

递推习题

T E A C H I N G C O U R S E W A R E P O W P O I N T

授课时间：2025.08.01



目录

- PART-01 题目描述 TEACHING COURSEWARE
- PART-02 概念分析 TEACHING COURSEWARE
- PART-03 经典例题 TEACHING COURSEWARE
- PART-04 教学反思 TEACHING COURSEWARE

01

题目描述

TEACHING
COURSEWARE

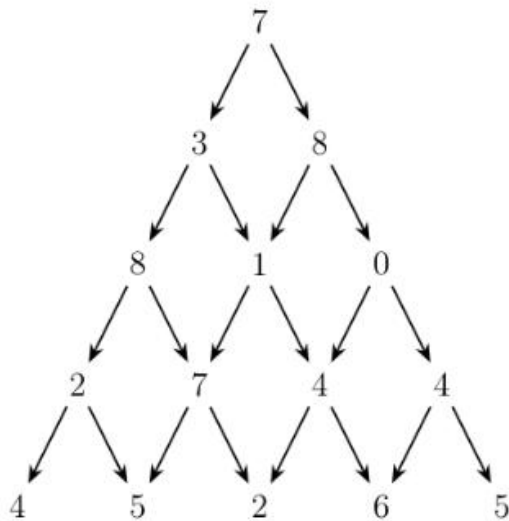
TEACH



题目描述

观察下面的数字金字塔。

写一个程序来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。每一步可以走到左下方的点也可以到达右下方的点。



洛谷

在上面的样例中，从 $7 \rightarrow 3 \rightarrow 8 \rightarrow 7 \rightarrow 5$ 的路径产生了最大权值。

输入格式

第一个行一个正整数 r ，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。

输出格式

单独的一行,包含那个可能得到的最大的和。

02

知识回归

TEACHING
COURSEWARE

TEACH

其核心逻辑是：

将一个复杂问题分解为与规模更小的同类问题的关联关系，即通过描述“第 n 项”与“前若干项”（如第 $n-1$ 项、第 $n-2$ 项等）之间的确定关系，结合初始条件（通常是规模最小的若干项的已知结果），逐步计算出更大规模问题的解。

03

题目解答

TEACHING
COURSEWARE

TEACH

逆向求解思路：

而是从最底层开始向上递推。具体做法是，从倒数第二行起，每行的每个元素都加上其正下方和右下方两个元素中的较大值，以此更新当前元素的值（此时该值代表从当前位置到底部的最大路径和）。

通过这种方式逐层向上更新，最终顶部元素的值就是从起点到终点的最大路径和。

这种方法利用递推思想，将复杂的路径选择转化为简单的局部最优判断。

04

标准程序

TEACHING
COURSEWARE

TEACH

标准程序

```
#include<iostream>
#include<cmath>
using namespace std;
int n;
int a[1001][1001]; // 下标从1开始, 数组大小设为1001以容纳n=1000的情况
int main()
{
    cin >> n;
    // 输入, i从1到n, j从1到i
    for(int i=1;i<=n;i++)
        for(int j=1;j<=i;j++)
            cin >> a[i][j];
    // 从倒数第二行 (i=n-1) 向上递推到第一行 (i=1)
    for(int i=n-1;i>=1;i--)
    {
        for(int j=1;j<=i;j++)
            // 当前位置的值加上下方或右下方的较大值
            a[i][j] += max(a[i+1][j], a[i+1][j+1]);
    }
    // 输出顶点 (1,1) 的值
    cout << a[1][1] << endl;
    return 0;
}
```