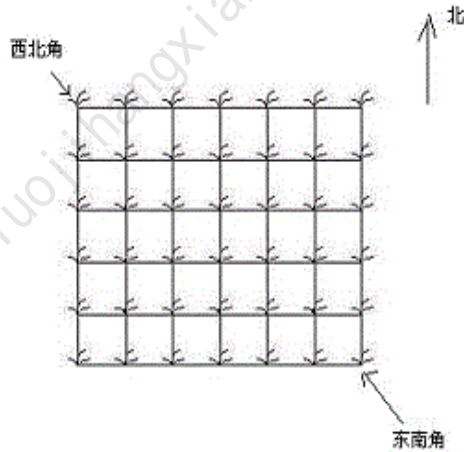


1284: 摘花生

题目描述

Hello Kitty想摘点花生送给她喜欢的米老鼠。她来到一片有网格状道路的矩形花生地(如下图),从西北角进去,东南角出来。地里每个道路的交叉点上都有种着一株花生苗,上面有若干颗花生,经过一株花生苗就能摘走该它上面所有的花生。Hello Kitty只能向东或向南走,不能向西或向北走。问Hello Kitty最多能够摘到多少颗花生。



输入

第一行是一个整数 T ,代表一共有多少组数据。 $1 \leq T \leq 100$,接下来是 T 组数据。

每组数据的第一行是两个整数,分别代表花生苗的行数 R 和列数 C ($1 \leq R, C \leq 100$)

每组数据的接下来 R 行数据,从北向南依次描述每行花生苗的情况。每行数据有 C 个整数,按从西向东的顺序描述了该行每株花生苗上的花生数目 M ($0 \leq M \leq 1000$)。

输出

对每组输入数据,输出一行,内容为Hello Kitty能摘到得最多的花生颗数。

输入样例

```
2
2 2
1 1
3 4
2 3
2 3 4
1 6 5
```

输出样例

8
16

分析

通过观察图像，我们可以发现，每个点的花生数量等于：当前点的数量 + max（上方点的数量，左侧点的数量），我们定义 $a[i][j]$ 代表每个点最初的花生数量，定义 $f[i][j]$ 代表每个点能够摘取的最大花生数量，则有 $f[i][j] = a[i][j] + \max(f[i-1][j], f[i][j-1])$

编码

```
#include<bits/stdc++.h>

using namespace std;
int ans[110]; //存储多组答案的数组
int a[101][101]; //原始图像数组
int f[101][101]; //i行j列的最大花生数

int main() {
    int t; //测试组数
    cin >> t;
    int b = t;
    while (t--) {
        int n, m; //行列数
        cin >> n >> m;
        //注意因为是多组数据，因此每次都需要重置
        memset(a, 0, sizeof(a));
        memset(f, 0, sizeof(f));
        for (int i = 1; i <= n; i++) //从1开始就是为了防止越界
        {
            for (int j = 1; j <= m; j++) {
                cin >> a[i][j];
                //这里初始化和最长上升序列不一样，要初始化为它本身
                f[i][j] = a[i][j];
            }
        }
        //依次计算每一个点的最大值
        for (int i = 1; i <= n; i++) {
            for (int j = 1; j <= m; j++) {
                //用当前点的数量加上左侧或者上侧的最大值
                f[i][j] = a[i][j] + max(f[i-1][j], f[i][j-1]);
                ans[t] = max(ans[t], f[i][j]);
            }
        }
        //输出最终的最大值
        for (int i = b - 1; i >= 0; i--)
```

```
    cout << ans[i] << endl;  
    return 0;  
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

