HIT
CS&E

第二章

算法分析的数学基础

程思瑶
计算机科学与技术学院

HIT
CS&E

参考资料

《Introduction to Algorithms》
- 第三章
- 第四章
- 附　录

《Concrete Mathematics:
A Foundation for Computer Science》
Ronald L.Graham, *Donald E.Knuth*, and *Oren Patashnik*

提要

2.1 计算复杂性函数的阶
2.2 标准符号和通用函数
2.3 和式的计算与估计
2.4 递归方程

2.1 计算复杂性函数的阶

- 同阶函数
- 低阶函数
- 高阶函数
- 严格低阶函数
- 严格高阶函数
- 函数阶的性质

2.1.1 同阶函数集合

定义2.1.1. 设$f(n)$和$g(n)$是正值函数。如果$\exists c_1, c_2>0, n_0, \forall n>n_0, c_1g(n)\leq f(n)\leq c_2g(n)$，则称$f(n)$与$g(n)$ 同阶，记作$f(n)=\theta(g(n))$。

$\theta(g(n))$可以视为如下集合：

$\{f(n) \mid \exists c_1, c_2>0, n_0, \forall n>n_0, c_1g(n)\leq f(n)\leq c_2g(n)\}$

即所有与$g(n)$同阶的函数的集合

2.1.1 同阶函数集合

- 例1，证明：$(1/3)n^2-3n=\theta(n^2)$

$\exists c_1, c_2>0, n_0, \forall n>n_0,$

$$c_1 n^2 \le (1/3)n^2-3n \le c_2 n^2$$

对于左侧不等式，$\forall n>1$，有:

$$c_1 \le (1/3)-3/n = (1/6)+(1/6)-3/n$$

即当$n>18$，$c_1=1/6$时，不等式成立

对于右侧不等式，$\forall n>1$，有: $(1/3)-3/n \le c_2$,

即当$n>18$，$c_2=1/3$时，不等式成立

2.1.1 同阶函数集合

例 2 证明 $f(n)=an^2+bn+c=\theta(n^2)$, $a>0$

证. 设 $c_1n^2 \le an^2+bn+c \le c_2n^2$，令 $c_1=a/4$, $c_2=7a/4$,则

$$\frac{a}{4}n^2 \le an^2+bn+c \le \frac{7}{4}an^2,$$

令 $n_0=2\cdot\max\left(\left(|b|/a\right),\sqrt{|c|/a}\right)$。当 $n>n_0$ 时 $c_1n^2 \le an^2+bn+c \le c_2n^2$ 成立。

2.1.1 同阶函数集合

例3 证明 $6n^3 \neq \theta(n^2)$

证. 如果存在c_1、$c_2>0$，n_0使得当$n \geq n_0$时，

$c_1 n^2 \leq 6n^3 \leq c_2 n^2$， 即 $c_1 \leq 6n \leq c_2$，$n \leq c_2/6$。

于是，当$n>c_2/6$时与$n \leq c_2/6$矛盾。

命题2.1.1：对于任意正整数d和任意常数$a_d>0$，有：

$$P(n)=\sum_{i=0}^{d}a_i n^i=\theta(n^d)$$

证． 由于 $\sum_{i=0}^{d}a_i n^i \le (d+1)\max\{a_i\}n^d=O(n^d)$， $\sum_{i=0}^{d}a_i n^i \ge a_d n^d=\Omega(n^d)$， 所以

$P(n)=\sum_{i=0}^{d}a_i n^i=\theta(n^d)$。

如果$f(n)=O(n^k)$，则称$f(n)$是多项式界限的。

2.1.2 低阶函数集合

定义2.1.2. 设$f(n)$和$g(n)$是正值函数。如果$\exists c>0$, n_0, $\forall n>n_0, f(n)\leq cg(n)$，则称$f(n)$比$g(n)$低阶或$g(n)$是$f(n)$的上界，记作$f(n)=O(g(n))$。

$O(g(n))$可以视为如下集合：

$$\{f(n) \mid \exists c,\ n_0,\ \forall n>n_0, f(n)\leq cg(n)\}$$

称为所有比$g(n)$低阶的函数的集合

2.1.2 低阶函数集合

例 1　$\theta(g(n)) = f(n) \Rightarrow f(n) = O(g(n))$

$\theta(g(n)) \subseteq O(g(n))$

例 2 证明 $n=O(n^2)$.

证. 令 $c=1$, $n_0=1$, 则当 $n \geq n_0$ 时， $0 \leq n \leq cn^2$ 。

2.1.3 高阶函数集合

定义2.1.3. 设$f(n)$和$g(n)$是正值函数。如果$\exists c>0,\ n_0,\ \forall n>n_0, f(n)\geq cg(n)$，则称$f(n)$比$g(n)$高阶或$g(n)$是$f(n)$的下界，记作$f(n)=\Omega(g(n))$。

$\Omega(g(n))$可以视为如下集合：

$\{f(n) \mid \exists c,\ n_0,\ \forall n>n_0, f(n)\geq cg(n)\}$

称为所有比$g(n)$高阶的函数的集合

定理 2.1.对于任意 f(n)和 g(n)，$f(n)=\theta(g(n))$ iff $f(n)=O(g(n))$ 而且 $f(n)=\Omega(g(n))$.

证. $\Rightarrow$ 如果 $f(n)=\theta(g(n))$，则 $\exists c_1, c_2>0, n_0>0$，当 $n \geq n_0$ 时，

$$c_1 g(n) \leq f(n) \leq c_2 g(n).$$

显然 $f(n)=\Omega(g(n))$ and $f(n)=O(g(n))$.

