HIT
CS&E

第二章

算法分析的数学基础

高宏
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*

HIT
CS&E

提要

2.1 计算复杂性函数的阶
2.2 和式的计算与估计
2.3 递归方程

HIT
CS&E

提要

2.1 计算复杂性函数的阶
2.2 和式的计算与估计
2.3 递归方程

2.1 计算复杂性函数的阶

HIT CS&E

- 计算复杂性函数的阶
  - 算法执行时间增长的阶（增长率）
  - 执行时间函数的主导项

例如：

$T(n)=an^2+bn+c$

主导项: an^2

当输入大小n较大时，其它低阶项相对来说意义不大

系数a也相对来说意义不大

即：函数$T(n)$的阶为n^2

2.1 计算复杂性函数的阶

HIT CS&E

2.1.1 同阶函数

2.1.2 低阶函数

2.1.3 高阶函数

2.1.4 严格低阶函数

2.1.5 严格高阶函数

2.1.6 函数阶的性质

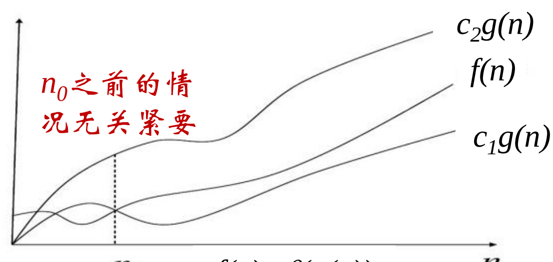
HIT CS&E

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$\exists c_1, c_2>0,\ n_0,\ \forall n>n_0,\ c_1g(n)\le f(n)\le c_2g(n)$，则称$f(n)$与$g(n)$同阶，记作$f(n)=\theta(g(n))$。

$\theta(g(n))$可以视为所有与$g(n)$同阶的函数集合：

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



$f(n)=\theta(g(n))$

HIT CS&E

2.1.1 同阶函数集合

- **例1，证明：$(1/3)n^2-3n= \theta(n^2)$**

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

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

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 $6n^3 \neq \theta(n^2)$

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。

HIT
CS&E

2.1.1 同阶函数集合

例 3 证明 $f(n) = an^2 + bn + c = \theta(n^2)$, $a>0$

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

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

令 $n_0 = 2 \cdot \max\left(\left(|b|/a\right), \sqrt{|c|/a}\right)$。当$n > n_0$时$c_1 n^2 \leq an^2 + bn + c \leq c_2 n^2$成立。

HIT
CS&E

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。

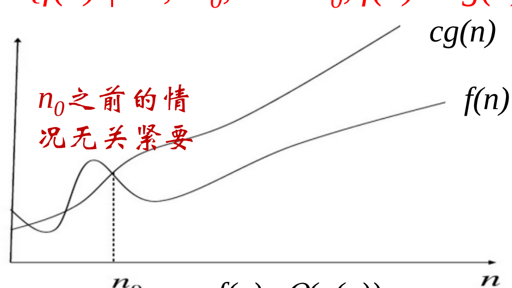
HIT
CS&E

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，$\forall n>n_0$, $f(n)\le cg(n)$，则称$f(n)$比$g(n)$低阶或$g(n)$是$f(n)$的上界，记作$f(n)=O(g(n))$。

$O(g(n))$可以视为所有比$g(n)$低阶的函数的集合：

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



$f(n)=O(g(n))$

12

HIT CS&E

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，$0 \leq n \leq cn^2$ 。

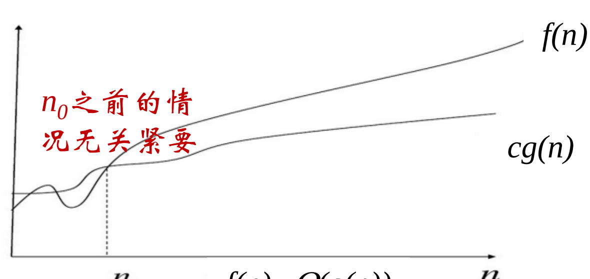
HIT CS&E

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$f(n)=\Omega(g(n))$。

$\Omega(g(n))$可以视为所有比$g(n)$高阶的函数集合：

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



$f(n)=\Omega(g(n))$

14

HIT CS&E

θ，O，Ω之间的关系

- θ 表示渐进紧界
- O 表示渐进上界
- Ω 表示渐进下界
- $\theta(g(n)) \subseteq O(g(n))$
- $f(n) = \theta(g(n)) \Rightarrow f(n) = O(g(n))$
- $f(n) = an^2+bn+c = \theta(n^2)$, $f(n) = O(n^2)$
- $an+b = O(n^2)$. 为什么？
- $n = O(n^2)$!!!

