递归式复杂度计算 | 迭代法

II. 迭代法

- 思想:模拟该递归关系执行过程，从而计算算法运行时间
- 两种方法 (代数algebraic 和几何geometrical)
  - 直接展开 (algebraic)
  - 递归树 (geometrical)

电子科技大学

直接展开

- 方法:

展开递归关系式并将该展开的递归关系式表达成仅依赖于n和初始条件的各项之和的形式。

例如

$$
\begin{aligned}
T(n) &= n+3T(n/4) \\
&= n+3(n/4+3T(n/16)) \\
&= n+3(n/4+n+3(n/16+3T(n/64)) \\
&= n+3n/4+9n/16+27T(n/64)
\end{aligned}
$$

$$
\begin{aligned}
T(n) &\le n+3n/4+9n/16+\cdots+3^{\log_4 n}\Theta(1) \\
&\le n\sum_{i=0}^{\infty}\left(\frac{3}{4}\right)^i+\Theta(n^{\log_4 3}) \\
&= 4n+o(n) = O(n)
\end{aligned}
$$

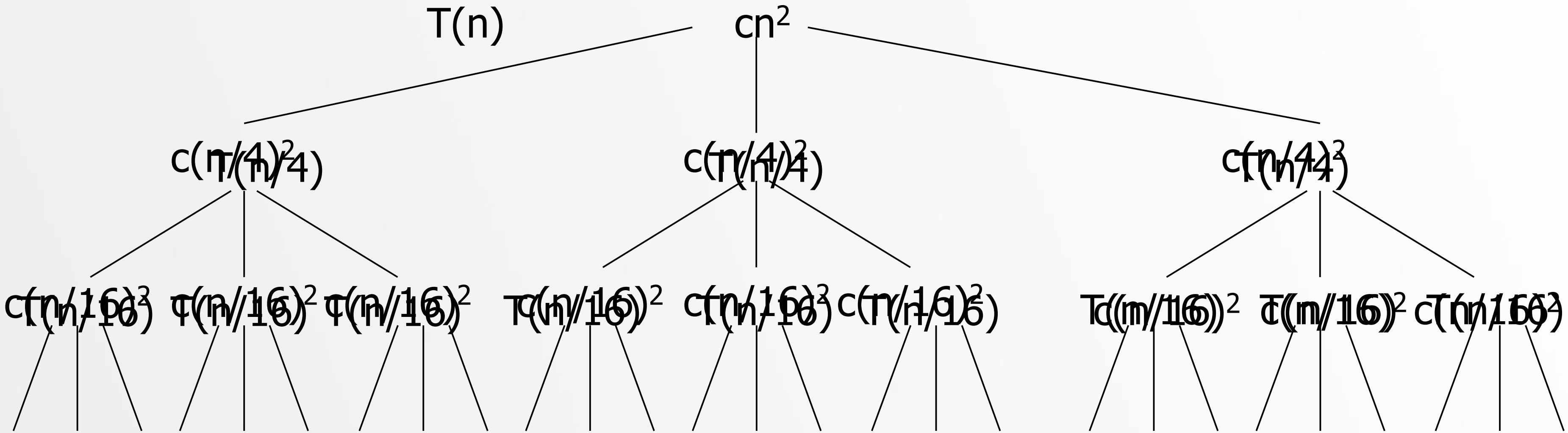
递归树

◆画出该递归关系式的递归树

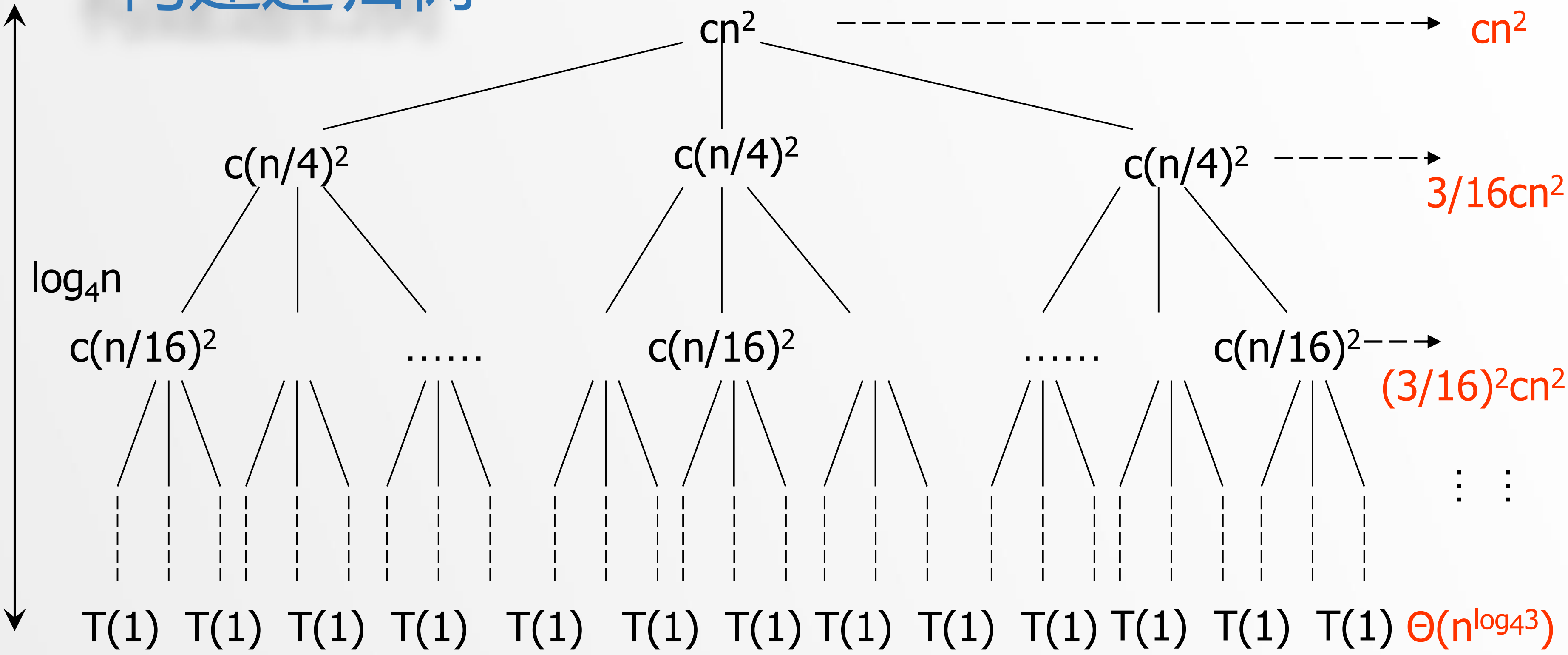
- 树中每个**节点**都代表递归函数调用集合中**一个子问题的代价**