$\Leftarrow$ 如果 $f(n)=O(g(n))$ 且 $f(n)=\Omega(g(n))$，则由 $f(n)=O(g(n))$ 可知，存在 $c_1, n_1 \geq 0$，使得，当 $n \geq n_1$，$f(n) \leq c_1 g(n)$。

由 $f(n)=\Omega(g(n))$ 可知，$\exists c_2, n_2 \geq 0$，使得当 $n \geq n_1$，$f(n) \leq c_2 g(n)$

令 $n_0 \geq \max\{n_1, n_2\}$，则当 $n \geq n_0$，$c_2 f(n) \leq f(n) \leq c_1 g(n)$。

HIT CS&E

2.1.4 严格低阶函数集合

定义2.1.4. 设$f(n)$和$g(n)$是正值函数。如果$\forall c>0$, n_0, $\forall n>n_0$, $f(n)<cg(n)$，则称$f(n)$严格比$g(n)$低阶或$g(n)$是$f(n)$的严格上界，记作$f(n)=o(g(n))$。

$o(g(n))$可以视为如下集合：

$\{f(n) \mid \forall c,\ n_0,\ \forall n>n_0, f(n)<cg(n)\}$

称为所有比$g(n)$严格低阶的函数的集合

例 1．证明 $2n = o(n^2)$

证．对 $\forall c > 0$，欲 $2n < cn^2$，必 $2 < cn$，即 $\frac{2}{c} < n$。所以，当 $n_0 = \frac{2}{c}$ 时，$2n < cn^2$ 对 $\forall c > 0$，$n \geq n_0$。

例 2．证明 $2n^2 \neq o(n^2)$

证．当 $c=1>0$，对于任何 n_0，当 $n \geq n_0$，$2n^2 < cn^2$ 都不成立

命题2.1.2 $f(n)=o(g(n)) \Rightarrow \lim_{n\to\infty}\frac{f(n)}{g(n)}=0$

证. 由于 $f(n)=o(g(n))$，对任意$\varepsilon>0$，存在 n_0，当 $n \ge n_0$ 时，

$$0 \le f(n) < \varepsilon g(n),$$

即 $0 \le \frac{f(n)}{g(n)} \le \varepsilon$． 于是，$\lim_{n\to\infty}\frac{f(n)}{g(n)}=0$．

HIT
CS&E

2.1.5 严格高阶函数集合

定义2.1.4. 设$f(n)$和$g(n)$是正值函数。如果$\forall c>0,\ n_0,\ \forall n>n_0, f(n)>cg(n)$，则称$f(n)$严格比$g(n)$高阶或$g(n)$是$f(n)$的严格下界，记作$f(n)=\omega(g(n))$。

$\omega(g(n))$可以视为如下集合：

$\{f(n) \mid \forall c,\ n_0,\ \forall n>n_0, f(n)>cg(n)\}$

称为所有比$g(n)$严格高阶的函数的集合

命题 2.1.3 $f(n) \in w(g(n))$ iff $g(n) \in o(f(n))$.

证:

$\Rightarrow$ 对 $\forall c > 0$， $1/c>0$. 由 $f(n) \in w(g(n))$ 知， 对 $1/c>0$, $\exists n_0$, 当 $n \geq n_0$ 时, $(1/c)g(n)<f(n)$， 即 $g(n)<cf(n)$. 于是， $g(n) \in o(f(n))$.

$\Leftarrow$ 对于任意 $c>0$， $1/c>0$. 由 $g(n) \in o(f(n))$ 可知, $\exists n_0 \geq 0$， 当 $n > n_0$ 时, $g(n)<(1/c)f(n)$， 即 $cg(n)<f(n)$. 于是， $f(n) \in w(g(n))$。

命题2.1.4 $f(n)=w(g(n))\Rightarrow\lim\limits_{n\to\infty}\dfrac{f(n)}{g(n)}=\infty$

证：对 $\forall\, c>0$，由于 $f(n)=w(g(n))$，必存在 n_0，使得当 $n\geq n_0$ 时，$f(n)>cg(n)$，即当 $n\geq n_0$ 时，$f(n)/g(n)>c$. 于是，$\lim\limits_{n\to\infty}\dfrac{f(n)}{g(n)}=\infty$.

2.1.6 函数阶的性质

A 传递性:

（a） $f(n)=\theta(g(n))\wedge g(n)=\theta(h(n))\Rightarrow f(n)=\theta(h(n))$

（b） $f(n)=O(g(n))\wedge g(n)=O(h(n))\Rightarrow f(n)=O(h(n))$

（c） $f(n)=\Omega(g(n))\wedge g(n)=\Omega(h(n))\Rightarrow f(n)=\Omega(h(n))$

（d） $f(n)=o(g(n))\wedge g(n)=o(h(n))\Rightarrow f(n)=o(h(n))$

（e） $f(n)=w(g(n))\wedge g(n)=w(h(n))\Rightarrow f(n)=w(h(n))$.

HIT
CS&E

2.1.6 函数阶的性质(续)

B 自反性：

（a） $f(n) = \theta(f(n))$，

（b） $f(n) = \mathrm{O}(f(n))$，

（c） $f(n) = \Omega(f(n))$.

C 对称性

$f(n) = \theta(g(n))$ *iff* $g(n) = \theta(f(n))$.

D 反对称性：

$f(n) = O(g(n))$ *iff* $g(n) = \Omega(f(n))$

$f(n) = o(g(n))$ *iff* $g(n) = w(f(n))$

问题

所有函数都是可比的吗？？？

$f(n) = n$ 与 $g(n) = n^{1+\sin(n)}$ 可比吗？

2.2 标准符号和通用函数

- Flour 和 ceiling
- 多项式

2.2.1 Flour和ceiling

定义2.2.1(Flours和ceiling). $\lfloor x \rfloor$表示小于或等于x的最大整数. $\lceil x \rceil$表示大于等于x的最小整数.

命题 2.2.1 $x-1<\lfloor x \rfloor \le x \le \lceil x \rceil < x+1$

HIT
CS&E

命题 2.2.2 对于任意整数n, $\lceil n/2 \rceil + \lfloor n/2 \rfloor = n$

证．若 $n = 2k$,则$\left\lceil \frac{n}{2} \right\rceil = k$， $\lfloor n/2 \rfloor = k$．于是 $\left\lceil \frac{n}{2} \right\rceil + \lfloor n/2 \rfloor = 2k = n$

若 n=2k+1， 则$\left\lceil \frac{n}{2} \right\rceil + \lfloor n/2 \rfloor = k+1+k = 2k+1 = n$.

