

递推

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 概念引入 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

概念引入

TEACHING
COURSEWARE

TEACH

概念引入

攒钱计划

假设你每月攒钱，第一个月攒 1000 元，从第二个月开始，每个月比上个月多攒 200 元。想知道第 6 个月攒了多少，以及半年总共攒了多少。

初始值：第 1 个月攒 1000 元。

递推规则：第 n 个月攒的钱 = 第 $n-1$ 个月攒的钱 + 200。

比如：

$$\text{第 2 个月} = 1000 + 200 = 1200$$

$$\text{第 3 个月} = 1200 + 200 = 1400$$

...

$$\text{第 6 个月} = \text{第 5 个月的钱} + 200$$

02

概念分析

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

经典例题

TEACHING
COURSEWARE

TEACH



经典例题

说明

输出斐波那契数列第 n 项。0, 1, 1, 2, 3, 5, 8, 13.....

输入格式

一个正整数 n ,表示第 n 项。

输出格式

第 n 项是多少。

样例

输入数据 1

4

Copy

输出数据 1

2

Copy



找递推式

```
#include <iostream>
using namespace std;
int fib[50];
int main() {
    int n;
    // 输入第 n 项
    cin >> n;
    // 初始化前两项, 根据题目数列 0, 1, 1, 2..., 第 1 项是 0, 第 2 项是 1
    fib[1] = 0;
    fib[2] = 1;
    // 递推计算从第 3 项到第 n 项
    for (int i = 3; i <= n; i++) {
        fib[i] = fib[i - 1] + fib[i - 2];
    }
    // 输出第 n 项结果
    cout << fib[n] << endl;
    return 0;
}
```

经典例题

说明

树老师爬楼梯，他可以每次走1级或者2级，输入楼梯的级数，求不同的走法数。

例如：楼梯一共有3级，他可以每次都走一级，或者第一次走一级，第二次走两级，也可以第一次走两级，第二次走一级，一共3种方法。

输入格式

输入包含若干行，每行包含一个正整数 N ，代表楼梯级数， $1 \leq N \leq 30$ 。

输出格式

不同的走法数，每一行输入对应一行输出。

样例

输入数据 1

```
5
8
10
```

[Copy](#)

输出数据 1

```
8
34
89
```

[Copy](#)



解题思路

实现思路：

利用递推公式从基础情况 ($n=1$ 、 $n=2$) 逐步计算到目标

n ；

使用循环迭代计算, 时间复杂度 $O(n)$, 空间复杂度 $O(1)$ ；

由于 $n \leq 30$, 结果不会超出 `int` 范围, 无需高精度处理。

标准程序

```
#include <iostream>
using namespace std;
int ways[31]; // 存储各级台阶的走法数
int main() {
    int n; // 楼梯级数
    while (cin >> n) {
        // 设置初始条件
        ways[1] = 1; // 1级台阶: 1种走法
        if (n >= 2) {
            ways[2] = 2; // 2级台阶: 2种走法
        }
        // 递推计算从3级到n级的走法数
        for (int i = 3; i <= n; ++i) {
            // 递推关系: 当前台阶的走法数 = 前1级的走法数 + 前2级的走法数
            ways[i] = ways[i - 1] + ways[i - 2];
        }
        // 输出n级台阶的走法数
        cout << ways[n] << endl;
    }
    return 0;
}
```

04

Q&A

TEACHING
COURSEWARE

TEACH



Q&A

Q:爬楼梯问题，如果改成 一次最多走 3 阶，递推式怎么变？

Q:递推的边界