算法设计与分析

第3讲 基础知识

汪小林

北京大学 信息科学技术学院

数学基础

- 函数的渐近的界
- 常见函数的阶
- 求和技巧与和的渐进界估计
- 递推方程求解
- 主定理证明及应用

函数的渐近的界

- 下列公式表示的确切含义是什么？

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

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

$$f(n) = o(g(n))$$

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

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

函数的渐近的界

定义1.1 设 f 和 g 是定义域为自然数集 $\mathbf{N}$上的函数

(1) 若存在正数 c 和 n_0，使得对一切 $n \geq n_0$ 有
$0 \leq f(n) \leq cg(n)$ 成立，则称 $f(n)$ 的渐近的上界是 $g(n)$，
记作 $f(n) = O(g(n))$.

(2) 若存在正数 c 和 n_0，使得对一切 $n \geq n_0$ 有
$0 \leq cg(n) \leq f(n)$ 成立，则称 $f(n)$ 的渐近的下界是 $g(n)$，
记作 $f(n) = \Omega(g(n))$.

函数的渐近的界

(3) 若对任意正数 c 都存在 n_0，使得当 $n \geq n_0$ 时有

$0 \leq f(n) < cg(n)$ 成立, 则记作 $f(n) = o(g(n))$.

(4) 若对任意正数 c 都存在 n_0，使得当 $n \geq n_0$ 时有

$0 \leq cg(n) < f(n)$ 成立, 则记作 $f(n) = \omega(g(n))$.

(5) 若 $f(n) = O(g(n))$ 且 $f(n) = \Omega(g(n))$, 则记作 $f(n) = \Theta(g(n))$.

例 函数 $f(n) = n^2 + n$,

$f(n) = O(n^2)$, $f(n) = O(n^3)$, $f(n) = o(n^3)$, $f(n) = \Theta(n^2)$

有关定理

定理1.1 设 f 和 g 是定义域为自然数集合的函数.

(1)如果 $\lim_{n\to\infty}\frac{f(n)}{g(n)}$ 存在且等于某个常数 $c>0$，那么 $f(n)=\Theta(g(n))$.

(2) 如果 $\lim_{n\to\infty}\frac{f(n)}{g(n)}=0$，那么 $f(n)=o(g(n))$.

(3) 如果 $\lim_{n\to\infty}\frac{f(n)}{g(n)}=+\infty$，那么 $f(n)=\omega(g(n))$.

证明定理1.1 (1)

证：根据极限定义，对于给定的正数$\varepsilon = c/2$, 存在某个n_0，只要 $n \geq n_0$，就有

$$\left|\frac{f(n)}{g(n)} - c\right| < \varepsilon \Rightarrow c - \varepsilon < \frac{f(n)}{g(n)} < c + \varepsilon$$

$$\Rightarrow \frac{c}{2} < \frac{f(n)}{g(n)} < \frac{3c}{2} < 2c$$

对所有的 $n \geq n_0$，$f(n) \leq 2cg(n)$，从而推出 $f(n) = O(g(n))$，

对所有的 $n \geq n_0$，$f(n) \geq (c/2)g(n)$，从而推出 $f(n) = \Omega(g(n))$，

于是 $f(n) = \Theta(g(n))$。

有关阶的一些性质

定理1.2 设 f, g, h 是定义域为自然数集合的函数，

(1) 如果 $f = O(g)$ 且 $g = O(h)$，那么 $f = O(h)$.

(2) 如果 $f = \Omega(g)$ 且 $g = \Omega(h)$，那么 $f = \Omega(h)$.

(3) 如果 $f = \Theta(g)$ 和 $g = \Theta(h)$，那么 $f = \Theta(h)$.

定理1.3 假设 f 和 g 是定义域为自然数集合的函数，若对某个其它的函数h, 有 $f = O(h)$ 和 $g = O(h)$，那么 $f + g = O(h)$.

实例

例题 设 $f(n)=\frac{1}{2}n^2-3n$ ，证明 $f(n)=\Theta(n^2)$.

证：因为

$$\lim_{n\to+\infty}\frac{f(n)}{n^2}=\lim_{n\to+\infty}\frac{\frac{1}{2}n^2-3n}{n^2}=\frac{1}{2}$$

根据定理1.1有 $f(n)=\Theta(n^2)$.

可以证明：多项式函数，幂函数的阶低于指数函数

$$n^d=o(r^n)，r>1，d>0$$

对数函数

符号：

$$\log n = \log_2 n$$

$$\log^k n = (\log n)^k$$

$$\log\log n = \log(\log n)$$

性质：

$$\log_b n = o(n^\alpha) \quad \alpha > 0$$

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

$$\log_k n = \Theta(\log_l n)$$

阶乘

Stirling公式 $$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\Theta\left(\frac{1}{n}\right)\right)$$

$$n! = o(n^n)$$

$$n! = \Omega(2^n)$$

$$\log(n!) = \Theta(n\log n)$$

$$\lim_{n\to+\infty}\frac{\log(n!)}{n\log n} = \lim_{n\to+\infty}\frac{\ln(n!)/\ln 2}{n\ln n/\ln 2} = \lim_{n\to+\infty}\frac{\ln(n!)}{n\ln n}$$

$$= \lim_{n\to+\infty}\frac{\ln\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\left(1+\left(\frac{c}{n}\right)\right)\right)}{n\ln n} = \lim_{n\to+\infty}\frac{\ln\sqrt{2\pi n}+n\ln\frac{n}{e}}{n\ln n} = 1$$

上述的c为某个常数

例题：函数的阶

按照阶从高到低对以下函数排序：

$$\log^2 n,\quad 1,\quad n!,\quad n2^n,\quad n^{1/\log n},\quad (3/2)^n,\quad \sqrt{\log n},\quad (\log n)^{\log n},$$

$$2^{2^n},\quad n^{\log\log n},\quad n^3,\quad \log\log n,\quad n\log n,\quad n,\quad 2^{\log n},\quad \log n,\quad \log(n!)$$

结果：

$$2^{2^n},\quad n!,\quad n2^n,\quad (3/2)^n,\quad (\log n)^{\log n} = n^{\log\log n},$$

$$n^3,\quad \log(n!) = \Theta(n\log n),\quad n = \Theta(2^{\log n}),$$

$$\log^2 n,\quad \log n,\quad \sqrt{\log n},\quad \log\log n,$$

$$n^{1/\log n} = \Theta(1)$$

取整函数

$\lfloor x \rfloor$：表示小于等于 x 的最大的整数

$\lceil x \rceil$：表示大于等于 x 的最小的整数

取整函数具有下述性质：

(1) $x-1 < \lfloor x \rfloor \le x \le \lceil x \rceil < x+1$

(2) $\lfloor x+n \rfloor = \lfloor x \rfloor + n$, $\lceil x+n \rceil = \lceil x \rceil + n$，其中$n$为整数