命题 2.2.3 设 n、a、b是任意整数， $a \neq 0, b \neq 0$， 则

(1) $\lceil \lceil n/a \rceil / b \rceil = \lceil n/ab \rceil$。

(2) $\lfloor \lfloor n/a \rfloor / b \rfloor = \lfloor n/ab \rfloor$

证．(1) 若 $n = kab$,则$\left\lceil \frac{n/a}{b} \right\rceil = \left\lceil \frac{kb}{b} \right\rceil = k = \left\lceil \frac{kab}{ab} \right\rceil = \left\lceil \frac{n}{ab} \right\rceil$。

若 $n = kab + \alpha$， $0 < \alpha < ab$， 则

$$\left\lceil \left\lceil \frac{n}{a} \right\rceil \Big/ b \right\rceil = \left\lceil \frac{kb + \lceil \alpha/a \rceil}{b} \right\rceil = k+1 = \left\lceil \frac{n}{ab} \right\rceil = \left\lceil \frac{kab+\alpha}{ab} \right\rceil = k+1$$

(2) 类似于(1)的证法。

2.3 和式的估计与界限

- 线性和
- 级数
- 直接求和的界限

2.3 和式的估计与界限

1. 线性和

命题 2.3.1 $$\sum_{k=1}^{n}(ca_k+b_k)=c\sum_{k=1}^{n}a_k+\sum_{k=1}^{n}b_k$$

命题 2.3.2 $$\sum_{k=1}^{n}\theta(f(k))=\theta\left(\sum_{k=1}^{n}f(k)\right)$$

证. 对 n 用数学归纳法证明。

当 $n=1$ 时，$\theta(f(1))=\theta(f(1))$ 显然成立。假设 $n\le m$ 时成立。

令 $n=m+1$，则
$$\sum_{k=1}^{m+1}\theta(f(k))=\sum_{k=1}^{m}\theta(f(k))+\theta(f(m+1))$$
$$=\theta\left(\sum_{k=1}^{m}f(k)\right)+\theta(f(m+1))$$
$$=\theta\left(\sum_{k=1}^{m}f(k)+f(m+1)\right)$$
$$=\theta\left(\sum_{k=1}^{m+1}f(k)\right)。$$

2. 级数

命题 2.3.3 $$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \theta(n^2)$$

命题 2.3.4 $$\sum_{k=0}^{n} x^k = 1 + x + x^2 + \ldots\ldots + x^n = \frac{x^{n+1} - 1}{x - 1} \qquad (x \neq 1)$$

$$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x} \qquad |x| < 1$$

命题 2.3.5 $$H_n = \sum_{k=1}^{n} \frac{1}{k} = \ln n + O(1)$$

命题 2.3.6 $$\sum_{k=1}^{n}(a_k - a_{k-1}) = a_n - a_0 .$$

命题 2.3.7 $$\sum_{k=0}^{n-1}(a_k - a_{k+1}) = a_0 - a_n$$

命题 2.3.8 $$\sum_{k=1}^{n-1}\frac{1}{k(k+1)} = \sum_{k=1}^{n-1}\left(\frac{1}{k} - \frac{1}{k+1}\right) = 1 - \frac{1}{n}$$

命题 2.3.9 $$\lg(\prod_{k=1}^{n} a_k) = \sum_{k=1}^{n} \lg a_k$$

3. 直接求和的界限

例1. $\sum_{k=1}^{n} k \le \sum_{k=1}^{n} n = n^2$

例 2. $\sum_{k=1}^{n} a_i \le n \times \max\{a_k\}$.

例 3. 设对于所有 $k \ge 0$, $a_{k+1}/a_k \le r < 1$，求 $\sum_{k=0}^{n} a_k$ 的上界.

解: $a_1/a_0 \le r \Rightarrow a_1 \le a_0 r$,

$a_2/a_1 \le r \Rightarrow a_2 \le a_1 r \le a_0 r^2$,

$a_3/a_2 \le r \Rightarrow a_3 \le a_2 r \le a_0 r^3$ ……

$a_k/a_{k-1} \le r \Rightarrow a_k \le a_{k-1} r \le a_0 r^k$

于是, $\sum_{k=0}^{n} a_k \le \sum_{k=0}^{\infty} a_0 r^k = a_0 \sum_{k=0}^{\infty} r^k = \dfrac{a_0}{1-r}$.

HIT
CS&E

例 4. 求 $\sum_{k=1}^{\infty}\left(k/3^k\right)$的界

解. 使用例 3 的方法. $\frac{k+1}{3^{k+1}}\Big/\frac{k}{3^k}=\frac{1}{3}\cdot\frac{k+1}{k}=\frac{1}{3}\left(1+\frac{1}{k}\right)\le\frac{2}{3}=r$. 于是

$$\sum_{k=1}^{\infty}\frac{k}{3^k}\le\sum_{k=1}^{\infty}a_1r^k=\sum_{k=1}^{\infty}\frac{1}{3}\left(\frac{2}{3}\right)^k=\frac{1}{3}\cdot\frac{1}{1-\frac{2}{3}}=1.$$

例 5. 用分裂和的方法求 $\sum_{k=1}^{n}k$ 的下界.

解: $\sum_{k=1}^{n}k=\sum_{k=1}^{n/2}k+\sum_{k=n/2+1}^{n}k\ge\sum_{k=1}^{n/2}0+\sum_{k=n/2+1}^{n}n/2\ge\left(\frac{n}{2}\right)^2=\Omega\left(n^2\right)$.

HIT
CS&E

例 6. 求 $\sum_{k=0}^{\infty}\frac{k^2}{2^k}$ 的上界

解：当 $k\geq 3$ 时，$\dfrac{(k+1)^2 / 2^{k+1}}{k^2 / 2^k}=\dfrac{(k+1)^2}{2k^2}\leq\dfrac{8}{9}$

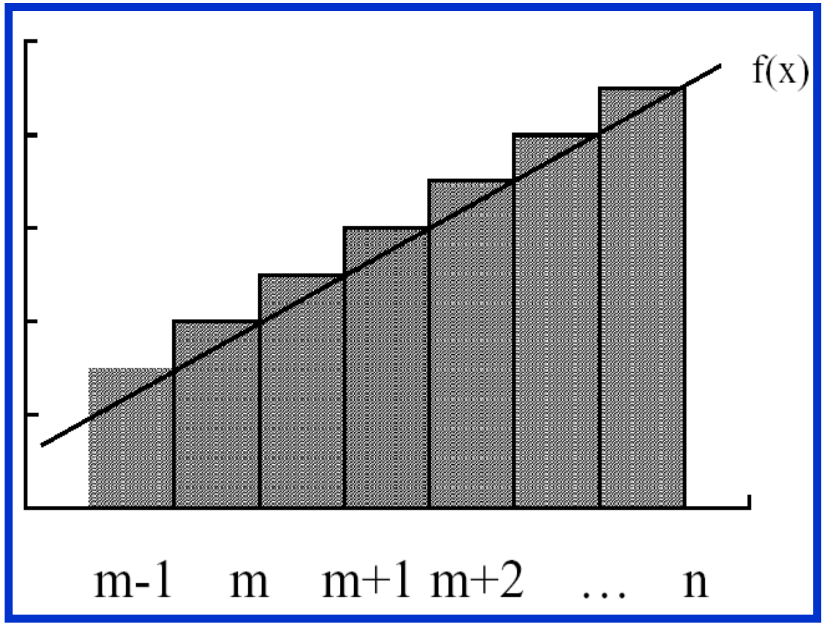
于是 $\sum_{k=0}^{\infty}\frac{k^2}{2^k}=\sum_{k=0}^{2}\frac{k^2}{2^k}+\sum_{k=3}^{\infty}\frac{k^2}{2^k}\leq\theta(1)+\sum_{k=3}^{\infty}\frac{9}{8}\cdot\left(\frac{8}{9}\right)^k=O(1)$.

