

Lesson 06

递归

常量降级 快速 review

Hanoi 塔

要把 n 个 板子 从 A 挪到 C 柱子，你可以有 B 柱子使

两个函数：

$S(n, a, b, c)$

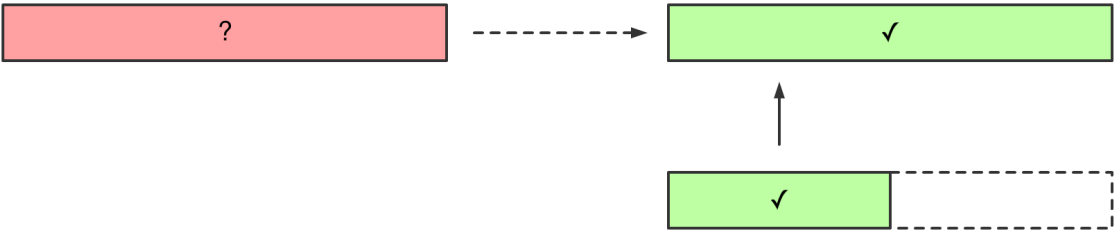
有 3 个柱子 a b c，将 n 个板子从 a 挪到 c

$move(a, b)$

将 1 个板子 从 a 挪到 b

倍数降级

套路



假设 你已经拿到了 $n/2$ 的问题的答案

想一下 是否可以通过这个答案 解决 n 的问题的答案

思路案例

问题：

求 n 个数的 最大值

对照表：

标题	对照
大小为 n 的问题	如何求出 n 个数 的 最大值
大小为 n 的答案 $S(n)$	n 个数 的 最大值
大小为 $n/2$ 的问题	如何求出 前一半 数 的 最大值
大小为 $n/2$ 的答案 $S(n/2)$	前一半 数 的 最大值

思路：

假设 我已经知道了 前 $n/2$ 个数 的最大值，
想一下 是否可以通过 这个值 求出 n 个数 的最大值呢

训练：

求 2 的 n 次方

递归下的时间复杂度分析

代数法

算法：

```

ArrayMax( $A, n$ )
  if  $n = 1$  then
    return  $A_1$ 
  return  $\max(\text{ArrayMax}(A, n - 1), A_n)$ 

```

分段分析：

$T(n) = O(1)$ when $n = 1$
 $T(n - 1) + c$ when $n > 1$

求值：

$$\begin{aligned}
 T(n) &= c + T(n - 1) \\
 &= c + (c + T(n - 2)) \\
 &= c + (c + (c + T(n - 3))) \\
 &= \dots \\
 &= c + c + c + \dots + c + O(1) \\
 &= c + c + c + \dots + c + c_2 \\
 &= cn + c_2 \\
 &= O(n)
 \end{aligned}$$

递归树法

公式：

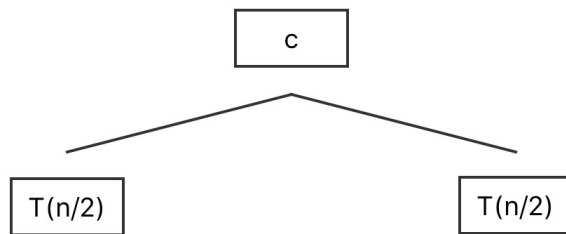
$T(n) = O(1)$ when $n = 1$
 $2 \cdot T(n/2) + c$ when $n > 1$

