一、有关算法时间复杂度的**符号介绍**

1. O

$$f(n)=\mathrm{O}(g(n))$$

等价于 $\exists c>0; n_0>0$，$\forall n\ge n_0$，使得$0\le f(n)\le cg(n)$。

2. Ω

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

等价于 $\exists c>0; n_0>0$，$\forall n\ge n_0$，使得$0\le cg(n)\le f(n)$。

3. Θ

$f(n)=\Theta(g(n))$ 等价于 $f(n)=\Theta(g(n))=\mathrm{O}(g(n))\cap\Omega(g(n))$

等价于 $\exists c_1,c_2>0; n_0>0$，$\forall n\ge n_0$，使得$c_1g(n)\le f(n)\le c_2g(n)$。

二、主定理推导

有一个形如$T(n)=aT(n/b)+f(n)$的递归式子，主定理就是关于该递归式子解的公式。

我们先给出该式子的解，再看其推导过程。

1. $T(n)=aT(n/b)+f(n)$的解

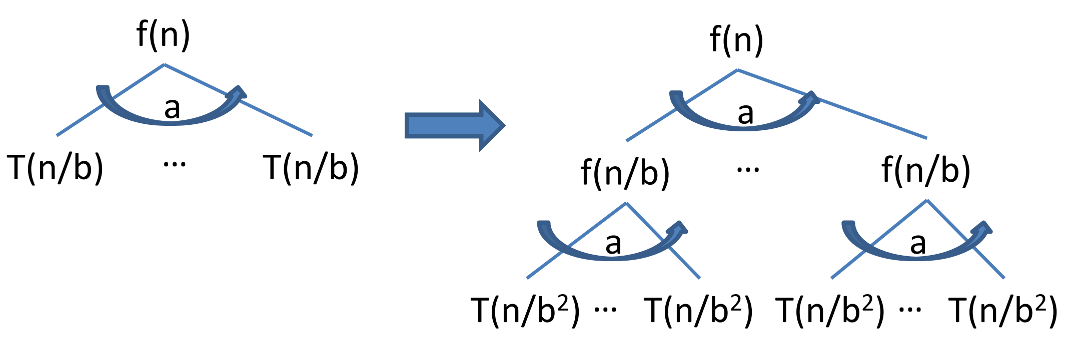
当 $f(n)=\mathrm{O}\left(n^{\log_b^a-\varepsilon}\right),\varepsilon>0$，则$T(n)=\Theta\left(n^{\log_b^a}\right)$

当 $f(n)=\Theta\left(n^{\log_b^a}\right)$，则$T(n)=\Theta\left((\lg n)n^{\log_b^a}\right)$

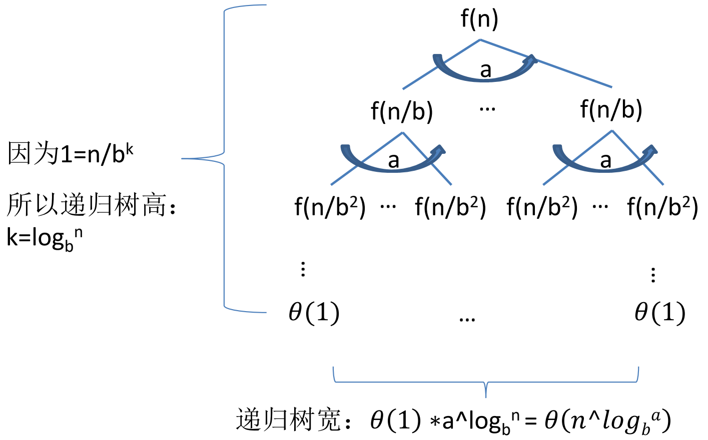
当 $f(n)=\Omega\left(n^{\log_b^a+\varepsilon}\right),\varepsilon>0$，及$af(n/b)\le(1-\varepsilon')f(n),\varepsilon'>0$，则$T(n)=\Theta(f(n))$

2. $T(n)=aT(n/b)+f(n)$解的推导

a. 首先我们可以将该式子画成**递归树**的形式，如下图：



以此类推，最终可以得到以下递归树：



根据该递归树，我们可以将递归公式写为如下形式：

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

b.首先证明当 $f(n)=\mathrm{O}\left(n^{\log_b^a-\varepsilon}\right),\varepsilon>0$，则 $T(n)=\Theta\left(n^{\log_b^a}\right)$