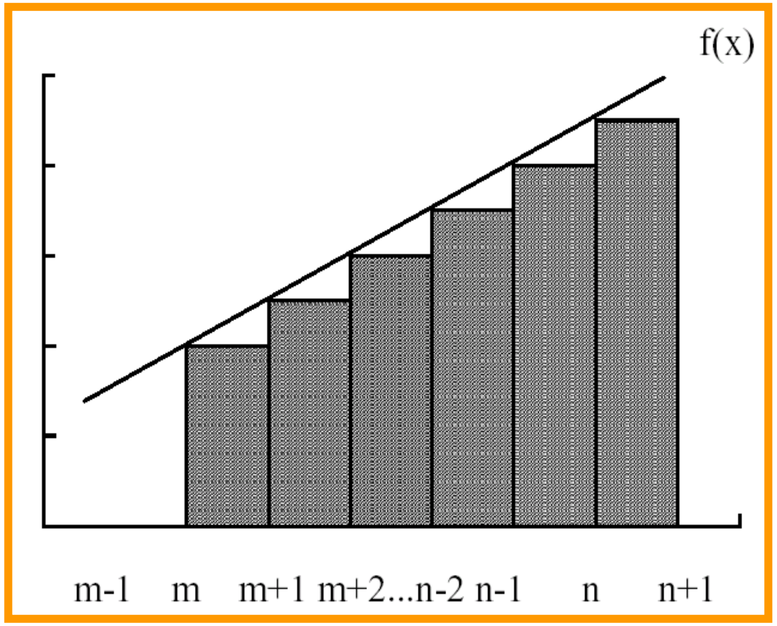
例 7. 求 $H_n = \sum_{k=1}^{n} \frac{1}{k}$ 的上界

解：
$$\sum_{k=1}^{n} \frac{1}{k} = \frac{1}{1} + \left(\frac{1}{2} + \frac{1}{3}\right) + \left(\frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7}\right)$$
$$+\left(\frac{1}{8} + \frac{1}{9} + \frac{1}{10} + \frac{1}{11} + \frac{1}{12} + \frac{1}{13} + \frac{1}{14} + \frac{1}{15}\right) + \cdots\cdots$$
$$\le \sum_{i=1}^{\lceil \log n \rceil} \sum_{j=0}^{2^i-1} \frac{1}{2^i + j} \le \sum_{i=0}^{\lceil \log n \rceil} \sum_{j=0}^{2^i-1} \frac{1}{2^i} \le \sum_{i=1}^{\lceil \log n \rceil} 1 \le \log n + 1 = O(\log n)$$

HIT
CS&E

例 8. 如果 $f(k)$ 单调递增，则 $\int_{m-1}^{n} f(x)dx \le \sum_{k=m}^{n} f(k) \le \int_{m}^{n+1} f(x)dx$.





例 9．当 $f(x)$ 单调递减时，$\int_m^{n+1} f(x)dx \le \sum_{k=m}^{n} f(k) \le \int_{m-1}^{n} f(x)dx$.

例 10． $\sum_{k=1}^{n} \frac{1}{k} \ge \int_1^{n+1} \frac{dx}{x} = \ln(n+1)$ ， $\sum_{k=2}^{n} \frac{1}{k} \le \int_1^{n} \frac{dx}{x} = \ln n$.

2.4 递归方程

- 递归方程的定义
- Substitution方法
- Iteration方法
- Master方法

2.4.1 何为递归方程

- 递归方程：递归方程是使用具有小输入值的相同方程来描述一个方程.

用自身来定义自身

- 递归方程例：Merge-sort排序算法的复杂性方程

$$T(n)=\theta(1) \qquad \text{if } n=1$$

$$T(n)=2T(n/2)+\theta(n) \qquad \text{if } n>1.$$

$T(n)$的解是$\theta(n\log n)$

边界条件是根据问题的不同而不同的！

求解递归方程的三个主要方法

- Substitution方法:
  - Guess first,
  - 然后用数学归纳法证明.
- Iteration方法:
  - 把方程转化为一个和式
  - 然后用估计和的方法来求解.
- Master方法:
  - 求解型为$T(n)=aT(n/b)+f(n)$的递归方程

Substitution方法I：联想已知的$T(n)$

例1. 求解$2T(n/2 + 17) + n$

解：猜测：$T(n)=2T\left(\frac{n}{2}+17\right)+n$与$T(n)=2T\left(\frac{n}{2}\right)+n$只相差一个17.

当n充分大时$T\left(\frac{n}{2}+17\right)$与$T\left(\frac{n}{2}\right)$的差别并不大，因为

$\frac{n}{2}+17$与$\frac{n}{2}$相差小．我们可以猜$T(n)=O(n\lg n)$.

证明：用数学归纳法

Substitution方法II：猜测上下界，减少不确定性范围

例 3．求解 $T(n)=2T\left(\frac{n}{2}\right)+n$．

解：首先证明 $T(n)=\Omega(n)$，$T(n)=O(n^2)$

然后逐阶地降低上界、提高下界。

$\Omega(n)$的上一个阶是$\Omega(n\log n)$
$O(n^2)$的下一个阶是$O(n\log n)$

细微差别的处理

- 问题：猜测正确，数学归纳法的归纳步似乎证不出来
- 解决方法：从guess中减去一个低阶项，可能work.