如果$f(n)=O(n^k)$, 则称$f(n)$是多项式界限的

15

HIT CS&E

θ，O，Ω之间的关系

- 一些讨论：
  - 当我们谈到插入排序的最坏运行时间是$O(n^2)$, 这个结论适用于所有的输入，即使对于已经排序的输入也成立，因为$O(n) \in O(n^2)$.
  - 然而插入排序的最坏运行时间$\theta(n^2)$不能应用到每个输入，因为对于已经排序的输入，$\theta(n) \neq \theta(n^2)$.

HIT CS&E

θ, O, Ω之间的关系

- Ω 用来描述运行时间的最好情况
- 对于插入排序，我们可以说
  - 最好运行时间是 $\Omega(n)$
  - 或者说，运行时间是$\Omega(n)$
  - 插入排序算法的运行时间在$\Omega(n)$和 $O(n^2)$之间
  - 插入排序算法的最坏运行时间是$\Omega(n^2)$
  - 但说插入排序算法的运行时间是$\Omega(n^2)$，是错误的！

极少用Ω来描述算法的运行时间和复杂性

17

HIT CS&E

θ, O, Ω之间的关系

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$n\ge n_0$时，

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

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$c_1,n_1\ge 0$,使得，当$n\ge n_1$， $f(n)\le c_1 g(n)$。由 $f(n)=\Omega(g(n))$可知，$\exists c_2,n_2\ge 0$,使得当$n\ge n_1$， $f(n)\le c_2 g(n)$
令 $n_0\ge \max\{n_1,n_2\}$,则当 $n\ge n_0$， $c_2 f(n)\le f(n)\le c_1 g(n)$。

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$\forall c>0, \exists n_0, \forall n>n_0, f(n)<cg(n)$，则称$f(n)$严格比$g(n)$低阶或$g(n)$是$f(n)$的严格上界，记作$f(n)=o(g(n))$。

$o(g(n))$可以视为所有比$g(n)$严格低阶的函数集合：

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

19

HIT CS&E

2.1.4 严格低阶函数集合

关于低阶O与严格低阶o的进一步说明

- O标记可能是或不是紧的
  - $2n^2 = O(n^2)$ 是紧的，但$2n = O(n^2)$不是紧的.
- o标记用于标记上界但不是紧的情况
  - $2n = o(n^2)$, 但是$2n^2 \neq o(n^2)$.
- 区别：某个正常数c在O标记中，但所有正常数c在o标记中.

20

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 $c=1>0$ 时，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 $n \geq n_0$ 时，

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

即 $0 \leq \frac{f(n)}{g(n)} \leq \varepsilon$. 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$\forall c>0, \exists n_0, \forall n>n_0, f(n)>cg(n)$，则称$f(n)$严格比$g(n)$高阶或$g(n)$是$f(n)$的严格下界，记作$f(n)=\omega(g(n))$。

$\omega(g(n))$可以视为所有比$g(n)$严格高阶的函数集合：

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

23

HIT CS&E

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，$f(n)\in w(g(n))$。

HIT CS&E

命题 2.1.4 $f(n)=w(g(n))\Rightarrow \lim_{n\to\infty}\frac{f(n)}{g(n)}=\infty$

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，$\lim_{n\to\infty}\frac{f(n)}{g(n)}=\infty$.

HIT CS&E

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） $f(n)=O(f(n))$，

（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 $g(n)=w(f(n))$

HIT CS&E

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？

HIT CS&E

提要

2.1 计算复杂性函数的阶

2.2 和式的计算与估计

2.3 递归方程

HIT CS&E

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
$$\begin{aligned}\sum_{k=1}^{m+1}\theta(f(k))&=\sum_{k=1}^{m}\theta(f(k))+\theta(f(m+1))\\&=\theta\left(\sum_{k=1}^{m}f(k)\right)+\theta(f(m+1))\\&=\theta\left(\sum_{k=1}^{m}f(k)+f(m+1)\right)\\&=\theta\left(\sum_{k=1}^{m+1}f(k)\right)。\end{aligned}$$

