关于主定理的证明

Proof of Master Theorem

网上关于主定理证明的材料不少，但是很多都只局限于在 b 的幂上的证明，将证明扩展到全体整数的论述相当少。本人查找了不少资料后在这里找到了一篇相对完整的证明(点击这里)，但细细读下来却发现它还是有一些瑕疵，因此打算自己补完整个证明，希望能对以后学习算法导论的同学有所帮助。由于本人数学水平不行，如果发现证明有错误或有地方需要补充，欢迎联系我 youth7@163.com

1，证明的总体思路

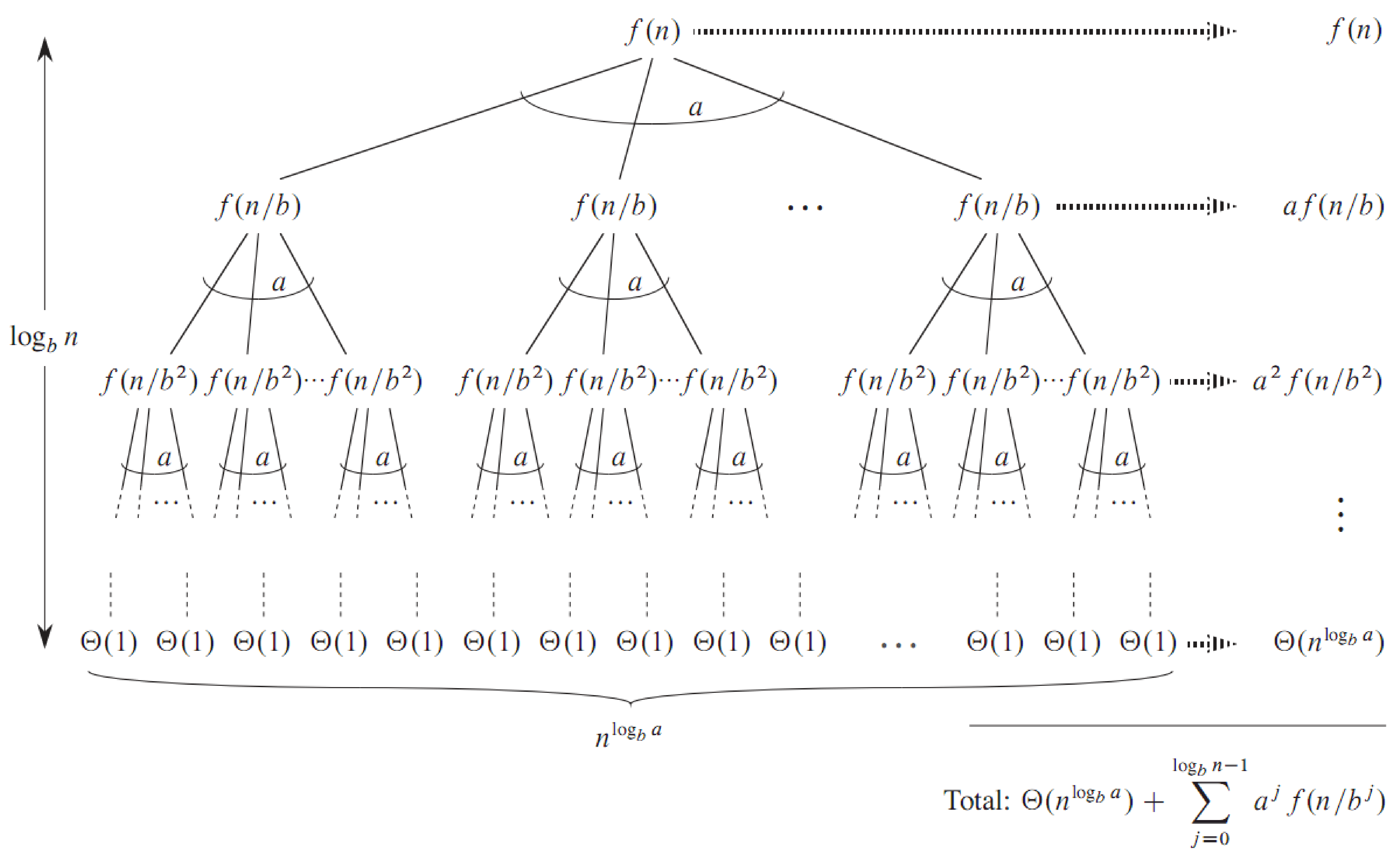
首先，求出递归式的非递归形式，即用多个多项式的和将递归式表示出来

接着，分析这个多项式的和，然后证明之。(证明内部又要分开两个阶段进行)