例 4．求解 $T(n)=T(\lfloor n/2 \rfloor)+T\left(\left\lceil \frac{n}{2} \right\rceil\right)+1$

解：(1) 我们猜 $T(n)=O(n)$

证：$T(n)\le c\lfloor n/2 \rfloor + c\lceil n/2 \rceil + 1 = cn+1 \ne cn$

证不出 $T(n)=O(cn)$

(2) 减去一个低阶项，猜 $T(n)\le cn-b$， $b\ge 0$ 是常数

证：设当 $\le n-1$ 时成立

$$T(n)=T(\lfloor n/2 \rfloor)+T\left(\left\lceil \frac{n}{2} \right\rceil\right)+1 \le c\left\lfloor \frac{n}{2} \right\rfloor - b + c\left\lceil \frac{n}{2} \right\rceil - b + 1$$

$$= cn-2b+1 = cn-b-b+1 \le cn-b$$ （只要 $b\ge 1$）。

HIT
CS&E

避免陷阱

例 5. 求解 $T(n)=2T(\lfloor n/2 \rfloor)+n$。

解： 猜 $T(n)=O(n)$

证： 用数学归纳法证明 $T(n)\le cn$。

$T(n)\le 2(c\lfloor n/2 \rfloor)+n\le cn+n=O(n)$

$\boxed{\text{--错！！}}$

错在哪里：过早地使用了*O(n)*而陷入了陷阱

应该在证明*T(n)*≤*cn*才可用

从T(n)≤cn+n不可能得到*T(n)*≤*cn*

因为对于任何c>0, 我们都得不到*cn*+*n* ≤*cn*

Substitution方法III：变量替换

经变量替换把递归方程变换为熟悉的方程.

例 6. 求解 $T(n) = 2T(\sqrt{n}) + \lg n$

解：令 $m = \lg n$，则 $n = 2^m$，$T(2^m) = 2T\left(2^{\frac{m}{2}}\right) + m$.

令 $S(m) = T(2^m)$ 则 $T\left(2^{\frac{m}{2}}\right) = S\left(\frac{m}{2}\right)$. 于是，$S(m) = 2S\left(\frac{m}{2}\right) + m$.

显然, $S(m) = O(m\lg m)$,即 $T(2^m) = O(m\lg m)$

由于 $2^m=n$, $m=\log n$, $T(n) = O(\lg n \times \lg(\lg n))$.

2.4.3 Iteration方法

方法：

循环地展开递归方程，
把递归方程转化为和式，
然后可使用求和技术解之。

例 1. $T(n)=n+3T\left(\left\lfloor n/4 \right\rfloor\right)$，**$T(1)=1$**

$$
\begin{aligned}
&= n+3\left(\left\lfloor n/4 \right\rfloor+3T\left(\left\lfloor n/16 \right\rfloor\right)\right)\\
&= n+3\left(\left\lfloor n/4 \right\rfloor+3\left(\left\lfloor \frac{n}{16} \right\rfloor+3T\left(\left\lfloor n/64 \right\rfloor\right)\right)\right)\\
&= n+3\left\lfloor n/4 \right\rfloor+9\left\lfloor \frac{n}{16} \right\rfloor+27T\left(\left\lfloor n/64 \right\rfloor\right)\\
&= n+3\left\lfloor n/4 \right\rfloor+3^2\left\lfloor \frac{n}{4^2} \right\rfloor+3^3\left(\left\lfloor \frac{n}{4^3} \right\rfloor\right)+\ldots\ldots+3^i\,T\left(\left\lfloor n/4^i \right\rfloor\right)
\end{aligned}
$$

$$\boxed{\text{令}\ \frac{n}{4^i}=1 \Rightarrow 4^i=n \Rightarrow i=\log_4 n}$$

$$
\begin{aligned}
&= n+3\left\lfloor n/4 \right\rfloor+3^2\left\lfloor \frac{n}{4^2} \right\rfloor+3^3\left(\left\lfloor \frac{n}{4^3} \right\rfloor\right)+\ldots\ldots+3^{\log_4 n}\,T\left(\left\lfloor 1 \right\rfloor\right)\\
&\le \sum_{i=0}^{\log_4 n} 3^i\, n/4^i \ \cdot\ \le n\sum_{i=0}^{\infty}\left(\frac{3}{4}\right)^i = n\times\frac{1}{1-\dfrac{3}{4}} = 4n = O(n)
\end{aligned}
$$

Master定理解析

目的：求解 $T(n)=aT\left(n/b\right)+f(n)$ 型方程，$a\ge 1, b>1$ 是常数，$f(n)$ 是正函数

一般的分治递归：把问题分成一些更小(或许有重叠)的子问题，递归地求解这些子问题，然后用所得到的子问题的解去求解原始问题。

$T(n)=aT\left(n/b\right)+f(n)$：将一个大小为**n**的问题分成大小为**n/b**的**a**个子问题，递归地求解这些子问题，然后用所得到的子问题的解以***f*(n)**的代价求解原始问题。

HIT
CS&E

Master 定理

定理 2.4.1 设$a\geq 1$和$b>1$是常数，$f(n)$是一个函数，$T(n)$是定义在非负整数集上的函数$T(n)=aT\left(n/b\right)+f(n)$. $T(n)$可以如下求解：

(1). 若 $f(n)=O\left(n^{\log_b a-\varepsilon}\right)$, $\varepsilon>0$是常数，则 $T(n)=\theta\left(n^{\log_b a}\right)$.

(2). 若 $f(n)=\theta\left(n^{\log_b a}\right)$，则 $T(n)=\theta\left(n^{\log_b a}\lg n\right)$.

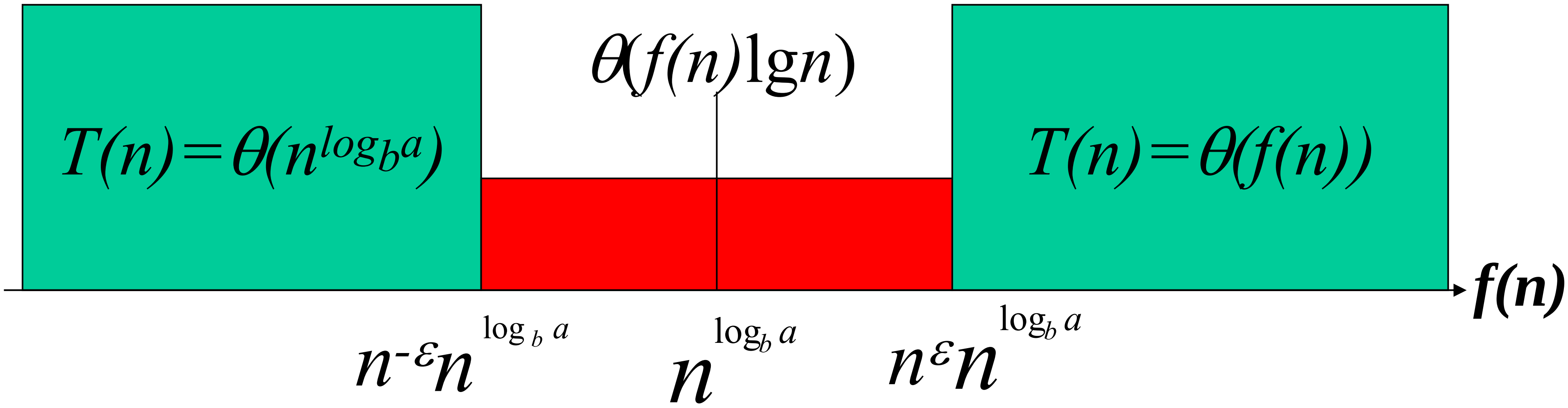
(3). 若 $f(n)=\Omega\left(n^{\log_b a+\varepsilon}\right)$, $\varepsilon>0$是常数，且对于所有充分大的n $af\left(n/b\right)\leq cf(n)$，$c$<1是常数，则 $T(n)=\theta(f(n))$.

