

# 递推方程与 算法分析

# 递推方程

设序列  $a_0, a_1, \dots, a_n, \dots$ , 简记为  $\{a_n\}$ ,  
一个把  $a_n$  与某些个  $a_i (i < n)$  联系起来的  
等式叫做关于序列  $\{a_n\}$  的递推方程

递推方程的求解:

给定关于序列  $\{a_n\}$  的递推方程和若干初值, 计算  $a_n$

# 递推方程的例子

Fibonacci数

1, 1, 2, 3, 5, 8, 13, 21,  
34, 55, ...



数学家Fibonacci  
意大利1170-1240

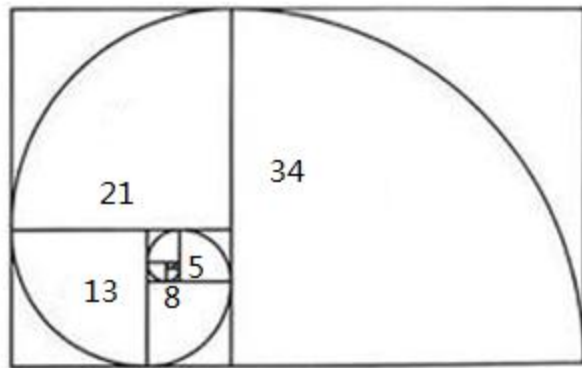
递推方程:  $f_n = f_{n-1} + f_{n-2}$

初值:  $f_0 = 1, f_1 = 1$

解:

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

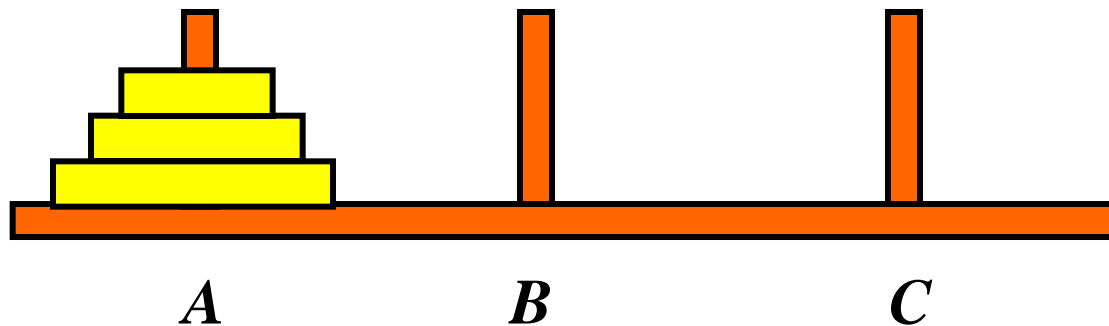
# Fibonacci数的存在



55



# Hanoi塔问题



$n$ 个盘子从大到小顺序放在A 柱上，  
要把它们从 A 移到 C，每次移动 1个  
盘子，移动时不允许大盘压在小盘上。  
设计一种移动方法。

# 递归算法

算法 Hanoi ( $A, C, n$ ) //  $n$ 个盘子 $A$ 到 $C$

1. if  $n=1$  then move ( $A, C$ ) // 1个盘子 $A$ 到 $C$
2. else Hanoi ( $A, B, n-1$ )
3.     move ( $A, C$ )
4.     Hanoi ( $B, C, n-1$ )

设  $n$ 个盘子的移动次数为  $T(n)$

$$T(n) = 2 T(n-1) + 1,$$

$$T(1) = 1,$$

# 分析算法

$$T(n) = 2 T(n-1) + 1, \quad T(1) = 1,$$

解

$$T(n) = 2^n - 1$$

1 秒移1个，64个盘子要多少时间？

5000亿年！千万亿次/秒，4个多小时



有没有更好的算法？

没有！这是一个难解的问题，不存在多项式时间的算法！

# 插入排序

算法 Insert Sort ( $A, n$ )

1. for  $j \leftarrow 2$  to  $n$
2.      $x \leftarrow A[j]$
3.      $i \leftarrow j-1$      // 把  $A[j]$  插入
4.     while  $i > 0$  and  $x < A[i]$  do
5.          $A[i+1] \leftarrow A[i]$
6.          $i \leftarrow i-1$
7.      $A[i+1] \leftarrow x$



# 最坏情况下时间复杂度

插入排序：

设基本运算是元素比较，对规模为  $n$  的输入最坏情况下的时间复杂度  $W(n)$

$$W(n) = W(n-1) + n - 1$$

$$W(1) = 0$$

解为

$$W(n) = n(n-1)/2$$

# 小结

- 递推方程的定义及初值
- 递推方程与算法时间复杂度的关系

Hanoi塔的递归算法

插入排序的迭代算法