总之，求出递归式的和式，是证明的开始的基础，所有后续的证明，都要基于这个和式展开的。

2，递归式的和

首先画出递归树



这里介绍一下几个变量是怎么来的

1）树的高度 h

仔细观察 f(n)的 n，可以发现里面的变量变化规律是：$\frac{n}{b^0}, \frac{n}{b^1}, \frac{n}{b^2} \cdots\cdots \frac{n}{b^h}$ 。而在最后一层（也就是叶子层），问题规模已经变成了 1.因此有$\frac{n}{b^h} = 1$，所以 $h = \log_b n$

2）叶子节点的和

每层叶子节点的和变化相当有规律，就是$a^0, a^1, a^2 \ldots\ldots a^h$，因此，最后一层有叶子节点$a^{\log_b n} = n^{\log_b a}$个，这里我们需要运用到另外一个公式 $x^{\log_n y} = y^{\log_n x}$，它是由换底公式推导出来的，推导过程相当简单，这里不提。

3）因此递归式最终变成了两部分，**第一部分是叶子节点的和**，它的总代价是$n^{\log_b a}$，**第二部分是除叶子节点外，各层节点的和。**

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

这里要注意内层节点是从第 0 层开始求到倒数最后一层（即 h-1 层），因此是$(\log_b n) - 1$，算法导论上印刷这个地方不是很清晰，有误导嫌疑，因此加上括号就非常明确了。

这里需要重点理解的是，从这个和式我们可以直观地感受到，**如果两个部分中谁的阶数较高，则谁决定和式的上界**。因此接着的具体证明就是围绕着这两部分的大小关系展开的。我们有以下几种情况：

(留意 f(n)是划分当前问题的代价，也是非叶子节点的代价。$\Theta(n^{\log_b a})$是所有叶子节点的代价，即基本情况的代价的和)

1）如果第一部分比第二部分的阶数要高，这意味着递归树的**总代价由叶子的代价决定**

2）如果两部分相等，这意味着递归树的**总代价分布均匀，由叶子节点和其它节点共同决定**。

3）如果第二部分阶数比第一部分要高，这意味着递归树的总代价由内层叶子决定，也即是说，划分问题的**代价决定递归树的总代价**

理解三种情形的现实意义很重要，它能帮助我们清晰体会到递归式的本质

3，证明的第 1 阶段

第一阶段的证明并不是在全体自然数上进行的，而是将 n 定义在 b 的幂上面，即 n=$b^0, b^1, b^2 \dots\dots b^k$,以下是证明过程

一阶段情况 1

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

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

$$\therefore f\left(\frac{n}{b^j}\right) = O\left(\frac{n}{b^j}^{(\log_b a)-\varepsilon}\right)$$ //用$\frac{n}{b^j}$代入

$$\therefore f\left(\frac{n}{b^j}\right) \le c\left(\frac{n}{b^j}\right)^{(\log_b a)-\varepsilon}$$

$$\therefore\ a^j f\left(\frac{n}{b^j}\right) \le a^j c\left(\frac{n}{b^j}\right)^{(\log_b a)-\varepsilon}$$ //两边同时乘以a^j

$$\therefore a^j f\left(\frac{n}{b^j}\right) \le cn^{(\log_b a)-\varepsilon} \cdot \left(\frac{a}{b^{(\log_b a)-\varepsilon}}\right)^j = cn^{(\log_b a)-\varepsilon}(b^\varepsilon)^j$$ ②//将关于 j 的项合并，常数项合并，这样做事为了方便求和

由①和②可得

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

$$\therefore g(n) \le cn^{(\log_b a)-\varepsilon} \cdot \frac{1(1-(b^\varepsilon)^{\log_b n})}{1-b^\varepsilon}$$ //等比数列求和，总共有$\log_b n$项目，首项是 1

$$\therefore g(n) \le cn^{(\log_b a)-\varepsilon}\frac{1-n^\varepsilon}{1-b^\varepsilon} = \frac{c}{1-b^\varepsilon}[n^{(\log_b a)-\varepsilon} - n^{(\log_b a)}]$$