*直观地：我们用 $f(n)$ 与 $n^{\log_b a}$ 比较

(1). 若 $n^{\log_b a}$ 大，则 $T(n)=\theta\left(n^{\log_b a}\right)$

(2). 若 $f(n)$ 大，则 $T(n)=\theta(f(n))$

(3). 若 $f(n)$ 与 $n^{\log_b a}$ 同阶，则 $T(n)=\theta\left(n^{\log_b a}\lg n\right)=\theta(f(n)\lg n)$.



对于红色部分，Master定理无能为力

更进一步：

(1). 在第一种情况，$f(n)$不仅小于$n^{\log_b a}$，必须多项式地小于，即对于一个常数$\varepsilon>0$，$f(n)=O(\frac{n^{\log_b a}}{n^{\varepsilon}})$.

(2). 在第三种情况，$f(n)$不仅大于$n^{\log_b a}$，必须多项式地大于，即对一个常数$\varepsilon>0$，$f(n)=\Omega(n^{\log_b a}\cdot n^{\varepsilon})$.

Master定理的使用

例 1. 求解 $T(n) = 9T\left(\frac{n}{3}\right) + n$.

解： $a = 9$， $b = 3$， $f(n) = n$ ， $n^{\log_b a} = \theta\left(n^2\right)$

$\because f(n) = n = O\left(n^{\log_b a - \varepsilon}\right)$ ， $\varepsilon = 1$

$\therefore T(n) = \theta\left(n^{\log_b a}\right) = \theta\left(n^2\right)$

例 2. 求解 $T(n) = T\left(\frac{2n}{3}\right) + 1$.

解： $a = 1$， $b = \left(\frac{3}{2}\right)$， $n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1$,

$f(n) = 1 = \theta(1) = \theta\left(n^{\log_b a}\right)$, $T(n) = \theta\left(n^{\log_b a} \lg n\right) = \theta(\lg n)$

HIT
CS&E

例 3. 求解 $T(n)=3T\left(n/4\right)+n\lg n$

解: $a=3$, $b=4$, $f(n)=n\lg n$, $n^{\log_b a}=n^{\log_4 3}=O\left(n^{0.793}\right)$

(1) $f(n)=n\lg n\geq n=n^{\log_b a+\varepsilon}$, $\varepsilon\approx 0.2$

(2) 对所有 **n**, $af\left(n/b\right)=3\times\frac{n}{4}\lg\frac{n}{4}=\frac{3}{4}n\lg\frac{n}{4}\leq\frac{3}{4}n\lg n=cf(n)$, $c=\frac{3}{4}$.

于是, $T(n)=\theta(f(n))=\theta(n\lg n)$

例 4. 求解 $T(n)=2T\left(n/2\right)+n\lg n$.

解: $a=2$, $b=2$, $f(n)=n\lg n$, $n^{\log_b a}=n$. $f(n)=n\lg n$ 大于 $n^{\log_b a}=n$, 但不是多项式地大于，Master定理不适用于该$T(n)$

Master定理的证明

$$T(n)=aT(n/b)+f(n)$$

- 证明思路
  - 首先，在$n=b^k$的情况下证明Master定理
  - 然后，在$n\neq b^k$的情况下证明Master定理

HIT
CS&E

当$n=b^k$时Master定理的证明

- 第一步：证明

$$T(n) = \theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n-1} a^j f(\frac{n}{b^j})$$

- 第二步：求解

$$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(\frac{n}{b^j})$$

- 第三步：完成Master定理的证明

第一步

引理 1：设 a≥1, b>1, $n=b^k$, k 是正整数，　则方程

$$T(n)=\begin{cases}\theta(1) & if \quad n=1\\ aT(n/b)+f(n) & if \quad n=b^k\end{cases}$$

的解为:

$$T(n)=\theta(n^{\log_b a})+\sum_{j=0}^{\log_b n-1}a^j f(n/b^j)$$

证明:
$$\begin{aligned}T(n)&=f(n)+aT(n/b)\\&=f(n)+af(n/b)+a^2T(n/b^2)\\&=f(n)+af(n/b)+a^2f(n/b^2)+a^3T(n/b^3)+\ldots\\&\quad+a^{\log_b n-1}f(n/b^{\log_b n-1})+a^{\log_b n}T(n/b^{\log_b n})\end{aligned}$$

由于 $a^{\log_b n}=n^{\log_b a}$，$a^{\log_b n}T(n/b^{\log_b n})=a^{\log_b n}T(1)=\theta(n^{\log_b a})$.

于是 $T(n)=\theta(n^{\log_b a})+\sum_{j=0}^{\log_b n-1}a^j f(n/b^j)$.

第三步

引理 2: 设 a≥1, b>1, $n=b^k$, k 是正整数， $g(n)=\sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$, 则

(1) if $f(n)=O(n^{\log_b a-\varepsilon})$ for $\varepsilon>0$, 则 $g(n)=O(n^{\log_b a})$

(2) if $f(n)=\theta(n^{\log_b a})$, then $g(n)=\theta(n^{\log_b a}\lg n)$

(3) if $af(n/b)\le cf(n)$ for some $0<c<1$ and all $n\ge b$, then $g(n)=\theta(f(n))$.