(3) $\left\lceil \dfrac{n}{2} \right\rceil + \left\lfloor \dfrac{n}{2} \right\rfloor = n$

(4) $\left\lceil \dfrac{\left\lceil \frac{n}{a} \right\rceil}{b} \right\rceil = \left\lceil \dfrac{n}{ab} \right\rceil$, $\left\lfloor \dfrac{\left\lfloor \frac{n}{a} \right\rfloor}{b} \right\rfloor = \left\lfloor \dfrac{n}{ab} \right\rfloor$

递推方程的求解

设序列 $a_0, a_1, \ldots, a_n, \ldots$，简记为 $\{a_n\}$，一个把 a_n与某些个$a_i(i<n)$联系起来的等式叫做关于序列 $\{a_n\}$ 的递推方程

例子：$a_n = a_{n-1} + a_{n-2}$, $a_0 = 0$, $a_1 = 1$

求解目标：给出 a_n 的关于 n 的显示公式

$$a_n = \frac{1}{\sqrt{5}}\left(\phi^n - \hat{\phi}^n\right)$$

$$\phi = \frac{1+\sqrt{5}}{2}, \hat{\phi} = \frac{1-\sqrt{5}}{2}$$

序列求和

- 几个有用的结果

$$\sum_{k=1}^{n} a_k = \frac{n(a_1 + a_n)}{2}$$ 等差级数 – $\{a_k\}$

$$\sum_{k=0}^{n} aq^k = \frac{a(1-q^{n+1})}{1-q}$$ 等比级数 – $\{aq^k\}$

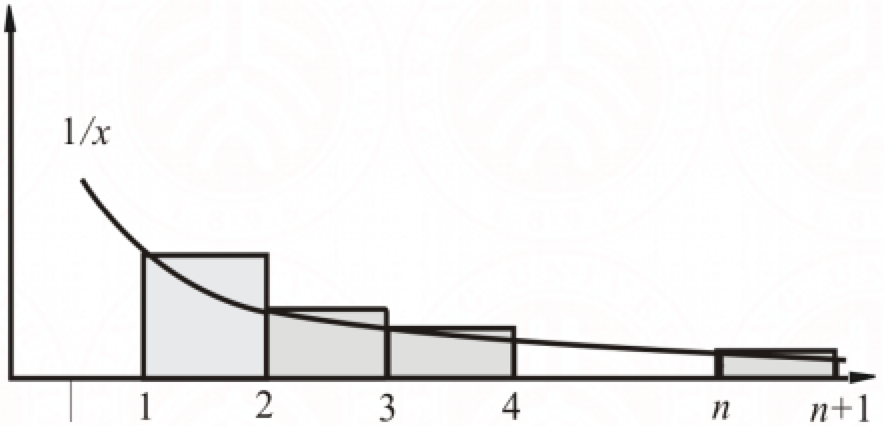
$$\sum_{k=1}^{n} \frac{1}{k} = \ln n + \mathrm{O}(1)$$ 调和级数 – $\{1/k\}$

估计和式的渐近的界

估计 $\sum_{k=1}^{n}\frac{1}{k}$ 的渐近的界.

$$\sum_{k=1}^{n}\frac{1}{k} \ge \int_{1}^{n+1}\frac{dx}{x}$$

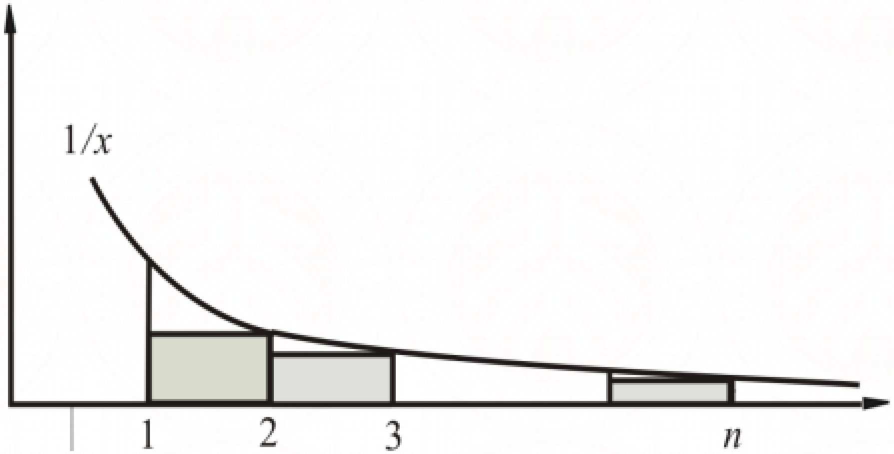
$$= \ln(n+1)$$



$$\sum_{k=1}^{n}\frac{1}{k} = 1+\sum_{k=2}^{n}\frac{1}{k}$$

$$\le 1+\int_{1}^{n}\frac{dx}{x}$$

$$= \ln n + 1$$



求和实例

$$\sum_{k=1}^{n-1}\frac{1}{k(k+1)}$$
$$=\sum_{k=1}^{n-1}\left(\frac{1}{k}-\frac{1}{k+1}\right)$$
$$=\sum_{k=1}^{n-1}\frac{1}{k}-\sum_{k=1}^{n-1}\frac{1}{k+1}$$
$$=\sum_{k=1}^{n-1}\frac{1}{k}-\sum_{k=2}^{n}\frac{1}{k}$$
$$=1-\frac{1}{n}$$

$$\sum_{t=1}^{k}t2^{t-1}=\sum_{t=1}^{k}t\left(2^t-2^{t-1}\right)=\sum_{t=1}^{k}t2^t-\sum_{t=1}^{k}t2^{t-1}$$
$$=\sum_{t=1}^{k}t2^t-\sum_{t=0}^{k-1}(t+1)2^t$$
$$=\sum_{t=1}^{k}t2^t-\sum_{t=0}^{k-1}t2^t-\sum_{t=0}^{k-1}2^t$$
$$=k2^k-\left(2^k-1\right)$$
$$=(k-1)2^k+1$$

序列求和

- 和的上界

$$\sum_{k=1}^{n} a_k \le n a_{\max}$$

- 假设存在常数 $r<1$，使得对一切 $k\ge 0$，$\dfrac{a_{k+1}}{a_k}\le r$成立，则

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

求和实例

例7 估计 $\sum_{k=1}^{n}\frac{k}{3^k}$ 的上界.