- **每一层的代价**相加得到一个每层代价的集合
- 层代价相加得到**递归的总代价.**

例如

- $T(n)=3T(n/4)+\Theta(n^2)$



构建递归树



$N^{\log_4 3}$叶子节点

$$a^{\log_b c} = c^{\log_b a}$$

电子科技大学

通过递归树来计算

- 第k层的每个子问题规模为 $n/4^k$
- 第k层每个节点的代价为 $c(n/4^k)^2$
- 第k层有 3^k 个节点
- 每一层的代价为 $3^k*c(n/4^k)^2=(3/16)^k cn^2$
- 该树有 $\log_4 n+1$ 层
- 有 $3^{\log_4 n}=n^{\log_4 3}$ 个叶子节点代价为 $\Theta(1)$

无穷等比数列和

$$
\begin{aligned}
T(n) &= \sum_{i=0}^{\log_4 n-1} (3/16)^i cn^2 + \theta\left(n^{\log_4 3}\right) \\
&< \sum_{i=0}^{\infty} (3/16)^i cn^2 + \theta\left(n^{\log_4 3}\right) \\
&= \frac{1}{1-3/16} cn^2 + \theta\left(n^{\log_4 3}\right) \\
&= \frac{16}{13} cn^2 + \theta\left(n^{\log_4 3}\right) \\
&= O\left(n^2\right)
\end{aligned}
$$