//上式体现到了为什么 f(n)要多项式地小于$n^{(\log_b a)-\varepsilon}$，因为只有这样，才能看出谁才是高阶项。如果没有这个条件，无法决定谁是高阶项

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

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

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

一阶段情况 2

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

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

$$\therefore\ f\left(\frac{n}{b^j}\right) = \Theta\left(\left(\frac{n}{b^j}\right)^{\log_b a}\right)$$ //用$\frac{n}{b^j}$代入

$\therefore c_1\left(\frac{n}{b^j}\right)^{\log_b a} \le f\left(\frac{n}{b^j}\right) \le c_2\left(\frac{n}{b^j}\right)^{\log_b a}$

$\therefore a^j c_1\left(\frac{n}{b^j}\right)^{\log_b a} \le a^j f\left(\frac{n}{b^j}\right) \le a^j c_2\left(\frac{n}{b^j}\right)^{\log_b a}$ //同时乘以a^j

$\therefore c_1 n^{(\log_b a)}\left(\frac{a}{b^{\log_b a}}\right)^j \le a^j f\left(\frac{n}{b^j}\right) \le c_2 n^{(\log_b a)}\left(\frac{a}{b^{\log_b a}}\right)^j$ //合并关于 j 的项

$\therefore c_1 n^{(\log_b a)}(1)^j \le a^j f\left(\frac{n}{b^j}\right) \le c_2 n^{(\log_b a)}(1)^j$

$\therefore c_1 n^{(\log_b a)} \log_b n \le g(n) \le c_2 n^{(\log_b a)} \log_b n$//等差数列求和，总共有$\log_b n$项，每项的值都相同

$\therefore c_1 n^{(\log_b a)} \frac{\lg n}{\ln b} \le g(n) \le c_2 n^{(\log_b a)} \frac{\lg n}{\ln b}$//换底公式

$\therefore \frac{c_1}{\lg b} n^{(\log_b a)} \lg n \le g(n) \le \frac{c_2}{\lg b} n^{(\log_b a)} \lg n$

$\because \frac{c_1}{\lg b}$与$\frac{c_2}{\lg b}$都是常数，令$c_3 = \frac{c_1}{\lg b}$ ，$c_4 = \frac{c_2}{\lg b}$ ，则有$c_3 n^{(\log_b a)} \lg n \le g(n) \le c_4 n^{(\log_b a)} \lg n$

$\therefore g(n) = \Theta(n^{(\log_b a)} \lg n)$

$\therefore T(n) = \Theta(n^{\log_b a}) + \Theta(n^{(\log_b a)} \lg n)$ //此时$n^{(\log_b a)} \lg n$是高阶项

$\therefore T(n) = \Theta(n^{(\log_b a)} \lg n)$

一阶段情况 3

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

$\because af\left(\frac{n}{b}\right) \le cf(n)$

$\therefore a^j f\left(\frac{n}{b^j}\right) \le c^j f(n)$ //用a^j b^j c^j 直接代入，不够严谨，严谨的推导是循环展开，有兴趣可以自己用搜索下英文的资料，这里不详细讨论

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

$\therefore g(n) = O(f(n))$ ③

此时根据$g(n)$的定义将其展开，则有

$g(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + \cdots a^{(\log_b n)-1} f\left(\frac{n}{b^{(\log_b n)-1}}\right) \ge f(n)$

$\therefore g(n) = \Omega(f(n))$ ④

由③和④可得

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

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

又$\because f(n) = O(n^{(\log_b a)+\varepsilon})$ //说明$f(n)$是高阶项，比$n^{(\log_b a)}$的阶数要高，因此$f(n)$将决定整体的代价

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

综合上述，第一阶段证明完毕

4，证明的第 2 阶段

第二阶段是在第一阶段的基础上，将主定理的定义域由b^n扩展到从全体自然数上，但是有两点必须注意：

1，当定义域扩展到全体实数时，主定理的表达式不是完全正确的，例如用归并排序对 101 个整数进行排序，第一次递归的时候会将拆成两个子过程，一个包含 50 个元素，另外一个包含 51 个元素，因此表达式应该是$T(100) = T(\lceil 100/2\rceil) + T(\lfloor 100/2\rfloor)$,这和主定理的定义是不一致的。