解：由

$$a_k=\frac{k}{3^k},\quad a_{k+1}=\frac{k+1}{3^{k+1}}$$

得

$$\frac{a_{k+1}}{a_k}=\frac{1}{3}\frac{k+1}{k}\le\frac{2}{3}$$

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

递推方程的求解

设序列 $a_0, a_1, \ldots, a_n, \ldots$，简记为 $\{a_n\}$，一个把 a_n 与某些个 $a_i (i<n)$ 联系起来的等式叫做关于序列 $\{a_n\}$ 的递推方程

求解方法：

- 迭代法
  直接迭代：插入排序最坏情况下时间分析
  换元迭代：二分归并排序最坏情况下时间分析
  差消迭代：快速排序平均情况下的时间分析
  迭代模型：递归树
- 尝试法：快速排序平均情况下的时间分析
- 主定理：递归算法的分析

直接迭代：插入排序

算法1.4 InsertSort(A,n) // A为n个数的数组

1. **for** $j \leftarrow 2$ **to** n **do**
2. $\quad x \leftarrow A[j]$
3. $\quad i \leftarrow j-1$ // 行3到行7把$A[j]$插入$A[1..j-1]$之中
4. $\quad$ **while** $i>0$ **and** $x<A[i]$ **do**
5. $\qquad A[i+1] \leftarrow A[i]$
6. $\qquad i \leftarrow i-1$
7. $\quad A[i+1] \leftarrow x$

$$\begin{cases} W(n) = W(n-1) + n - 1 \\ W(1) = 0 \end{cases}$$

$$W(n) = W(n-1)+n-1 = [W(n-2)+n-2]+n-1 =$$
$$W(n-2)+(n-2)+(n-1) \qquad = [W(n-3)+n-3]+(n-2)+(n-1) = \ldots$$
$$= W(1)+1+2+\ldots+(n-2)+(n-1)$$
$$= 1+2+\ldots+(n-2)+(n-1)=n(n-1)/2$$

二分归并排序

算法1.5 MergeSort(A, p, r) // 归并排序数组 $A[p..r]$

1. **if** $p<r$
2. **then** $q \leftarrow \lfloor (p+r)/2 \rfloor$
3. MergeSort(A,p,q)
4. MergeSort($A,q+1,r$)
5. Merge(A,p,q,r)

$$\begin{cases} W(n) = 2W(n/2) + n - 1 \\ W(1) = 0 \end{cases}$$

归并过程

算法1.6 Merge(*A*,*p*,*q*,*r*) // 将排序数组*A*[*p*..*q*]与*A*[*q*+1,*r*]合并

```
x←q−p+1, y←r−q                  // x, y分别为两个子数组的元素数
将A[p..q]复制到B[1..x], 将A[q+1..r]复制到C[1..y]
i←1, j←1, k←p
while i≤x and j≤y do
    if B[i]≤C[j]                 // B的首元素不大于C的首元素
    then  A[k]←B[i]              // 将B的首元素放到A中
              i←i+1
    else
              A[k]←C[j]
              j←j+1
    k←k+1
if i>x then 将C[j..y]复制到A[k..r]    // B已经是空数组
else 将B[i..x]复制到A[k..r]           // C已经是空数组
```

换元迭代

$$\begin{cases} W(n) = 2W(n/2) + n - 1, \quad n = 2^k \\ W(1) = 0 \end{cases}$$

$$W(n) = 2W(2^{k-1}) + 2^k - 1$$

$$= 2[2W(2^{k-2}) + 2^{k-1} - 1] + 2^k - 1$$

$$= 2^2 W(2^{k-2}) + 2^k - 2 + 2^k - 1$$

$$= 2^2[2W(2^{k-3}) + 2^{k-2} - 1] + 2^k - 2 + 2^k - 1 = \ldots$$

$$= 2^k W(1) + k2^k - (2^{k-1} + 2^{k-2} + \ldots + 2 + 1)$$

$$= k2^k - 2^k + 1$$

$$= n\log n - n + 1$$

* 使用迭代法，对解可以通过数学归纳法验证

差消化简后迭代

$$\begin{cases} T(n) = \dfrac{2}{n}\sum_{i=1}^{n-1} T(i) + cn, & n \geq 2 \\ T(1) = 0 \end{cases}$$