证明引理2 $g(n)=\sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$

(1) if $f(n)=O(n^{\log_b a-\varepsilon})$ for $\varepsilon>0$，则 $g(n)=O(n^{\log_b a})$

证明：

$$g(n) = O\left(\sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a-\varepsilon}\right)$$

$$\sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a-\varepsilon} = n^{\log_b a-\varepsilon}\sum_{j=0}^{\log_b n-1}\left(\frac{ab^{\varepsilon}}{b^{\log_b a}}\right)^j$$

$$= n^{\log_b a-\varepsilon}\sum_{j=0}^{\log_b n-1}(b^{\varepsilon})^j$$

$$= n^{\log_b a-\varepsilon}\left(\frac{b^{\varepsilon\log_b n}-1}{b^{\varepsilon}-1}\right)$$

$$= n^{\log_b a-\varepsilon}\left(\frac{n^{\varepsilon}-1}{b^{\varepsilon}-1}\right) = n^{\log_b a-\varepsilon}O(n^{\varepsilon})$$

$$g(n)=O(n^{\log_b a-\varepsilon}O(n^{\varepsilon}))=O(n^{\log_b a})$$

$$g(n)=\sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$$

$$\boxed{(2)\quad \text{if}\quad f(n)=\theta(n^{\log_b a}),\ \text{then}\quad g(n)=\theta(n^{\log_b a}\lg n)}$$

证明:

(2) 由于 $f(n/b^j)=\theta((n/b^j)^{\log_b a})$, $g(n)=\theta(\sum_{j=0}^{\log_b n-1} a^j(\frac{n}{b^j})^{\log_b a})$.

$$\sum_{j=0}^{\log_b n-1} a^j(\frac{n}{b^j})^{\log_b a}=n^{\log_b a}\sum_{j=0}^{\log_b n-1}(\frac{a}{b^{\log_b a}})^j=n^{\log_b a}\sum_{j=0}^{\log_b n-1}1=n^{\log_b a}\log_b n$$

$$\Rightarrow g(n)=\theta(n^{\lg_b a}\log_b n)=\theta(n^{\log_b a}\lg n).$$

(3) if $af(n/b) \le cf(n)$ for some 0<c<1 and all n≥b, then $g(n) = \theta(f(n))$.

(3) g(n)中的所有项皆为正. 由于对于 0<c<1 和 all n≥b,

$af(n/b) \le cf(n)$,

$af(\frac{n}{b^2}) \le cf(\frac{n}{b})$,

$af(\frac{n}{b^3}) \le cf(\frac{n}{b^2})$,

…

$af(\frac{n}{b^j}) \le cf(\frac{n}{b^{j-1}})$

f(n)是非负函数

$$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$$

g(n)≥f(n) ⇒ g(n)=Ω(f(n))

g(n)=θ(f(n))

我们有 $a^j f(n/b) \cdots \cancel{f(n/b^{j-1})} f(n/b^j) \le c^j f(n) f(n/b) \cdots \cancel{f(n/b^{j-1})}$

$$\Rightarrow a^j f(\frac{n}{b^j}) \le c^j f(n)$$

于是,

$$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(\frac{n}{b^j}) \le \sum_{j=0}^{\log_b n-1} c^j f(n) \le f(n)\sum_{j=0}^{\infty} c^j = f(n)(\frac{1}{1-c}) = O(f(n))$$

第三步

引理 3: a≥1, b>1, $n=b^k$, k 为正整数，则

$$T(n)=\begin{cases}\theta(1) & if \quad n=1\\ aT(n/b)+f(n) & if \quad n=b^k\end{cases}$$

的解为:

(1) if $f(n)=O(n^{\log_b a-\varepsilon})$ for some $\varepsilon>0$, then $T(n)=\theta(n^{\log_b a})$

(2) if $f(n)=\theta(n^{\log_b a})$, then $T(n)=\theta(n^{\log_b a}\lg n)$

(3) if $f(n)=\Omega(n^{\log_b a+\varepsilon})$ for some $\varepsilon>0$, and if

$af(n/b)\le cf(n)$ for some 0<c<1 and all 充分大的 n, then $T(n)=\theta(f(n))$

证明：由引理1和引理2，可得：

$$T(n) = \theta(n^{\log_b a}) + O(n^{\log_b a}) = \theta(n^{\log_b a})$$

$$T(n) = \theta(n^{\log_b a}) + \theta(n^{\log_b a} \lg n) = \theta(n^{\log_b a} \lg n)$$

$$T(n) = \theta(n^{\log_b a}) + \theta(f(n)) = \theta(f(n))$$

当$n \neq b^k$时Master定理的证明

基本思想

- $aT(\left\lfloor \frac{n}{b} \right\rfloor) + f(n) \le aT(\frac{n}{b}) + f(n) \le aT(\left\lceil \frac{n}{b} \right\rceil) + f(n)$
- 求$T(n) = aT(\left\lceil \frac{n}{b} \right\rceil) + f(n)$的上界、$T(n) = aT(\left\lfloor \frac{n}{b} \right\rfloor) + f(n)$的下界可得到$T(n)$的界限。
- 求$T(n) = aT(\left\lfloor \frac{n}{b} \right\rfloor) + f(n)$的下界类似于求$T(n) = aT(\left\lceil \frac{n}{b} \right\rceil) + f(n)$的上界，所以我们只求$T(n) = aT(\left\lceil \frac{n}{b} \right\rceil) + f(n)$的上界

- 方法仍然是循环展开 $T(n) = aT(\left\lceil \frac{n}{b} \right\rceil) + f(n)$
- 需要处理序列：

$$n$$

$$\left\lceil \frac{n}{b} \right\rceil$$

$$\left\lceil \left\lceil \frac{n}{b} \right\rceil \Big/ b \right\rceil$$

$$\left\lceil \left\lceil \left\lceil \frac{n}{b} \right\rceil \Big/ b \right\rceil \Big/ b \right\rceil$$

…

处理多重Ceiling

定义：$n_i = \begin{cases} n & if \quad i = 0 \\ \lceil n_{i-1}/b \rceil & if \quad i > 0 \end{cases}$

引理 **4.** $n_0 \le n$，$n_1 \le \frac{n}{b} + 1$，$n_2 \le \frac{n}{b^2} + \frac{1}{b} + 1$，$n_3 \le \frac{n}{b^3} + \frac{1}{b^2} + \frac{1}{b} + 1$，

$$\ldots, \quad n_i \le \frac{n}{b^i} + \sum_{j=0}^{i-1} \frac{1}{b^j} \le \frac{n}{b^i} + \frac{b}{b-1} 。$$

证：由$\lceil x \rceil \le x + 1$可证。

引理 5：当$i=\lfloor \log_b n \rfloor$时，$n_i \le b+\dfrac{b}{b-1}=O(1)$

证：由于$n_i \le \dfrac{n}{b^i}+\dfrac{b}{b-1}$，我们有

$$n_{\lfloor \log_b n \rfloor} \le \frac{n}{b^{\lfloor \log_b n \rfloor}}+\frac{b}{b-1} \le \frac{n}{b^{(\log_b n)-1}}+\frac{b}{b-1}=b+\frac{b}{b-1}=O(1)\text{。}$$