2，它的思路是无论证明递归式向上或向下取整的时候，它都有确定的上下界，即

$\boldsymbol{T(n) = aT(\lceil \frac{n}{b}\rceil) + f(n)}$**和**$\boldsymbol{T(n) = aT(\lfloor \frac{n}{b}\rfloor) + f(n)}$ **都有上下界** ⑤

在开始之前我们先要证明另外一个很重要的结论$(log_b\ a) - \varepsilon \geq 0$，这对完善证明至关重要，但是目前为止在网上提供的资料中并没有见有所提及。

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

$$\therefore f(n)\text{渐近小于}cn^{(log_b\ a)-\varepsilon}$$

此时假设$(log_b\ a) - \varepsilon < 0$，则$n^{(log_b\ a)-\varepsilon}$是减函数，当 n 趋向于正无穷的时候，其极限是 0.

由上可得推论：$f(n)$渐近小于 0,。

然而由书上的题设可知，**$f(n)$是一个大于 0 的函数**，因此$f(n)$渐近小于 0 与题设矛盾，**所以假设$(log_b a) - \varepsilon < 0$是错误的**，所以$\boldsymbol{(log_b a) - \varepsilon \geq 0}$。

以下开始正式证明第二阶段

二阶段情况 1

令 $g(n) = \sum_{j=0}^{\lfloor log_b\ n\rfloor - 1} a^j f\left(\left\lceil \frac{n}{b^j}\right\rceil\right)$ //其实这里还需要讨论向下取整的情况，后续的证明可以见到无论向上还是向下去整都是可以被证明的

//此时树的高度已经变成了$\lfloor log_b\ n\rfloor - 1$，书上有证明在深度为$\lfloor log_b\ n\rfloor - 1$的时候，问题的规模最多为常数，这个结论同时适合于向上、向下取整两种情况

$$\because f(n_j) \leq c(\left\lceil \frac{n}{b^j}\right\rceil)^{(log_b\ a)-\varepsilon} \leq (\frac{n}{b^j} + 1)^{(log_b\ a)-\varepsilon} \quad ⑥$$

//注意这个不等式无论是是对$\frac{n}{b^j}$向上或者向下取整，**不等式依然成立**，所以由这一步推导出来的结论，能够满足⑤的两种情况

$$\therefore c(\frac{n}{b^j} + 1)^{(log_b\ a)-\varepsilon} = c\left[\frac{n}{b^j}\left(1 + \frac{b^j}{n}\right)^{(log_b\ a)-\varepsilon}\right] = c\left(\frac{n}{b^j}\right)^{(log_b\ a)-\varepsilon}(1 + \frac{b^j}{n})^{(log_b\ a)-\varepsilon} \text{//一个巧妙的和化积}$$

$$\because 1 + \frac{b^j}{n} \leq 2 \text{//因为}\frac{b^j}{n} \leq 1$$

$$\therefore c\left(\frac{n}{b^j}\right)^{(log_b\ a)-\varepsilon}\left(1 + \frac{b^j}{n}\right)^{(log_b\ a)-\varepsilon} \leq c\left(\frac{n}{b^j}\right)^{(log_b\ a)-\varepsilon} 2^{(log_b\ a)-\varepsilon} \quad ⑦$$

由⑥与⑦可得

$$f(n_j) \leq c\left(\frac{n}{b^j}\right)^{(log_b\ a)-\varepsilon} . 2^{(log_b\ a)-\varepsilon}$$

$\therefore$

$$a^j f(n_j) \leq a^j c\left(\frac{n}{b^j}\right)^{(log_b\ a)-\varepsilon} . 2^{(log_b\ a)-\varepsilon} = c(2n)^{(log_b\ a)-\varepsilon}\left[\frac{a}{b^{(log_b\ a)-\varepsilon}}\right]^j = c(2n)^{(log_b\ a)-\varepsilon}(b^\varepsilon)^j$$

$$\therefore\ g(n) \leq \sum_{j=0}^{\lfloor log_b\ n\rfloor - 1} c(2n)^{(log_b\ a)-\varepsilon}(b^\varepsilon)^j = c(2n)^{(log_b\ a)-\varepsilon}\sum_{j=0}^{\lfloor log_b\ n\rfloor - 1}(b^\varepsilon)^j$$