快速排序平均时间分析

$$nT(n) = 2\sum_{i=1}^{n-1} T(i) + cn^2$$

$$(n-1)T(n-1) = 2\sum_{i=1}^{n-2} T(i) + c(n-1)^2$$

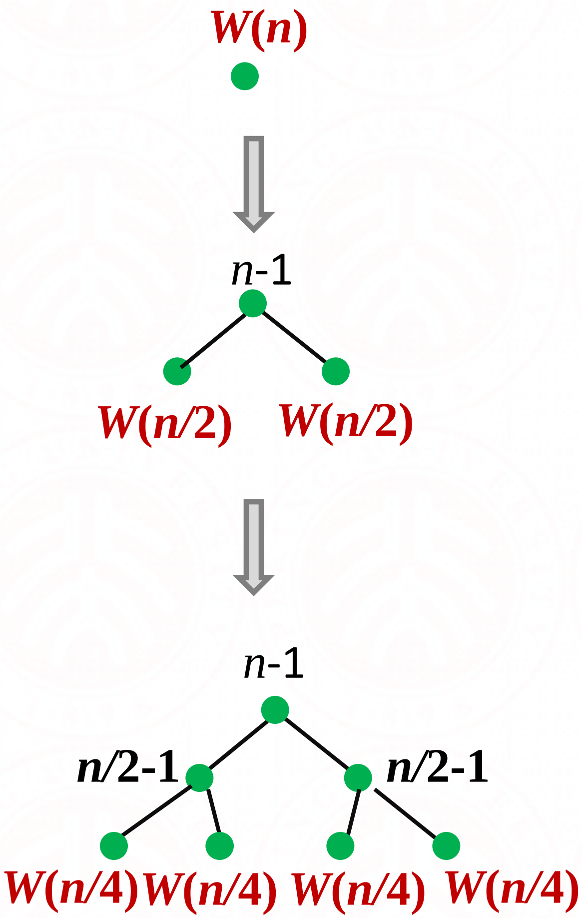
$$nT(n) = (n+1)T(n-1) + 2cn - c$$

$$\frac{T(n)}{n+1} = \frac{T(n-1)}{n} + \frac{2cn-c}{n(n+1)} = \cdots$$

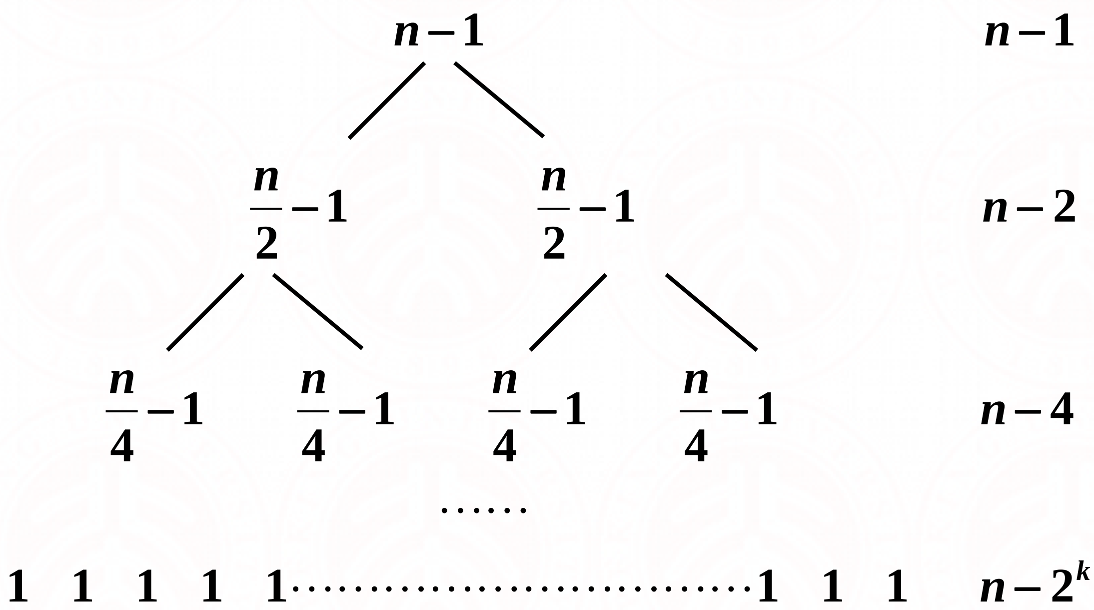
$$= 2c[\frac{1}{n+1} + \frac{1}{n} + \ldots + \frac{1}{3} + \frac{T(1)}{2}] - O(\frac{1}{n}) = \Theta(\log n)$$

$$T(n) = \Theta(n \log n)$$

迭代模型：递归树



生成过程

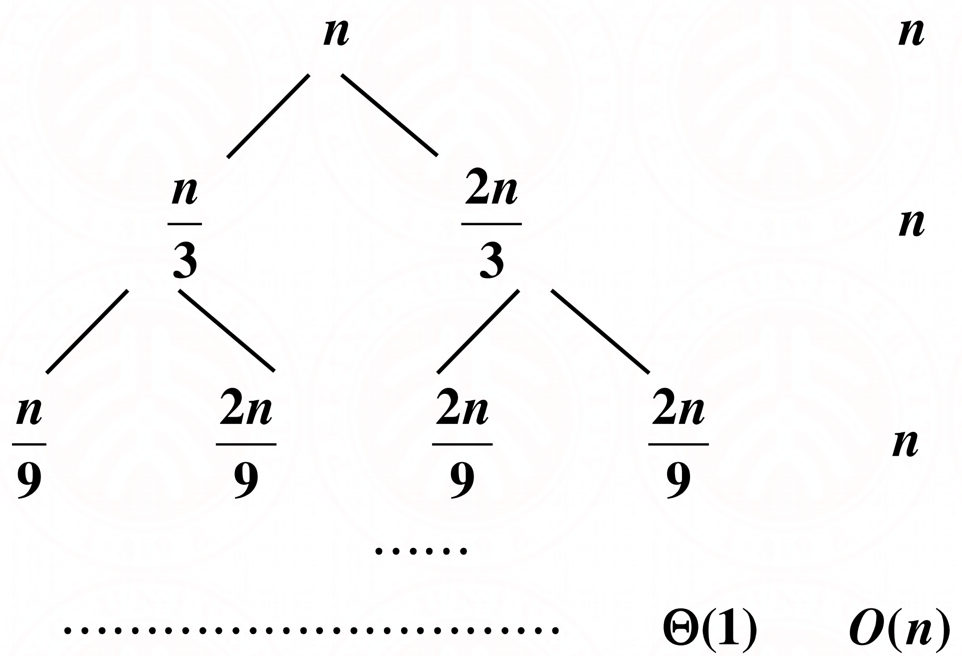


$$W(n)=2W(n/2)+n-1,\ n=2^k,\quad W(1)=0$$

$$W(n)=n-1+n-2+\ldots+n-2^k$$
$$=kn-(2^k-1)=n\log n-n+1$$

递归树的应用实例

求解：$T(n)=T(n/3)+T(2n/3)+n$



层数 k: $n(2/3)^k = \Theta(1) \Rightarrow (3/2)^k = \Theta(n) \Rightarrow k=O(\log_{3/2} n)$

$T(n)=O(n\log n)$

尝试法：快速排序

$$T(n)=\frac{2}{n}\sum_{i=1}^{n-1}T(i)+O(n)$$

(1) $T(n)=C$为常函数，左边$=O(1)$

右边$=\frac{2}{n}C(n-1)+O(n)=2C-\frac{2C}{n}+O(n)$

(2) $T(n)=cn$，左边$=cn$

右边$=\frac{2}{n}\sum_{i=1}^{n-1}ci+O(n)=\frac{2c}{n}\frac{(1+n-1)(n-1)}{2}+O(n)=cn-c+O(n)$

(3) $T(n)=cn^2$，左边$=cn^2$

右边$=\frac{2}{n}\sum_{i=1}^{n-1}ci^2+O(n)=\frac{2}{n}[\frac{cn^3}{3}+O(n^2)]+O(n)=\frac{2c}{3}n^2+O(n)$