令 $g(n)=\sum_{i=0}^{\log_b^n-1} a^i f\left(n/b^i\right)$ ，由于 $f(n)=\mathrm{O}\left(n^{\log_b^a-\varepsilon}\right),\varepsilon>0$

则 $g(n)=\sum_{i=0}^{\log_b^n-1} a^i \mathrm{O}\left(\left(n/b^i\right)^{\log_b^a-\varepsilon}\right)$。

因为 $\exists c>0,\forall n\geq n_0, 0\leq f\left(n/b^i\right)\leq c\left(n/b^i\right)^{\log_b^a-\varepsilon}$，

则 $\exists c>0,\forall n\geq n_0, 0\leq a^i f\left(n/b^i\right)\leq ca^i\left(n/b^i\right)^{\log_b^a-\varepsilon}$，

所以 $0\leq g(n)\leq \sum_{i=0}^{\log_b^n-1} ca^i\left(n/b^i\right)^{\log_b^a-\varepsilon}$，

由于
$$\begin{aligned}&\sum_{i=0}^{\log_b^n-1} ca^i\left(n/b^i\right)^{\log_b^a-\varepsilon}=cn^{\log_b^a-\varepsilon}\sum_{i=0}^{\log_b^n-1}\left(b^\varepsilon\right)^i=cn^{\log_b^a-\varepsilon}\frac{1-n^\varepsilon}{1-b^\varepsilon}\\&=\frac{cn^{\log_b^a-\varepsilon}-cn^{\log_b^a}}{1-b^\varepsilon}=c'n^{\log_b^a}\end{aligned}$$，

所以 $g(n)=\mathrm{O}\left(n^{\log_b^a}\right)$。

$$T(n)=\mathrm{O}\left(n^{\log_b^a}\right)+\Theta\left(n^{\log_b^a}\right)=\Theta\left(n^{\log_b^a}\right)$$

c.其次证明当 $f(n)=\Theta\left(n^{\log_b^a}\right)$，则 $T(n)=\Theta\left((\lg n)n^{\log_b^a}\right)$

$\exists c_1, c_2 > 0, \forall n \ge n_0, c_1\left(n/b^i\right)^{\log_b^a} \le f\left(n/b^i\right) \le c_2\left(n/b^i\right)^{\log_b^a}$

$c_1 a^i\left(n/b^i\right)^{\log_b^a} \le a^i f\left(n/b^i\right) \le c_2 a^i\left(n/b^i\right)^{\log_b^a}$

$$\sum_{i=0}^{\log_b^n - 1} c_1 a^i\left(n/b^i\right)^{\log_b^a} \le \sum_{i=0}^{\log_b^n - 1} a^i f\left(n/b^i\right) \le \sum_{i=0}^{\log_b^n - 1} c_2 a^i\left(n/b^i\right)^{\log_b^a}$$

$$c_1 n^{\log_b^a} \log_b^n \le \sum_{i=0}^{\log_b^n - 1} a^i f\left(n/b^i\right) \le c_2 n^{\log_b^a} \log_b^n$$

$$c_1' n^{\log_b^a} \lg n \le \sum_{i=0}^{\log_b^n - 1} a^i f\left(n/b^i\right) \le c_2' n^{\log_b^a} \lg n$$

所以$T(n) = \Theta\left(n^{\log_b^a} \lg n\right) + \Theta\left(n^{\log_b^a}\right) = \Theta\left(n^{\log_b^a} \lg n\right)$

d.最后证明当 $f(n) = \Omega\left(n^{\log_b^a + \varepsilon}\right), \varepsilon > 0$，及 $af(n/b) \le (1-\varepsilon') f(n), \varepsilon' > 0$，则

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

首先由于 $g(n) = \sum_{i=0}^{\log_b^n - 1} a^i f\left(n/b^i\right) = f(n) + af(n/b) + \ldots \ge f(n)$，

所以 $g(n) = \Omega(f(n))$。

其次由于 $af(n/b) \le (1-\varepsilon') f(n), \varepsilon' > 0$，

所以 $a^2 f\left(n/b^2\right) \le a(1-\varepsilon') f(n/b) \le (1-\varepsilon')^2 f(n), \varepsilon' > 0$。

以此类推，$a^i f\left(n/b^i\right) \le (1-\varepsilon')^i f(n), \varepsilon' > 0$

则 $g(n) = \sum_{i=0}^{\log_b^n - 1} a^i f\left(n/b^i\right) \le \sum_{i=0}^{\log_b^n - 1} (1-\varepsilon')^i f(n)$，则 $g(n) = O(f(n))$。

综上，$g(n) = \Theta(f(n))$