$$\therefore g(n) \leq c(2n)^{(log_b\ a)-\varepsilon} . \frac{1-(b^\varepsilon)^{\lfloor log_b\ n\rfloor}}{1-b^\varepsilon} \leq c(2n)^{(log_b\ a)-\varepsilon} . \frac{1-(b^\varepsilon)^{log_b\ n}}{1-b^\varepsilon} \text{//上式等比数列求和}$$

$$\therefore g(n) \leq c(2n)^{(log_b\ a)-\varepsilon} . \frac{1-(b^\varepsilon)^{log_b\ n}}{1-b^\varepsilon} = c(2n)^{(log_b\ a)-\varepsilon} . \frac{1-n^\varepsilon}{1-b^\varepsilon}$$

$$\therefore g(n) \leq \frac{c.2^{(log_b\ a)-\varepsilon}}{1-b^\varepsilon} . (n^{(log_b\ a)-\varepsilon} - n^{log_b\ a}) \text{// } f(n) = O(n^{(log_b\ a)-\varepsilon})\text{在此体现了价值}$$

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

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

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

二阶段情况 2

令 $g(n) = \sum_{j=0}^{\lfloor \log_b n \rfloor - 1} a^j f\left(\left\lceil \frac{n}{b^j} \right\rceil\right)$ //这里只是向上取整，严格的证明应该包含向上向下取整两种情况，先讨论向上这种情况

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

$\therefore f\left(\left\lceil \frac{n}{b^j} \right\rceil\right) = \Theta(\left\lceil \frac{n}{b^j} \right\rceil^{\log_b a})$ //用$\frac{n}{b^j}$代入

$\therefore c_1 \left\lceil \frac{n}{b^j} \right\rceil^{\log_b a} \le f\left(\left\lceil \frac{n}{b^j} \right\rceil\right) \le c_2 \left\lceil \frac{n}{b^j} \right\rceil^{\log_b a}$

$\therefore a^j c_1 \left\lceil \frac{n}{b^j} \right\rceil^{\log_b a} \le a^j f\left(\left\lceil \frac{n}{b^j} \right\rceil\right) \le a^j c_2 \left\lceil \frac{n}{b^j} \right\rceil^{\log_b a}$ //同时乘以a^j

又$\because a^j c_2 \left\lceil \frac{n}{b^j} \right\rceil^{\log_b a} \le a^j c_2 (1 + \frac{n}{b^j})^{\log_b a} = a^j c_2 (\frac{n}{b^j})^{\log_b a} (1 + \frac{b^j}{n})^{\log_b a} \le a^j c_2 (\frac{n}{b^j})^{\log_b a} 2^{\log_b a}$

$\therefore a^j c_1 \left\lceil \frac{n}{b^j} \right\rceil^{\log_b a} \le a^j f\left(\left\lceil \frac{n}{b^j} \right\rceil\right) \le a^j c_2 (\frac{n}{b^j})^{\log_b a} 2^{\log_b a}$

又$\because a^j c_2 (\frac{n}{b^j})^{\log_b a} 2^{\log_b a} = c_2 2^{\log_b a} n^{\log_b a} (\frac{a}{b^{\log_b a}})^j = c_2 2^{\log_b a} n^{\log_b a} (1)^j$

$$\therefore \sum_{j=0}^{\lfloor \log_b n \rfloor - 1} a^j c_1 \left\lceil \frac{n}{b^j} \right\rceil^{(\log_b a)} \le g(n) \le \sum_{j=0}^{\lfloor \log_b n \rfloor - 1} c_2 2^{\log_b a} n^{\log_b a} (1)^j \le c_2 2^{\log_b a} n^{\log_b a} \log_b n$$

$$\because g(n) \ge \sum_{j=0}^{\lfloor \log_b n \rfloor - 1} a^j c_1 \left\lceil \frac{n}{b^j} \right\rceil^{\log_b a} \ge \sum_{j=0}^{\lfloor \log_b n \rfloor - 1} a^j c_1 \left(\frac{n}{b^j}\right)^{\log_b a} \ge c_1 n^{\log_b a} \sum_{j=0}^{\lfloor \log_b n \rfloor - 1} \left(\frac{a}{b^{\log_b a}}\right)^j$$

$$= \log_b(n-1)\, c_1 n^{\log_b a}$$

$\therefore c_1 n^{\log_b a} \log_b(n-1) \le g(n) \le c_2 2^{\log_b a} n^{\log_b a} \log_b n$