(4) $T(n)=cn\log n$，左边$=cn\log n$

右边 $=\frac{2c}{n}\sum_{i=1}^{n-1}i\log i+O(n)=cn\log n+O(n)$

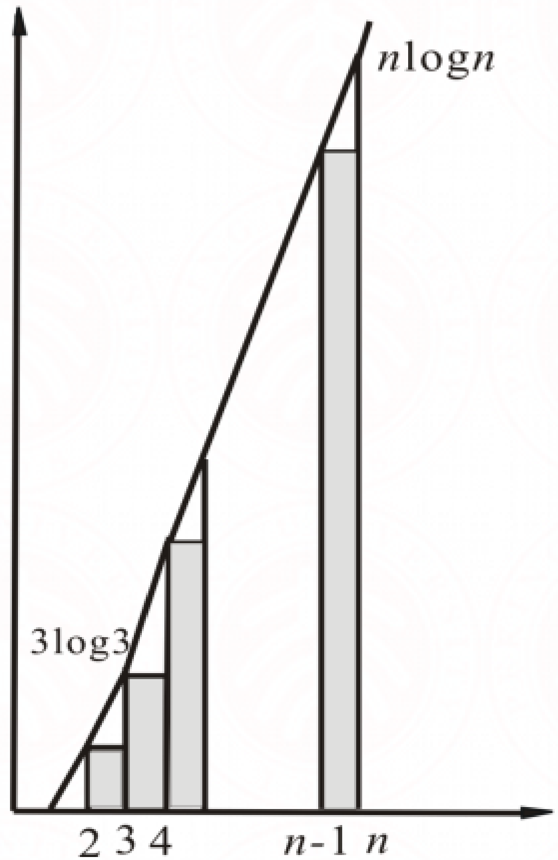
以积分作为求和的近似

$$\int_1^{n-1} x\log x dx \le \sum_{i=1}^{n-1} i\log i \le \int_2^n x\log x dx$$

$$\int_2^n x\log x dx = \int_2^n \frac{x}{\ln 2}\ln x dx$$

$$= \frac{1}{\ln 2}\left[\frac{x^2}{2}\ln x - \frac{x^2}{4}\right]\Bigg|_2^n$$

$$= \frac{1}{\ln 2}\left(\frac{n^2}{2}\ln n - \frac{n^2}{4}\right) - \frac{1}{\ln 2}\left(\frac{4}{2}\ln 2 - \frac{4}{4}\right)$$



递推方程的归纳证明

- 尝试（猜测）递推方程的解应用归纳法严格证明

- 归纳法求解递推方程的三个步骤
  - 猜测解的形式
  - 用数学归纳法证明
    - 找出使解有效的常数
  - 确定常数使边界条件成立
- 常用技巧
  - 通过引入低阶项获得更紧的解的形式

递推方程的归纳证明

例：$T(n) = 4T(n/2) + n$

- [假定$T(1)=\Theta(1)$]
- 猜测$T(n)= O(n^3)$（分别证明O和Ω 关系）
- 假设，对于所有的$k<n$

$$T(k) \le ck^3$$

- 通过归纳法证明

$$T(n) \le cn^3$$

例：$T(n) = 4T(n/2) + n$

$$
\begin{aligned}
T(n) &= 4T(n/2) + n \\
&\le 4c(n/2)^3 + n \\
&= (c/2)n^3 + n \\
&= cn^3 - ((c/2)n^3 - n) \\
&\le cn^3
\end{aligned}
$$

期望的形式 – 余项

期望的形式

这里要保证：$((c/2)n^3 - n) \ge 0$,

余项

这只需要：$c \ge 2$并且$n \ge 1$

于是有 $T(n) = O(n^3)$

例：$T(n) = 4T(n/2) + n$

- 还必须处理初始情形，才能使归纳成立。
- 注意到，因为对所有的 $1 \le n < n_0$ 都有 $T(n) = \Theta(1)$（其中n_0是某个适当的常数）
- 于是当 $1 \le n < n_0$时，只要 c 足够大，就有

$$\text{“}\Theta(1)\text{”} \le cn^3$$

- 但这个界并不够紧

更紧的上界

- 我们来证明 $T(n) = \mathrm{O}(n^2)$
- 假设对于所有的 $k<n$，有 $T(k) \le ck^2$

$$T(n) = 4T(n/2) + n$$
$$\le 4c(n/2)^2 + n = cn^2 + n$$
$$= \cancel{\mathrm{O}(n^2)}$$
$$= cn^2 - (-n)$$
$$\le cn^2$$

错！必须证明完全一致的形式

$\longleftarrow$ 期望的形式 – 余项

但对任何 $c > 0$，上式最后一步不可能成立！

更紧的上界

要点：加强归纳假设

*减去一个低阶项

假设：对于 $k < n$，有 $T(k) \le c_1k^2 - c_2k$

$$T(n) = 4T(n/2) + n$$

$$\le 4(c_1(n/2)^2 - c_2(n/2)) + n$$

$$= c_1n^2 - 2c_2n + n$$

$$= c_1n^2 - c_2n - (c_2n - n)$$

$$\le c_1n^2 - c_2n \text{ 当 } c_2 > 1$$

可以取 c_1 足够大来处理初始情况。

例：$T(n) = 4T(n/2) + n = \Omega(n^2)$

- 再证明： $T(n) = \Omega(n^2)$
- 假设对于 $k < n$，有 $T(k) \geq ck^2$

$$
\begin{aligned}
T(n) &= 4T(n/2) + n \\
&\geq 4c(n/2)^2 + n \\
&= cn^2 + n
\end{aligned}
$$

- 取 c 足够小来处理初始情况。
- $T(n) = O(n^2)$且$T(n) = \Omega(n^2)$得$T(n) = \Theta(n^2)$

换元法的求解递推方程

- 例：$T(n)=2T(\sqrt{n})+\log n$
- 通过改变变量转化递归式，将 $\sqrt{n}$ 转化为整数。
  令 $m = \log n$，于是

  $$T(2^m) = 2T(2^{m/2}) + m$$

- 再令 $S(m) = T(2^m)$，于是

  $$S(m) = 2S(m/2) + m$$

  $$= \Theta(m\log m)$$

  $$T(n) = T(2^m) = S(m) = \Theta(m\log m)$$

  $$= \Theta(\log n \log\log n)$$

主定理

定理： 设$a\geq 1$, $b>1$为常数，$f(n)$为函数，$T(n)$为非负整数，且

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

则有以下结果：

1. 若$f(n)=O(n^{\log_b a-\varepsilon}), \varepsilon>0$, 那么$T(n)=\Theta(n^{\log_b a})$
2. 若$f(n)=\Theta(n^{\log_b a})$, 那么$T(n)=\Theta(n^{\log_b a}\log n)$
3. 若$f(n)=\Omega(n^{\log_b a+\varepsilon}), \varepsilon>0$,
   且对某个常数 $c<1$ 和 所有充分大的 n 有
   $a\, f(n/b)\leq c\, f(n)$，那么 $T(n)=\Theta(f(n))$

