

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  $\lfloor W/2 \rfloor$  (整除)，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。

输入描述

第一行输入为M和N，表示矩阵的大小M\*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

用例

输入	3 3 1 0 0 0 1 0 0 0 1
输出	3 3
说明	给定的3*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。

输入	5 3 -1 0 1 0 0 0 -1 0 0 0 -1 0 0 0 0
输出	5 3
说明	给定的5*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表示稀疏行，所以有5个稀疏行,3个稀疏列。

## 题目解析

本题题目中说：

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  $[W/2]$  (整除)，则认为该行或者该列是稀疏的。

而用例里面对于稀疏行和稀疏列的确认，却不是根据连续0的个数，而是以0的个数（即不连续也可以）。

### 以用例说明为准（不以连续0为判断标准）

我的解题思路是定义两个数组：

- rowZeroCount数组，长度为m，rowZeroCount[i] 代表第 i 行中含0个数
- colZeroCount数组，长度为n，colZeroCount[j] 代表第 j 列中含0个数

这样的话，只要遍历输入的矩阵matrix的每一个元素matrix[i][j]，

如果matrix[i][j]==0，那么说明在第 i 行找到一个0，此时rowZeroCount[i]++，以及在第 j 列找到一个0，此时colZeroCount[j]++。

最后，只要分别统计rowZeroCount中有多少个大于  $n/2$ ，注意一行有n个元素，以及colZeroCount中有多少个大于  $m/2$ ，注意一列有m个元素。

```
1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int m = sc.nextInt();
9         int n = sc.nextInt();
10
11         int[] rowZeroCount = new int[m];
12         int[] colZeroCount = new int[n];
13
14         for (int i = 0; i < m; i++) {
15             for (int j = 0; j < n; j++) {
16                 if (sc.nextInt() == 0) {
17                     rowZeroCount[i]++;
18                     colZeroCount[j]++;
19                 }
20             }
21         }
22     }
```

```
23     System.out.println(Arrays.stream(rowZeroCount).filter(val -> val >= n /  
24     2).count());  
24     System.out.println(Arrays.stream(colZeroCount).filter(val -> val >= m /  
25     2).count());  
25 }  
26 }
```