

# 算法分析与设计选做第二次作业

57119134 黄浩

2021 年 8 月 17 日

证明下述结论。

## 分治法的计算效率

一个分治法将规模为 $n$ 的问题分成 $k$ 个规模为 $n/m$ 的子问题去解。设分解阈值 $n_0=1$ ，且adhoc解规模为1的问题耗费1个单位时间。再设将原问题分解为 $k$ 个子问题以及用merge将 $k$ 个子问题的解合并为原问题的解需用 $f(n)$ 个单位时间。用 $T(n)$ 表示该分治法解规模为 $|P|=n$ 的问题所需的计算时间，则有：

$$T(n) = \begin{cases} O(1) & n = 1 \\ kT(n/m) + f(n) & n > 1 \end{cases}$$

**通过迭代法求得方程的解：**  $T(n) = n^{\log_m k} + \sum_{j=0}^{\log_m n - 1} k^j f(n/m^j)$

证明：

我们可以分析每一层的复杂度，再加上最后一层的复杂度从而得到总复杂度：

第零层：1 个节点，并开销  $f(n)$

第一层： $k$  个节点，并开销  $k \cdot f(n/m)$

第二层： $k^2$  个节点，并开销  $k^2 \cdot f(n/m^2)$

第三层： $k^3$  个节点，并开销  $k^3 \cdot f(n/m^3)$

...

第  $x$  层： $k^x$  个节点，并开销  $k^x \cdot f(n/m^x)$

...

由上我们也易知最后一层的叶子数为  $k^{\log_m n} = n^{\log_m k}$ ，所以最后一层的最简问题开销为  $n^{\log_m k}$ 。

所以，我们将每一层的开销和最终最简问题的开销求和即是问题总开销：（注：最后一层由于已经是最简问题了，所以最后一层没有分解子问题的开销）

$$T(n) = n^{\log_m k} + \sum_{j=0}^{\log_m n - 1} k^j \cdot f(n/m^j)$$

得证。

另外：

我们可以总结出一些直接得出递归式计算复杂度的方法：

当  $O(n^{\log_m k}) < O(f(n))$  时， $T(n) = O(f(n))$

当  $O(n^{\log_m k}) = O(f(n))$  时， $T(n) = O(n^{\log_m k} \cdot \log_m n)$

当  $O(n^{\log_m k}) > O(f(n))$  时， $T(n) = O(n^{\log_m k})$