主定理的应用

例9 求解递推方程 $T(n)=9T(n/3)+n$

解 上述递推方程中的 $a=9, b=3, f(n)=n$，那么

$$n^{\log_3 9}=n^2, \quad f(n)=O(n^{\log_3 9-1}),$$

相当于主定理的第一种情况，其中ε=1. 根据定理得到

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

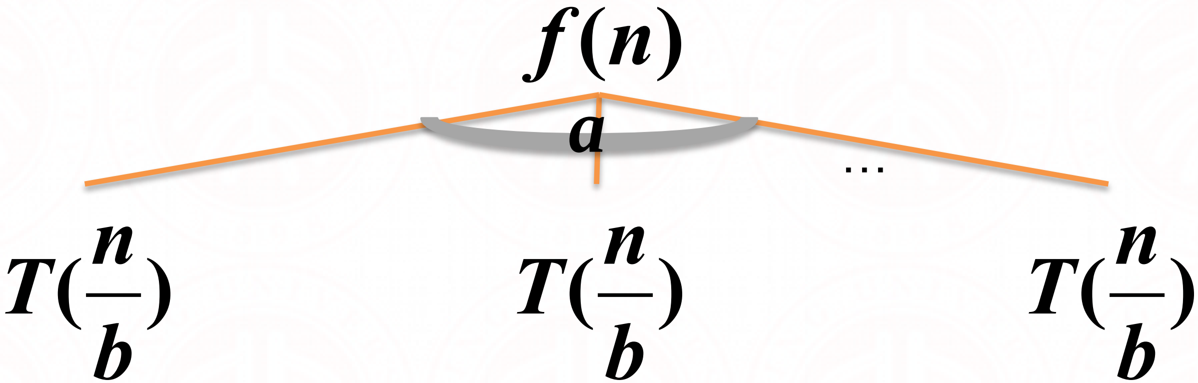
例10 求解递推方程 $T(n)=T(2n/3)+1$

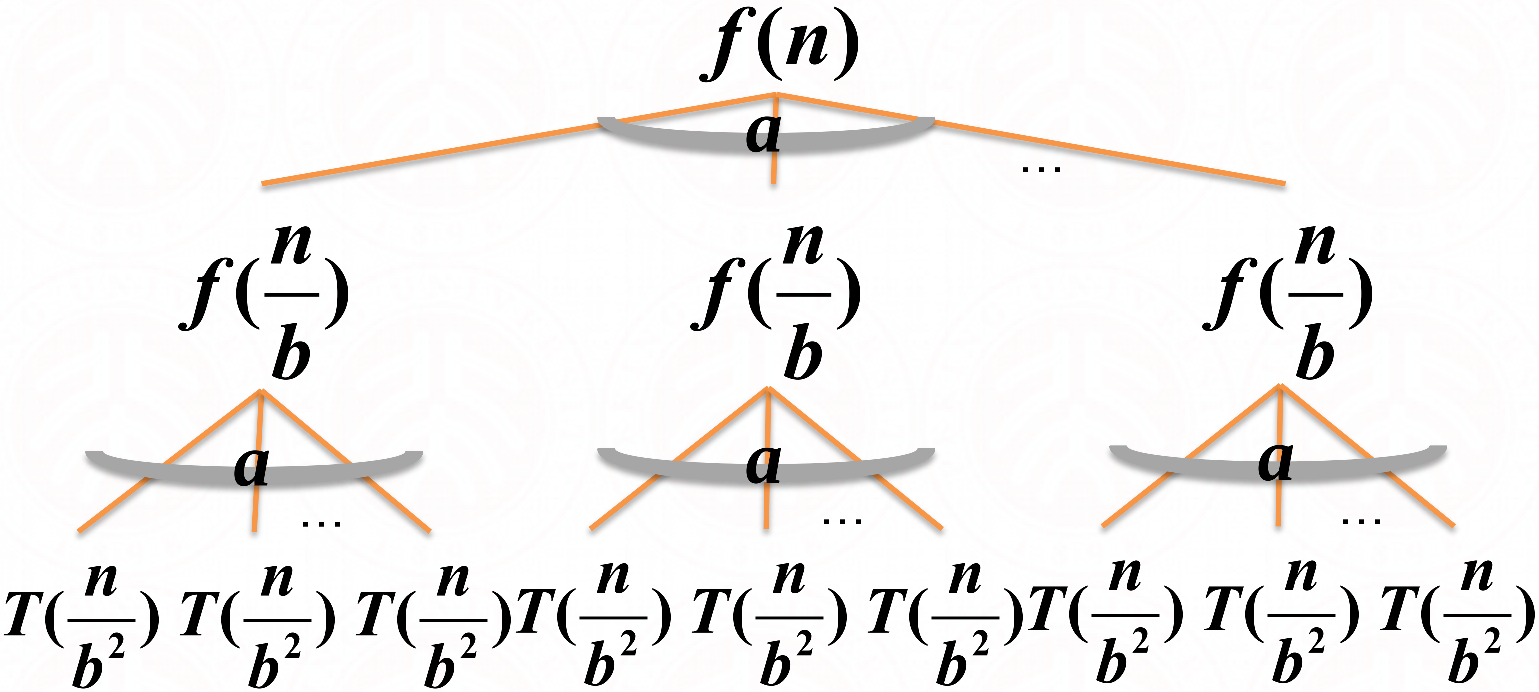
解 上述递推方程中的 a=1, b=3/2, $f(n)$=1，那么

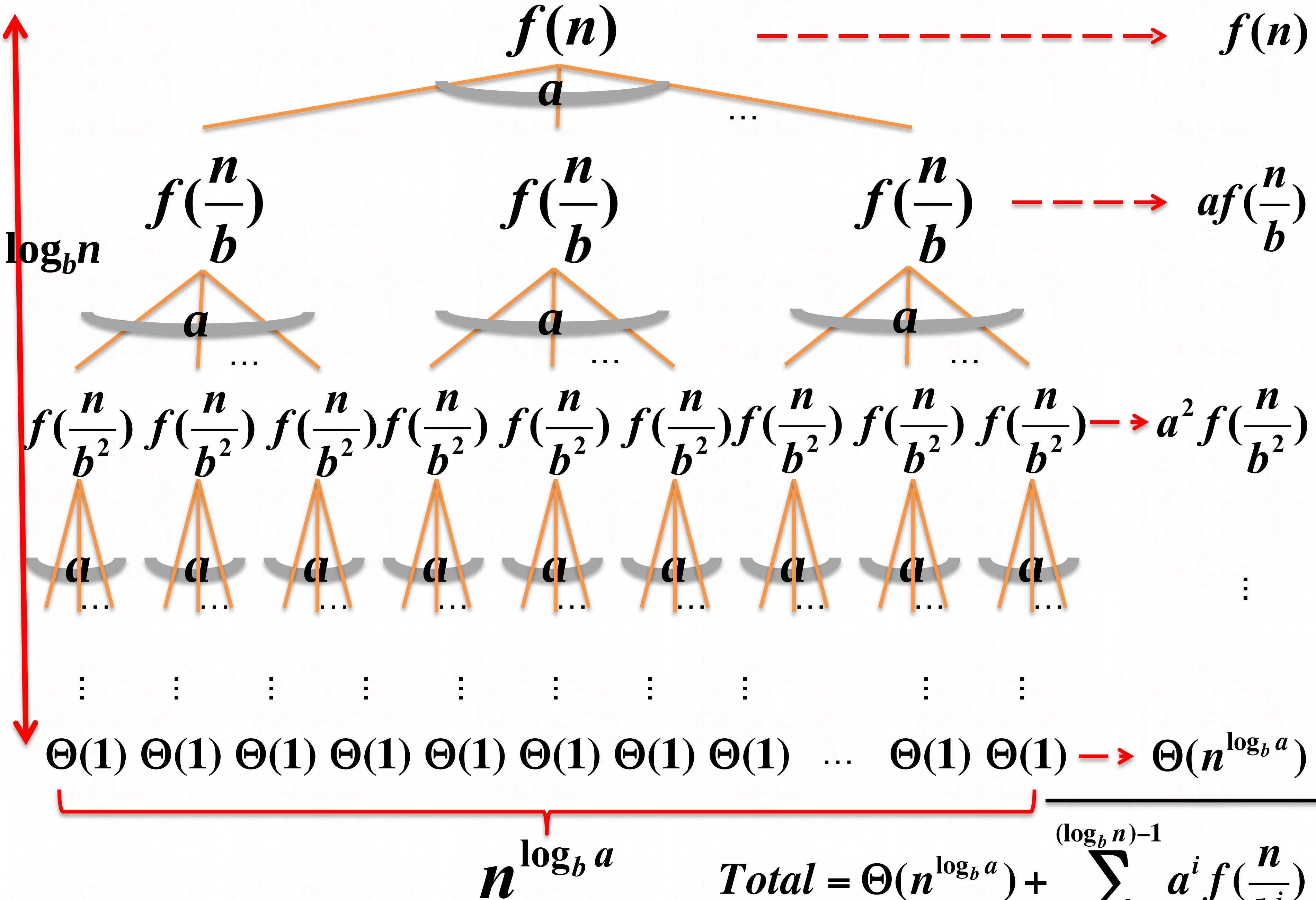
$$n^{\log_{3/2} 1}=n^0=1, \quad f(n)=1$$

相当于主定理的第二种情况. 根据定理得到 .

$$T(n)=\Theta(n^0 \log n)=\Theta(\log n)$$


$f(n)$
a
$\cdots$
$T(\frac{n}{b})$
$T(\frac{n}{b})$
$T(\frac{n}{b})$


$f(n)$
a
…
$f(\frac{n}{b})$ $f(\frac{n}{b})$ $f(\frac{n}{b})$
a a a
… … …
$T(\frac{n}{b^2})$ $T(\frac{n}{b^2})$ $T(\frac{n}{b^2})$ $T(\frac{n}{b^2})$ $T(\frac{n}{b^2})$ $T(\frac{n}{b^2})$ $T(\frac{n}{b^2})$ $T(\frac{n}{b^2})$ $T(\frac{n}{b^2})$



$$Total = \Theta(n^{\log_b a}) + \sum_{i=0}^{(\log_b n)-1} a^i f\left(\frac{n}{b^i}\right)$$

主定理的证明

不妨设$n=b^k$