HIT
CS&E

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

HIT
CS&E

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 $$\lg\left(\prod_{k=1}^{n} a_k\right) = \sum_{k=1}^{n} \lg a_k$$

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

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

$$\sum_{k=1}^{\infty} \frac{k}{3^k} \le \sum_{k=1}^{\infty} a_1 r^k = \sum_{k=1}^{\infty} \frac{1}{3}\left(\frac{2}{3}\right)^k = \frac{1}{3} \cdot \frac{1}{1-\frac{2}{3}} = 1 .$$

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 $k\ge 3$ 时，$\frac{(k+1)^2/2^{k+1}}{k^2/2^k}=\frac{(k+1)^2}{2k^2}\le\frac{8}{9}$

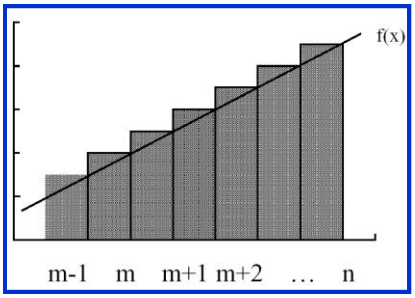
于是 $\sum_{k=0}^{\infty}\frac{k^2}{2^k}=\sum_{k=0}^{2}\frac{k^2}{2^k}+\sum_{k=3}^{\infty}\frac{k^2}{2^k}\le\theta(1)+\sum_{k=3}^{\infty}\frac{9}{8}\cdot\left(\frac{8}{9}\right)^k=O(1)$.

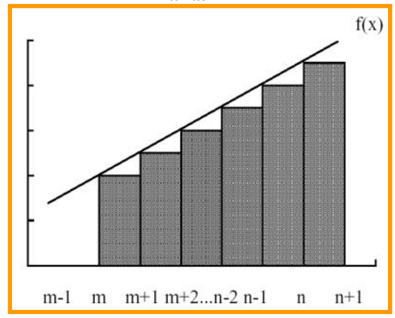
HIT CS&E

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

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

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





$$\sum_{k=m}^{n} f(k) \ge \sum_{k=m-1}^{n-1} f(k)\Delta x \ge \int_{m-1}^{n} f(x)dx\,,\quad f(m-1) < f(n)\,,\quad \Delta x = 1$$

$$\sum_{k=m}^{n} f(k) \le \sum_{k=m}^{n+1} f(k)\Delta x \le \int_{m}^{n+1} f(x)dx$$

HIT
CS&E

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

HIT CS&E

提要

2.1 计算复杂性函数的阶

2.2 和式的计算与估计

2.3 递归方程

HIT CS&E

2.2 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

  $T(n)=2T(n/2)+\theta(n)$ if $n>1$.

  $T(n)=\theta(1)$ if $n=1$(边界条件)

  $T(n)$的解是$\theta(n\log n)$

边界条件是根据问题的不同而不同的！

HIT CS&E

求解递归方程的三个主要方法

- 替换方法：
  - 先猜测方程的解，
  - 然后用数学归纳法证明.
- 迭代方法：
  - 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

41

HIT CS&E

2.3.1 替换(Substitution)方法

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

HIT CS&E

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 $O(n\log n)$。

43

HIT CS&E

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.**

HIT CS&E

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 $T(n)\le cn$ 。

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

错在哪里？ 过早使用了$O(n)$而陷入了陷阱！

应该在证明了$T(n)\le cn$后才可使用。
从$T(n)\le cn+n$不可能得到$T(n)\le cn$
因为对于$\forall c>0$，我们都得不到$cn+n\le cn$。

HIT
CS&E

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

47

HIT
CS&E

2.3.2 迭代(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。

48

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

$$=n+3\left(\left\lfloor n/4\right\rfloor+3T\left(\left\lfloor n/16\right\rfloor\right)\right)$$

$$=n+3\left(\left\lfloor n/4\right\rfloor+3\left(\left\lfloor \frac{n}{16}\right\rfloor+3T\left(\left\lfloor n/64\right\rfloor\right)\right)\right)$$

$$=n+3\left\lfloor n/4\right\rfloor+9\left\lfloor \frac{n}{16}\right\rfloor+27T\left(\left\lfloor n/64\right\rfloor\right)$$

$$=n+3\left\lfloor n/4\right\rfloor+3^2\left|\frac{n}{4^2}\right|+3^3\left(\left|\frac{n}{4^3}\right|\right)+\ldots\ldots+3^i\,T\left(\left\lfloor n/4^i\right\rfloor\right)$$