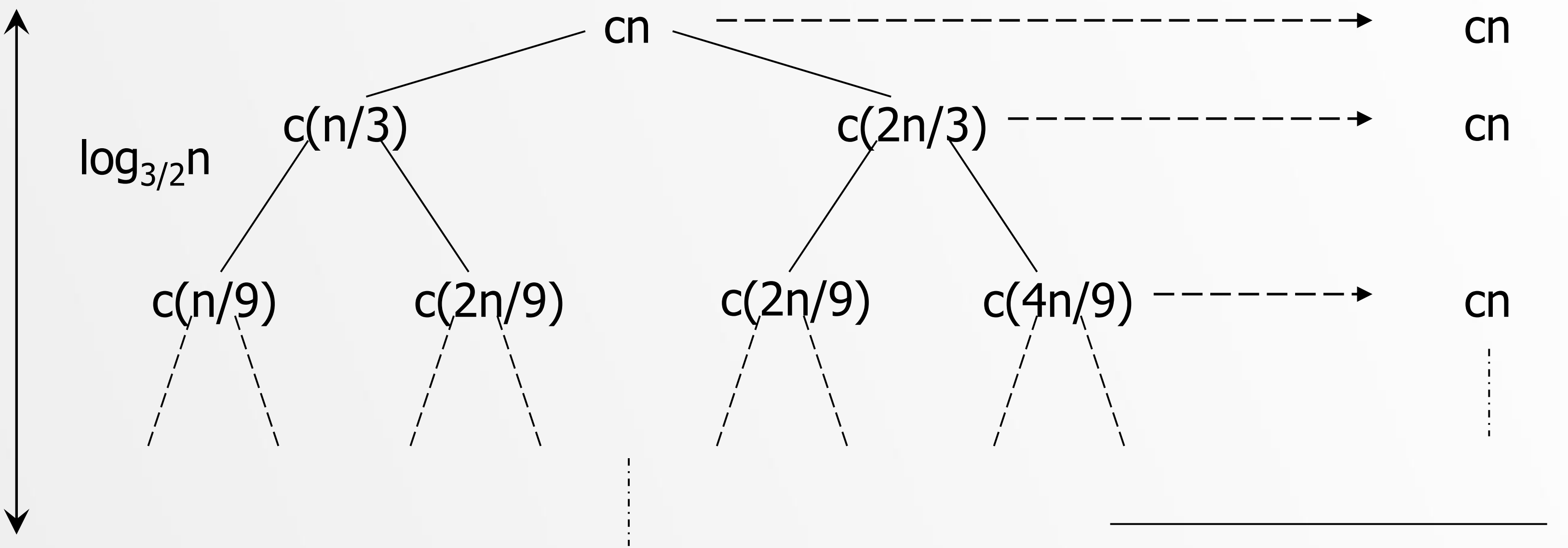
思 考：

请用递归树法求解：

$$T(n)=T(n/3)+T(2n/3)+O(n)$$

一个更复杂的例子

$$T(n)=T(n/3)+T(2n/3)+O(n)$$



共 O(nlgn)

通过递归树计算

- 树的高度 h: $n(2/3)^h=1 \rightarrow h=\log_{3/2}n$
- 每一层的代价为 cn
- $T(n)=O(cn*\log_{3/2}n)=O(n\lg n)$
- 是否正确?

电子科技大学

通过递归树计算

- 树的高度 h: $n(2/3)^h=1 \rightarrow h=\log_{3/2}n$
- 每一层的代价为 cn
- $T(n)=O(cn*\log_{3/2}n)=O(n\lg n)$
- 我们还没有将叶子节点的代价考虑进去!
- 如果是完全二叉树，叶子节点至多有 $2^{\log_{3/2}n}$, 每一个的代价为 $\Theta(1)$.
- 所以叶子节点的总代价为 $\Theta(2^{\log_{3/2}n})=\Theta(n^{\log_{3/2}2})=\omega(n\lg n)$.
- 可是, 这并不是一颗完全的二叉树, 所以叶子节点小于 $2^{\log_{3/2}n}$

利用代换法证明

- T(n)=O(nlgn)
- 思路，设一个常数d，证明T(n) ≤dnlgn

$$
\begin{aligned}
T(n) &\le T(n/3) + T(2n/3) + cn \\
&\le d(n/3)\lg(n/3) + d(2n/3)\lg(2n/3) + cn \\
&= (d(n/3)\lg n - d(n/3)\lg 3) + (d(2n/3)\lg n - d(2n/3)\lg(3/2)) + cn \\
&= dn\lg n - d((n/3)\lg 3 + (2n/3)\lg(3/2)) + cn \\
&= dn\lg n - d((n/3)\lg 3 + (2n/3)\lg 3 - (2n/3)\lg 2) + cn \\
&= dn\lg n - dn(\lg 3 - 2/3) + cn \le dn\lg n
\end{aligned}
$$

当d≥c/(lg3-(2/3))

- 递归树最适合用来产生好的猜测，然后用代换法加以验证。产生猜测时，可以容忍小的“不良量”，因为后面会证明所做的猜测。
- 如果使用递归树非常仔细，将所有代价都累加了起来，则可以直接用递归树作为递归式解的证明。

作 业

请用递归树法求解：

$$T(n)=2T(n/3)+T(2n/3)+n^2$$