分叉：

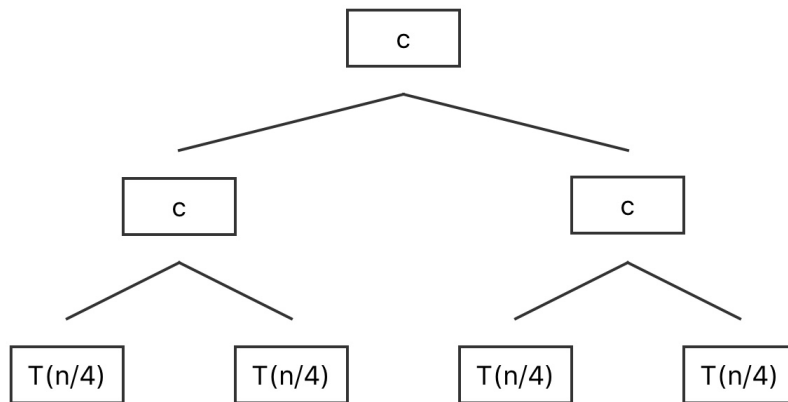
1

$T(n)$

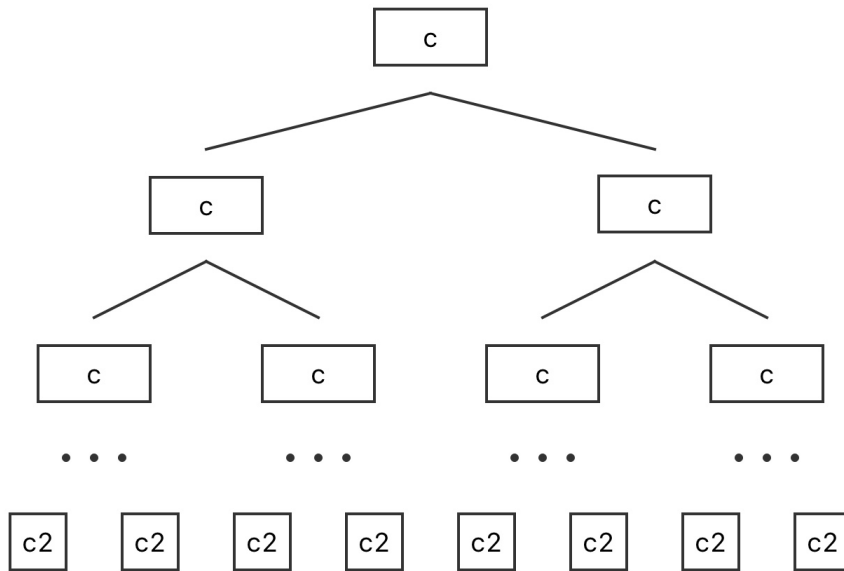
2



3



4



计算:

计算每层的总合

$$\begin{aligned}
 T(n) &= c + 2c + 4c + \dots + nc + n \cdot c_2 \\
 &= 2^0 c + 2^1 c + 2^2 c + \dots + 2^k c + n \cdot c_2 \\
 &= 2 \cdot 2^k c - 1 + n \cdot c_2 \\
 &\quad k = \log_2 n \\
 &= 2 \cdot n \cdot c - 1 + n \cdot c_2 \\
 &= O(n)
 \end{aligned}$$

尝试

1:

```

1  public int power2(int n) {
2      int result = power2(n / 2) * power2(n / 2);
3      if (n % 2 == 1) {
4          result *= 2;
5      }
6      return result;
7  }

```

2:

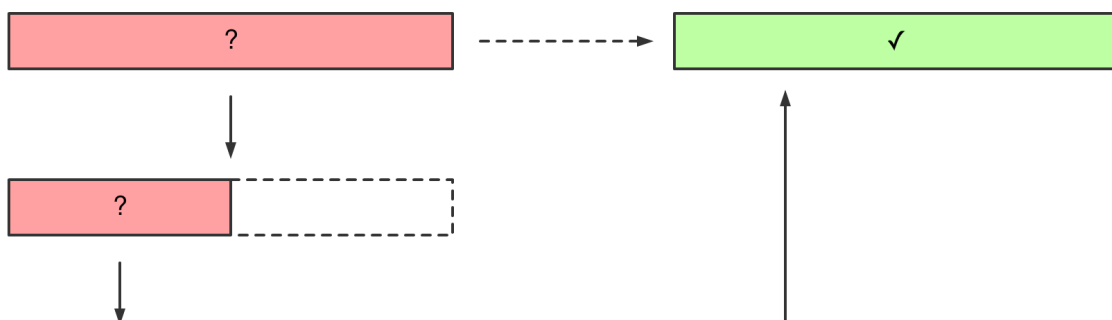
```
1 public int power2(int n) {  
2     int half = power2(n / 2);  
3     int result = half * half;  
4     if (n % 2 == 1) {  
5         result *= 2;  
6     }  
7     return result;  
8 }
```

分区

Partition

分区思维

思路





不直接解决问题

转而去想，我能不能把问题拆成 2 个，分别解决，

或者，在有可能的情况下，扔掉 1 个

二叉搜索

Binary Search

问题

给定一个升序序列 $A = \langle A_1, A_2, \dots, A_n \rangle$

找到这个序列里是否存在一个数 q

任务

代码

时间复杂度分析

递归 Basecase 分析

找相对低的值，尝试，看最后问题都落到了多少规模的。

sublist 复杂度解除

传参