$\therefore c_1 n^{\log_b a} \frac{\lg n-1}{\lg b} \le g(n) \le c_2 2^{\log_b a} n^{\log_b a} \frac{\lg n}{\lg b}$ //换底公式

$\therefore g(n) = \Theta(n^{\log_b a} \lg n)$

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

$\therefore T(n) = \Theta(n^{\log_b a} \lg n)$ //向上取整证明完毕

//以下是对向下取整时候的证明

当向下取整时，与向上取整类似(只需要将向上取整的符号改成向下取整)，我们容易求得 T(n)上界是

$$g(n) \le \sum_{j=0}^{\lfloor \log_b n \rfloor - 1} c_2 2^{\log_b a} n^{\log_b a} (1)^j \le c_2 2^{\log_b a} n^{\log_b a} \log_b n$$

$\therefore g(n) = O(n^{\log_b a} \lg n)$

$\therefore T(n) = \Theta(n^{\log_b a}) + O(n^{\log_b a} \lg n)$

$\therefore T(n) = O(n^{\log_b a} \lg n)$

类似，下界是

$$g(n) \geq \sum_{j=0}^{\lfloor log_b n \rfloor - 1} a^j c_1 \left\lfloor \frac{n}{b^j} \right\rfloor^{(log_b a)} \geq \sum_{j=0}^{\lfloor log_b n \rfloor - 1} a^j c_1 \left(\frac{n}{\mathbf{2} b^j} \right)^{(log_b a)}$$

//上面这个不等式放缩想了好长时间，最后还是受到后面章节一些证明的启发最后才想出来的，本人愚钝，呵呵

$$\therefore g(n) \geq c_1 \left(\frac{n}{2}\right)^{log_b a} \sum_{j=0}^{\lfloor log_b n \rfloor - 1} \left(\frac{a}{b^{log_b a}}\right)^j \geq log_b(n-1)\, c_1 \left(\frac{n}{2}\right)^{log_b a}$$

$\therefore g(n) = \Omega(n^{log_b a}\, lg\, n)$ //参照上面证明使用换底公式可得此结果

又$\because T(n) = O(n^{log_b a}\, lg\, n)$

$\therefore T(n) = \Theta(n^{log_b a}\, lg\, n)$ //向上取整证明完毕

二阶段情况 3

令 $g(n) = \sum_{j=0}^{\lfloor log_b n \rfloor - 1} a^j f\left(\left\lceil \frac{n}{b^j} \right\rceil\right)$ //其实这里还可以向下去整，其实下面的证明适用于向上和向下取整，只要将取整符号换成向下取整即可

$\because af(\left\lceil \frac{n}{b} \right\rceil) \leq cf(n)$

$\therefore a^j f(\left\lceil \frac{n}{b^j} \right\rceil) \leq c^j f(n)$ //用a^j，$b^j c^j$直接代入，但严谨的做法是代入循环推导，这里不详细描述 ，网上英文资料有关于这步的详细推导

$$\therefore\ g(n) \leq f(n) \sum_{j=0}^{(log_b n)-1} c^j \leq f(n) \sum_{j=0}^{\infty} c^j = \frac{1}{1-c} f(n)$$

$\therefore g(n) = O(f(n))$ ⑧

此时根据$g(n)$的定义将其展开，则有

$$g(n) = f(n) + af\left(\frac{n}{b^1}\right) + a^2 f(\frac{n}{b^2}) + \cdots a^{\lfloor log_b n \rfloor - 1} f(\frac{n}{b^{\lfloor log_b n \rfloor - 1}}) \geq f(n)$$

$\therefore g(n) = \Omega(f(n))$ ⑨

由⑧和⑨可得

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

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

又$\because f(n) = O(n^{(log_b a)+\varepsilon})$ //说明$f(n)$是高阶项，比$n^{(log_b a)}$的阶更高

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

综合上述，第二阶段证明完毕

5，扩展

我们可以看到，其实主定理是不能覆盖所有情况的，主定理的本质是递归式，递归式的求解有很长历史，算法导论的本章注记中谈到了 Akra-Bazzi 方法(个人认为是一种关于递归式的普遍性解法)，有兴趣可以参考一下附录参考资料 228，这种方法能够解决任何递归问题。

Tom Leighton. Notes on better master theorems for divide-and-conquer recurrences. Class notes. Available at http://citeseer.ist.psu.edu/252350.html, October 1996.