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

$$=n+3\left\lfloor n/4\right\rfloor+3^2\left\lfloor \frac{n}{4^2}\right\rfloor+3^3\left(\left\lfloor \frac{n}{4^3}\right\rfloor\right)+\ldots\ldots+3^{\log_4 n}T\left(\lfloor 1\rfloor\right)$$

$$\le \sum_{i=0}^{\log_4 n}3^i\,n/4^i\;.\;\le n\sum_{i=0}^{\infty}\left(\frac{3}{4}\right)^i=n\times\frac{1}{1-\frac{3}{4}}=4n=O(n)$$

HIT CS&E

2.3.3 Master method

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。

50

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$T(n)=aT\left(\frac{n}{b}\right)+f(n)$．$T(n)$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$T(n)=\theta\left(n^{\log_b a}\lg n\right)$．

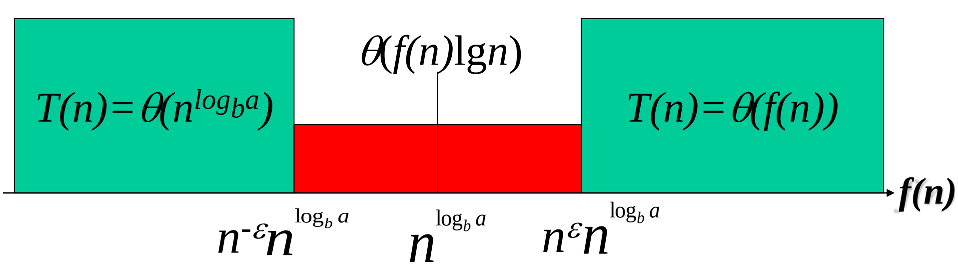
(3)．若$f(n)=\Omega\left(n^{\log_b a+\varepsilon}\right)$，$\varepsilon>0$是常数，且对于所有充分大的n $af\left(\frac{n}{b}\right)\leq cf(n)$，$c$<1是常数，则$T(n)=\theta(f(n))$．

*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

HIT CS&E

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$\varepsilon>0$，$f(n)=\Omega(n^{\log_b a}\cdot n^{\varepsilon})$.

53

HIT CS&E

Master定理的使用

例 1. 求解 $T(n)=9T\left(n/3\right)+n$.

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 $T(n)=T\left(2n/3\right)+1$.

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

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

54

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

解：$a=2^n$，非常数项，不满足master定理条件，
故master定理不适用。

55

HIT
CS&E

例 5. 求解 $T(n)=0.5T(\frac{n}{2})+\frac{1}{n}$

解：$a<1$，不满足master定理条件，
故master定理不适用。

例 6. 求解 $T(n)=64T(n/8)-n^2\log n$

解：$f(n)$非正函数，不满足master定理条件，
故master定理不适用。

HIT
CS&E

扩展master定理

定理 2.4.2 设$a\ge 1$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 $f(n)=\theta(n^{\log_b a}\log^k n), k\ge 0$， 则 $T(n)=\theta(n^{\log_b a}\log^{k+1} n)$

(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 $af\left(n/b\right)\le cf(n)$，$c<1$是常数，则$T(n)=\theta(f(n))$.

57

(2). 若 $f(n)=\theta(n^{\log_b a}\log^k n), k\ge 0$， 则 $T(n)=\theta(n^{\log_b a}\log^{k+1} n)$

例 7. 求解 $T(n)=2T\left(n/2\right)+n\lg n$.

解： $f(n)=n\lg n$

$a=2, b=2$, 根据原始的 master定理， $n^{\log_b a}=n$

$\log n$ 和 n^{ε} 的大小关系：对于任意$\varepsilon>0$，$\log n\in o(n^{\varepsilon})$

显然无法找到大于零的ε使得：

$n\log n=O(n^{1-\varepsilon})$ 或者 $n\log n=\Omega(n^{1+\varepsilon})$ 成立

但根据扩展的master定理：

$a=2, b=2, k=1$, 故： $n^{\log_b a}\log^k n=n\lg n=f(n)$

$T(n)=n\lg^2 n$

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