != 1){
                            monitor_mat[i][j] = 1;
                            ans += 1;
                        }
                        continue;
                    }
                }
            }
        }
    }

    return ans;
}

```