

#### 题目描述：

在一个机房中，服务器的位置标识在  $n \times m$  的整数矩阵网格中，1 表示单元格上有服务器，0 表示没有。如果两台服务器位于同一行或者同一列中紧邻的位置，则认为它们之间可以组成一个局域网。

请你统计机房中最大的局域网包含的服务器个数。

#### 输入描述：

第一行输入两个正整数， $n$  和  $m$ ， $0 < n, m \leq 100$

之后为  $n \times m$  的二维数组，代表服务器信息

#### 输出描述：

最大局域网包含的服务器个数。

#### 补充说明：

##### 示例 1

#### 输入：

2 2

1 0

1 1

#### 输出：

3

#### 说明：

[0][0]、[1][0]、[1][1] 三台服务器相互连接，可以组成局域网

```
import java.util.Scanner;
```

// 注意类名必须为 Main，不要有任何 package xxx 信息

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner in = new Scanner(System.in);
```

```
        // 注意 hasNext 和 hasNextLine 的区别
```

```
        while (in.hasNextInt()) { // 注意 while 处理多个 case
```

```
            int a = in.nextInt();
```

```
            int b = in.nextInt();
```

```
            int[][] arr = new int[a][b];
```

```
            boolean[][] tag = new boolean[a][b];
```

```
            for (int i = 0; i < a; i++) {
```

```
                for (int j = 0; j < b; j++) {
```

```
                    arr[i][j] = in.nextInt();
```

```
                }
```

```
            }
```

```
            int max = 0;
```

```
            for (int i = 0; i < a; i++) {
```

```
                for (int j = 0; j < b; j++) {
```

```

        max = Math.max(func2(arr, tag, i, j), max);
    }
}

    System.out.println(max);
}
}

private static int func2(int[][] arr, boolean[][] tag, int i, int j) {
    int result = 0;

    if (i < 0 || j < 0 || i >= arr.length || j >= arr[0].length ||
        tag[i][j] == true || arr[i][j] == 0) {
        return 0;
    }

    if (arr[i][j] == 1 && tag[i][j] == false) {
        result += 1;
    }
    tag[i][j] = true;

    result += func2(arr, tag, i - 1, j);
    result += func2(arr, tag, i + 1, j);
    result += func2(arr, tag, i, j + 1);
    result += func2(arr, tag, i, j - 1);

    return result;
}
}

```