$$T(n)=aT(\frac{n}{b})+f(n)$$

$$=a[aT(\frac{n}{b^2})+f(\frac{n}{b})]+f(n)=a^2T(\frac{n}{b^2})+af(\frac{n}{b})+f(n)$$

$$=\ldots$$

$$=a^kT(\frac{n}{b^k})+a^{k-1}f(\frac{n}{b^{k-1}})+\ldots+af(\frac{n}{b})+f(n)$$

$$=a^kT(1)+\sum_{j=0}^{k-1}a^j f(\frac{n}{b^j})$$

$$=c_1 n^{\log_b a}+\sum_{j=0}^{k-1}a^j f(\frac{n}{b^j}) \qquad T(1)=c_1$$

情况1

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

$$T(n)=c_1 n^{\log_b a}+\sum_{j=0}^{k-1} a^j f(\frac{n}{b^j})$$

$$=c_1 n^{\log_b a}+O(\sum_{j=0}^{\log_b n-1} a^j (\frac{n}{b^j})^{\log_b a-\varepsilon})$$

$$=c_1 n^{\log_b a}+O(n^{\log_b a-\varepsilon}\sum_{j=0}^{\log_b n-1}\frac{a^j}{(b^{\log_b a-\varepsilon})^j})$$

$$=c_1 n^{\log_b a}+O(n^{\log_b a-\varepsilon}\sum_{j=0}^{\log_b n-1}(b^{\varepsilon})^j)$$

$$=c_1 n^{\log_b a}+O(n^{\log_b a-\varepsilon}\frac{b^{\varepsilon\log_b n}-1}{b^{\varepsilon}-1})$$

$$=c_1 n^{\log_b a}+O(n^{\log_b a-\varepsilon}n^{\varepsilon})=\Theta(n^{\log_b a})$$

