

题目讲解

- Leetcode 207. Course Schedule
- 赤裸裸的拓扑排序题
- 拓展：拓扑排序可以判断有向图的环

递归

Ben

递归定义

- 直接或间接调用自身
- 算法思想：
 - 原问题可分解子问题（必要条件）
 - 原与分解后的子问题相似（递归方程）
 - 分解次数有限（子问题有穷）
 - 最终问题可直接解决（递归边界）

递归奥义

- 递归 = 递 + 归
- 会想还要会写 => 实践出真知
- 递归奥义——复制自己
 - 如何造纳米机器人

递归奥义

- 递归框架

```
• int robot(int x, int y)           // 机器人的输入
    if (边界条件)                   // 什么时候不用造了（自己就能干完）
        return 0;
    int a = robot(x1, y1);           // 造一个小的自己帮忙干活
    int b = robot(x2, y2);           // 再造一个小的自己帮忙干活
    return a + b;                    // 自己要做的就是别人的成果组装起来
```

经典问题

- 斐波那切数列
- 汉诺塔问题
- 快速排序
- 二叉树遍历

菲波那切数列

- 递归方程
 - $F(n) = F(n - 1) + F(n - 2)$ //可分解, 有穷性
- 递归边界
 - $F(0) = 1, F(1) = 1$
- 经典例子
 - 兔子生兔子
 - 爬楼梯

菲波那切数列

- 直接递归
 - 时间复杂度**预估**
- 优化递归
 - 不做重复的事
- 递归转递推
 - 递归式，递推式？仅仅是方向不一样
- 为什么用递归
 - 方便而已！

菲波那切数列

- 如何更优?
- 直接使用公式

$$F(n) = \frac{1}{\sqrt{5}} \left(\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right)$$

- 公式的问题
 - N次方如何算?

幂运算

- 求 a^n
- $F(n) = aF(n - 1), \quad F(0) = 1$
- 时间复杂度 $O(n)$
- 换种写法
- $F(n) = F(n / 2)^2 * a \quad \text{if } n \% 2 == 1$
- $\quad = F(n / 2)^2 \quad \text{if } n \% 2 == 0$
- 时间复杂度 $O(\log n)$

斐波那切数列

- 斐波那切数列的本质
- 二阶齐次线性递推式

$$\begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} f(n-1) \\ f(n-2) \end{bmatrix}$$

- 反复应用

$$\begin{bmatrix} f(n) \\ f(n-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} f(n-1) \\ f(n-2) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^2 \begin{bmatrix} f(n-2) \\ f(n-3) \end{bmatrix} = \cdots = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{n-2} \begin{bmatrix} f(2) \\ f(1) \end{bmatrix}$$

菲波那切数列

- 矩阵运算具有结合律性质
- 适用幂运算的优化方法
- 时间复杂度： $O(\log n)$
- 优化： $O(\text{指数级}) \rightarrow O(n) \rightarrow O(\log n)$
- 本质： 减少重复计算

菲波那切数列

- 课堂作业
 - 非齐次
 - $F(n) = F(n - 1) + F(n - 2) + a$
- 课后作业
 - K阶
 - $F(n) = F(n - 1) + F(n - 2) + \dots + F(n - k)$
- 终极作业
 - $F(n) = F(n - 1) + F(n - 2) + \dots + F(n - k) + a$

汉诺塔

- 有三根柱子A,B,C。在A上，按照从小到大摞着N片圆盘。现要把圆盘按原顺序重新摆放在C上，并且规定，任何时刻，在小圆盘上不能放大圆盘，一次只能从一个柱子最顶部移一个圆盘到另一柱子顶部
- 最短移动次数，以及移动方案
- 试试 $N=1$ ， $N=2$
- $N=3$?

汉诺塔

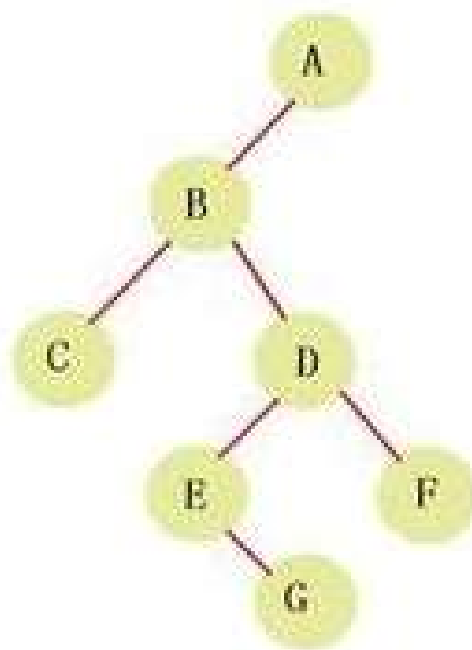
- 找到 $F(N)$ 和 $F(N-1)$ 的关系
- \Rightarrow 已知 $F(N-1)$, 如何求出 $F(N)$?
- 数学归纳法的程序版
- 方案输出
 - 从A柱子, 利用B柱, 到达C柱
 - 递归时候如何记录状态?
 - $F(N, A, B, C)$

快速排序

- 期望时间复杂度 $O(N\log N)$
- 思想：
 - 1. 找任意一个元素作为中间值 m
 - 2. 比 m 小的放在数组前部，大的放后部
 - 3. 前部后部分别排序（递归）
- 思想 -> 实现
- 复杂度计算

二叉树遍历

- 三种不同的遍历顺序
- 前序遍历：ABCDEGF (根左右)
- 中序遍历：CBEGDFA (左根右)
- 后序遍历：CGEFDDBA (左右根)
- 为什么要区别各种序
 - 三种序 -> 三视图



二叉树遍历

- 前序遍历：
 - `write(根); find(left); find(right)`
- 中序遍历：
 - `find(left); write(根); find(right)`
- 后序遍历：
 - `find(left); find(right); write(根);`

作业

- 汉诺塔问题，输出步骤并验证
- 亲手写一个快速排序，并通过测试
- 实现二叉树遍历的三种方式