证明的第一步

引理 6: $T(n)=aT(\left\lceil \frac{n}{b} \right\rceil)+f(n)\le\theta(n^{\log_b a})+\sum_{j=0}^{\lfloor\log_b n\rfloor-1}a^j f(n_j)$

证: $T(n)=f(n_0)+aT(n_1)=f(n_0)+af(n_1)+a^2T(n_2)$

$$\le f(n_0)+af(n_1)+a^2f(n_2)+\ldots+a^{\lfloor\log_b n\rfloor-1}f(n_{\lfloor\log_b n\rfloor-1})$$

$$+a^{\lfloor\log_b n\rfloor}T(n_{\lfloor\log_b n\rfloor})$$

$$=\theta(n^{\log_b a})+\sum_{j=0}^{\lfloor\log_b n\rfloor-1}a^j f(n_j)$$

证明的第二步

引理 7: $g(n)=\sum_{j=0}^{\lfloor \log_b n \rfloor -1} a^j f(n_j)$ 可以界限如下：

（1） If $f(n)=O(n^{\log_b a-\varepsilon})$ for $\varepsilon>0$ ， $g(n)=O(n^{\log_b a})$.

（2） If $f(n)=\theta(n^{\log_b a})$, then $g(n)=\theta(n^{\log_b a}\lg n)$.

（3） If $af(\lceil n/b \rceil)\le cf(n)$ for 0<c<1 and all 充分大的 n, then $g(n)=\theta(f(n))$.

证明：（3）由 $af(\lceil n/b \rceil) \le cf(n)$ 有:

$$af\left(\left\lceil \frac{n}{b} \right\rceil\right) \le cf(n) \Leftrightarrow af(n_1) \le cf(n_0)$$

$$af\left(\left\lceil \left\lceil \frac{n}{b} \right\rceil \Big/ b \right\rceil\right) \le cf\left(\left\lceil \frac{n}{b} \right\rceil\right) \Leftrightarrow af(n_2) \le cf(n_1)$$

$$\cdots$$

$$af\left(\left\lceil \cdots \left\lceil \frac{n}{b} \right\rceil \Big/ b \quad \cdots \right\rceil\right) \le cf\left(\left\lceil \cdots \left\lceil \frac{n}{b} \right\rceil \quad \cdots \right\rceil\right) \Leftrightarrow af(n_j) \le cf(n_{j-1})$$

$$\Rightarrow a^j f(n_1) \cdots f(n_{j-1}) f(n_j) \le c^j f(n_0) f(n_1) \cdots f(n_{j-1})$$

$$\Rightarrow a^j f(n_j) \le c^j f(n_0) = c^j f(n)$$

证明的其余部分与引理 2 的（3）的证明类似。

（2）只要证明 $f(n_j)=O(n^{\log_b a}/a^j)=O((\frac{n}{b^j})^{\log_b a})$，即可用引理 2 的（2）的证明完成本证明。

$$j\le\lfloor\log_b n\rfloor\Rightarrow b^j\le b^{\lfloor\log_b n\rfloor}=b^{\log_b n-\delta}=n\cdot\frac{1}{b^\delta}\qquad(0\le\delta<1)$$

$$\Rightarrow\frac{b^j}{n}\le\frac{1}{b^\delta}<1\ (\because b>1)\Rightarrow b^j/n<1\text{。}$$

由于 $f(n)=\theta(n^{\log_b a})$，$\exists c>0$，使对于充分大的 n_j，

$$f(n_j)\le cn_j^{\log_b a}\le c(\frac{n}{b^j}+\frac{b}{b-1})^{\log_b a}$$

$$=c(\frac{n}{b^j})^{\log_b a}(1+\frac{b^j}{n}\cdot\frac{b}{b-1})^{\log_b a}$$

$$\le c(\frac{n^{\log_b a}}{a^j})(1+\frac{b}{b+1})^{\log_b a}\qquad(\because\frac{b^j}{n}<1)$$

$$\le O(\frac{n^{\log_b a}}{a^j})\qquad(\because c(1+\frac{b}{b+1})^{\log_b a}\text{是常数})$$

于是（2）被证明。

(1) 只要证明 $f(n_j)=O((\frac{n}{b^j})^{\log_b a-\varepsilon})$，则本证明的其余部分与引理 2 的（1）相同。类似（2）可证明 $f(n_j)=O((\frac{n}{b^j})^{\log_b a-\varepsilon})$。

证明的第三步与 $n=b^k$ 的情况相同

至此，我们完成了 **Master** 定理的证明。

求解递归方程的其它方法

- 生成函数：算法平均情形分析时使用的一类方法.
  - 给定序列 $a_0, a_1, \ldots, a_k, \ldots$
  - 函数 $A(z) = \sum_{k\geq 0} a_k z^k$ 称为该序列的常规生成函数
  - 函数 $A(z) = \sum_{k\geq 0} a_k \frac{z^k}{k!}$ 称为该序列的指数生成函数
  - 概率生成函数
  - 二元生成函数（双下标序列 a_{uv}）
  - ……

HIT
CS&E

求解递归方程的其它方法

- 利用生成函数求解递归

给定一个表示某序列$\{a_n\}_{n\geq 0}$的递归,通常按以下步骤求解:

– 在递归式的两边乘以z^n，然后关于n求和

– 处理所得的和，得到一个关于生成函数的函数方程

– 解函数方程，得到生成函数的显式公式

– 将显式公式展开为一个幂级数，得到系数表达式，进而得到a_n

HIT CS&E

求解递归方程的其它方法

- 利用生成函数求解递归

例如求解常系数线性递归：T(n)=T(n-1)+1

– T(n), T(n-1),…, T(1), T(0)对应序列a_n, a_{n-1},…, a_1, a_0

– 递归式的两边乘以z^n，并关于n求和，得到

$$\sum_{n\ge1} a_n z^n = \sum_{n\ge1} a_{n-1} z^n + \frac{z}{1-z}$$

常规生成函数

$$A(z) = zA(z) + \frac{z}{1-z}$$

即：$A(z)=z/(1-z)^2$

幂级数展开，得到：$a_n=n$，即：T(n)=n

算法分析参考书：

***An Introduction to
the Analysis of Algorithms***

By Robert Sedgewick and Philippe Flajolet

本章总结

- 计算复杂性函数的阶
  - 同阶、低阶、高阶、严格低阶、严格高阶
  - 算法的复杂性与问题的复杂性
- 递归方程
  - 定义
  - 求解方法：替换法、迭代展开、master方法等