情况2

$$f(n)=\Theta(n^{\log_b a})$$

$$T(n)=c_1 n^{\log_b a}+\sum_{j=0}^{k-1} a^j f(\frac{n}{b^j})$$

$$=c_1 n^{\log_b a}+\Theta(\sum_{j=0}^{\log_b n-1} a^j (\frac{n}{b^j})^{\log_b a})$$

$$=c_1 n^{\log_b a}+\Theta(n^{\log_b a}\sum_{j=0}^{\log_b n-1} \frac{a^j}{a^j})$$

$$=c_1 n^{\log_b a}+\Theta(n^{\log_b a}\log n)$$

$$=\Theta(n^{\log_b a}\log n)$$

情况3

$$f(n) = \Omega(n^{\log_b a+\varepsilon})$$

$$T(n) = c_1 n^{\log_b a} + \sum_{j=0}^{k-1} a^j f(\frac{n}{b^j})$$

$$\le c_1 n^{\log_b a} + \sum_{j=0}^{\log_b n-1} c^j f(n) \qquad (af(\frac{n}{b}) \le cf(n))$$

$$= c_1 n^{\log_b a} + f(n)\frac{c^{\log_b n} - 1}{c-1}$$

$$= c_1 n^{\log_b a} + \Theta(f(n)) \qquad （c<1）$$

$$= \Theta(f(n)) \qquad (f(n) = \Omega(n^{\log_b a+\varepsilon}))$$

应用实例

例11 求解递推方程

$$T(n) = 3T(n/4) + n\log n$$

解 上述递推方程中的a=3, b=4, $f(n)$=nlog$n$，那么

$$n\log n = \Omega(n^{\log_4 3+\varepsilon}) = \Omega(n^{0.793+\varepsilon})$$ ，ε≈0.2

此外，要使 $a\,f(n/b) \le cf(n)$成立，代入 $f(n)$=nlog$n$，得到

$$\frac{3n}{4}\log\frac{n}{4} \le cn\log n$$

显然只要 $c \ge 3/4$，上述不等式就可以对充分大的n成立. 相当于主定理的第三种情况. 因此有

$$T(n) = \Theta(f(n)) = \Theta(n\log n)$$

不能直接使用主定理的例子

例12 求解 $T(n)=2T(n/2)+n\log n$

$a=b=2$, $n^{\log_2 2}=n$, $f(n)=n\log n$

不存在$\varepsilon>0$ 使得 $n\log n=\Omega(n^{1+\varepsilon})$ 成立，

$$n\log n \qquad\qquad n\log n$$

$$\frac{n(\log n-1)}{2} \quad \frac{n(\log n-1)}{2} \qquad\qquad n(\log n-1)$$

$$\frac{n(\log n-2)}{4} \quad \frac{n(\log n-2)}{4} \quad \frac{n(\log n-2)}{4} \quad \frac{n(\log n-2)}{4} \qquad n(\log n-2)$$

....

$$\begin{aligned} T(n) &= n\log n + n(\log n-1)+n(\log n-2)+\ldots+n(\log n-k+1) \\ &= (n\log n)\log n - n(1+2+\ldots+k-1) \\ &= n\log^2 n - nk(k-1)/2 = O(n\log^2 n) \end{aligned}$$

主定理扩展

定理： 设$a \ge 1$, $b>1$为常数，$f(n)$为函数，$T(n)$为非负整数，且

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

则有以下结果：

1. $\text{if } f(n)=O(n^{\log_b a-\varepsilon}), \varepsilon>0, \text{ then } T(n)=\Theta(n^{\log_b a})$
2. $\text{if } f(n)=\Theta(n^{\log_b a}\log^k n), \text{ then } T(n)=\Theta(n^{\log_b a}\log^{k+1} n)$
3. $\text{if } f(n)=\Omega(n^{\log_b a+\varepsilon}), \varepsilon>0, \text{ then } T(n)=\Theta(f(n))$

通用定理 (Akra-Bazzi Theorem)

对于形如下式的递推公式，其中 $a_i \geq 1, b_i > 1$ **为常数**

$$T(n) = \sum_{i=0}^{k} a_i T\left(\frac{n}{b_i}\right) + f(n)$$

如果存在唯一的正实数 p 为使得下式成立。

$$\sum_{i=0}^{k}\left(\frac{a_i}{(b_i)^p}\right) = 1$$

则有

情形 **1.** $f(n) = O(n^{p-\varepsilon}) \rightarrow T(n) = \Theta(n^p)$

情形 **2.** $f(n) = \Theta(n^p \log^k n) \rightarrow T(n) = \Theta(n^p \log^{k+1} n)$

情形 **3.** $f(n) = \Omega(n^{p+\epsilon}) \rightarrow T(n) = \Theta(f(n))$

通用定理推论

对于形如下式的递推公式，

$$T(n)=\sum_{i=0}^{k}a_iT\left(\frac{n}{b_i}\right)+f(n)$$

则有

推论 1：

$$\left.\begin{aligned}\sum_{i=0}^{k}\frac{a_i}{b_i}=1\\ f(n)=\Theta\left(n\log^{k}n\right)\end{aligned}\right\}\to T(n)=\Theta\left(n\log^{k+1}n\right)$$

推论 2：

$$\left.\begin{aligned}\sum_{i=1}^{k}\frac{a_i}{b_i}<1\\ f(n)=\Omega(n)\end{aligned}\right\}\to T(n)=\Theta\left(f(n)\right)$$

通用定理 例题

$$T(n)=T\left(\frac{n}{3}\right)+T\left(\frac{2n}{3}\right)+cn$$

解：

因为 $a_1=a_2=1$，$b_1=3$，$b_2=\frac{3}{2}$,

所以有 $\frac{a_1}{b_1}+\frac{a_2}{b_2}=\frac{1}{3}+\frac{2}{3}=1$,

又有 $f(n)=cn=\Theta(n)$，根据通用定理推论 1，得

$$T(n)=\Theta(n\log n)$$

通用定理 例题

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{3}\right) + cn$$

解：

因为 $a_1 = a_2 = 1$， $b_1 = 2$， $b_2 = 3$，

所以有 $\frac{a_1}{b_1} + \frac{a_2}{b_2} = \frac{1}{2} + \frac{1}{3} = \frac{5}{6} < 1$，

又有 $f(n) = cn = \Omega(n)$，根据通用定理推论 2，得

$$T(n) = \Theta(n)$$

小结

- 函数的渐近的界
- 常见函数的阶
- 求和技巧与和的渐进界估计
- 递推方程求解
- 主定理证明及应用