

矩阵稀疏扫描

题目描述：

如果矩阵中的许多系数都为零，那么该矩阵就是稀疏的。对稀疏现象有兴趣是因为它的开发可以带来巨大的计算节省，并且在许多大的实践中都会出现矩阵稀疏的问题。

给定一个矩阵，现在需要逐行和逐列地扫描矩阵，如果某一行或者某一列内，存在连续出现的 0 的个数超过了行宽或者列宽的一半 $\lfloor W/2 \rfloor$ （地板除），则认为该行或者该列是稀疏的。

扫描给定的矩阵，输出稀疏的行数和列数。

扫描给定的矩阵，输出稀疏的行数和列数。

输入描述：

第一行输入为 M 和 N ，表示矩阵的大小 $M \times N$ ，

$0 < M \leq 100, 0 < N \leq 100$ 。

接下来 M 行输入为矩阵的成员，每行 N 个成员，矩阵成员都是有符号整数，范围 $-32,768$ 到 $32,767$ 。

输出描述：

输出两行，第一行表示稀疏行的个数，第二行表示稀疏列的个数。

示例 1

输入：

3 3

1 0 0

0 1 0

0 0 1

输出：

3

3

说明：

给定的 $3/\textit{times}3$ 矩阵里，每一行和每一列内都存在 2 个 0，行宽 3，列宽 3， $\lfloor 3/2 \rfloor = 1$ ，因此稀疏行有 3 个，稀疏列有 3 个。

示例 2

输入：

```
5 3
-1 0 1
0 0 0
-1 0 0
0 -1 0
0 0 0
10
```

输出：

```
5
3
```

说明：

给定的 $5/\textit{times}3$ 矩阵，每行里面 0 的个数大于等于 1 表示稀疏行，每列里面 0 的个数大于等于 2 表示稀疏列，所以有 5 个稀疏行，3 个稀疏列。

```
const rl = require("readline").createInterface({ input: process.stdin });
```

```
var iter = rl[Symbol.asyncIterator]();
```

```
const readline = async () => (await iter.next()).value;
```

```
void async function () {
```

```
    //创建一个空矩阵
```

```
    let data=[];
```

```
    while(line=await readline()){
```

```

        data.push(line.split(' '));
    }

    //遍历整个矩阵逐行逐列

    let n=parseInt(data[0][0]),m=parseInt(data[0][1]);

    //定义稀疏行稀疏列

    let ansx=0;

    let ansy=0;

    //count 计数,先求出稀疏行的数目(注意地板除 floor!!)

    for(let i=1;i<=n;i++){

        let count=0;

        //有一个 0 就计一个数

        for(let j=0;j<m;j++){

            if(data[i][j]=='0') count++;

        }

        if(count>=Math.floor(m/2)){

            ansx++;

        }

    }

    //count 计数,再遍历列求出稀疏列的数目(注意地板除 floor!!)

    for(let j=0;j<m;j++){

        let count=0;

        for(let i=1;i<=n;i++){

```

```
        if(data[i][j]=='O') count++;

    }

    if(count>=Math.floor(n/2)){

        ansy++;

    }

}

console.log(ansx);

console.log(ansy);

// Write your code here

while(line = await readline()){

    let tokens = line.split(' ');

    let a = parseInt(tokens[0]);

    let b = parseInt(tokens[1]);

    console.log(a + b);

}

}()
```