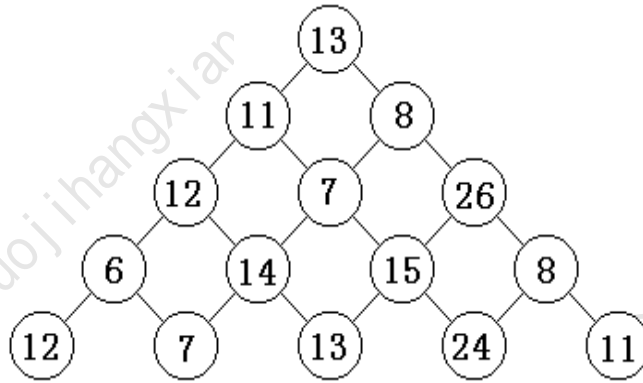


1258: 数字金字塔

题目描述

观察下面的数字金字塔。写一个程序查找从最高点到底部任意处结束的路径，使路径经过数字的和最大。每一步可以从当前点走到左下方的点也可以到达右下方的点。



在上面的样例中，从13到8到26到15到24的路径产生了最大的和86。

输入

第一个行包含 $R(1 \leq R \leq 1000)$ ，表示行的数目。

后面每行为这个数字金字塔特定行包含的整数。

所有的被供应的整数是非负的且不大于100。

输出

```
5
13
11 8
12 7 26
6 14 15 8
12 7 13 24 11
```

输出

86

解析

看图分析，参考上升子序列的思想，计算出每一个点与上一行可接触数字的最大和，然后和全局最大值比较即可。

将输入的数据进行整理如下：

13				
11	8			
12	7	26		
6	14	15	8	
12	7	13	24	11

13				
24	21			
36	31	47		
42	61	62	8	
73	69	75	86	19

编码

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

using namespace std;
const int M = 1001;
int Dp[M][M], Map[M][M];
//最终结果
int MaxValue;

int main() {
    int n;
    //读入行数
    cin >> n;
    //读入初始数据
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            cin >> Map[i][j];
        }
    }
    //从上至下寻找最大值
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            //dp[行][列] = max(dp[行-1][列-1], dp[行-1][列]) + map[行][列]
            Dp[i][j] = max(Dp[i - 1][j - 1], Dp[i - 1][j]) + Map[i][j];
            //寻找最大值
            MaxValue = max(MaxValue, Dp[i][j]);
        }
    }

    cout << MaxValue;
    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。



] [列]