第四章 递归式

这一章内容相对比较简单和轻松，最近忙于考试，六级，而且还生了场小病，我的学习计划已经比我原定的目标落后不少了，因此只能将一些不是不要的内容先跳过，比如，打*号的和一些比较难得思考题，等到有时间回过来修正笔记的时候再加上。第一部分现在就剩下第五章了，等我们把这些基础（比较枯燥，但还算是有点意思）都学完，就可以开始富有挑战性的算法导论之旅了。顺便说下我的学习计划，暑假打算把算法导论除了最后一部分外的所有章节都看完（至少是基本看完），最后一部分是算法问题选编，等有时间时再慢慢研究，反正，我不会放弃把算法导论整本书做成读书笔记的想法。然后就是学英语，呵呵，想尽外企，英语不好不行的，但是我目前的英语真是不容乐观。在，这个算法导论读书笔记告一段落后，我将开始学习设计模式，到时候，也打算以读书笔记的形式与大家分享、交流。

顺便说下我的感受，这第一部分读下来，深深的感觉到，要做个好的程序员一定要熟知算法和数据结构，但是如果要做个好的计算机专家，特别是大师，那么数学知识可是相当重要啊。学习算法是为了当一个优秀的程序员做准备（虽然大学刚毕业有可能找不到理想的职位，但是，人总要往上看的，要随时做好准备迎接机会和挑战）；学好数学是为自己成为大师做积累。一个不想当将军的兵不是好兵，同样，一个永远不梦想自己有朝一日能成为大师级人物的程序员，也不是个真正好的程序员 :)。

高中三年，没好好读书，结果高三用了半年突击上了一所二本，虽然有所遗憾，但是，有得也必有失，来了大学后先是失望，再是失落，到后悔，最后再到重整信心，迎接挑战，我比以往任何时候都珍惜机会和时间，每当感到自己又重新充满激情的时候，我就觉得我的天空特别蓝，特别灿烂，大家祝我好运吧！我也祝大家平平安安，事业顺利！

1. 三种解递归式的基本方法：**代换法**(substitution method)，**递归树方法**(recursion-tree method)，**主方法**(master method)

代换法：

步骤：

1） 猜测解的形式

2） 用数学归纳法找出使解真正有效的常数

"代换法"这一名称源于当归纳假设用较小值，用所猜测的值代替函数的解。这种方法很有效，但是只能用于解的形式很容易猜的情况，不存在通用的方法来猜测递归式的正式解。

在运用数学归纳法的证明中，应当避开一些比较低的特殊边界，这个可以设一个常数n_0，只要求对$n \geqslant n_0$时能够得证即可。

当递归式与先前见过的类似，则可猜测递归式有类似的解。猜测的另一种方法是先证明较松的上下界，然后逐步降低上界，提高下界。

一些细微的问题与陷阱

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$$

猜测其解为$O(n)$，即要证明对适当选择的c，有$T(n) \leqslant cn$。当$T(\lfloor n/2 \rfloor) \leqslant c\lfloor n/2 \rfloor$,$T(\lceil n/2 \rceil) \leqslant c\lceil n/2 \rceil$时，则$T(n) = c\lfloor n/2 \rfloor + c\lceil n/2 \rceil + 1 = cn + 1$。

最终的形式总是比我们猜测的界大一个低阶量。解决这个问题的方法是从**猜测项中减去一个低阶项**。反之，则加上一个低阶项。

假设$T(n) \leqslant cn - b, b \geqslant 0$,b是常数。当$T(\lfloor n/2 \rfloor) \leqslant c\lfloor n/2 \rfloor - b$,$T(\lceil n/2 \rceil) \leqslant c\lceil n/2 \rceil - b$时，则$T(n) = c\lfloor n/2 \rfloor - b + c\lceil n/2 \rceil - b + 1 = cn - 2b + 1 \leqslant cn - b$，只要使$b \geqslant 1$，就存在常数$c$使不等式成立。这样就证明了假设。在这里，说一下陷阱，在证明的最后一步一定要做大符合**归纳假设的准确形式。**

如在对$T(n) = 2T(\lfloor n/2 \rfloor) + n$作出猜测$T(n) \leqslant cn$后，证明

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

即说明证明了问题。这个是**错误**的。

改变变量（作为技巧）

考虑$T(n) = 2T(\sqrt{n}) + \lg n$

为了方便，不考虑数的截取整数问题，如将$\sqrt{n}$化为整数，设$\lg n = m$，这样将原问题转化为$T(2^m) = 2T(2^{m/2}) + m$,再令$S(m) = T(2^m)$,做替换$S(m) = 2S(m/2) + m$，通过构造递归树可以得到$S(m) = O(m \lg m)$，故

$$T(n) = T(2^m) = S(m) = O(m \lg m) = O(\lg n \lg \lg n)$$
$$(\because n = 2^m, S(m) = T(2^m), m = \lg n)$$

一点补充

$$\frac{n}{2} \leqslant \lceil n/2 \rceil \leqslant \frac{n+1}{2}$$
$$\frac{n}{2} \geqslant \lfloor n/2 \rfloor \geqslant \frac{n-1}{2}$$

另外,书上的另一句话也很重要:在表示并解递归式时,常忽略上取整、下取整以及边界条件。

练习

4.1-1：（1-4 解答时没有忽略上下取整条件）

设$T(n) \leqslant c \lg n$，当$T(\lceil n/2 \rceil) \leqslant c \lg(\lceil n/2 \rceil)$，

$$\begin{aligned}
T(n) &\leqslant c \lg(\lceil n/2 \rceil) + 1 \\
&\leqslant c \lg\left(\frac{n+1}{2}\right) + 1 \\
&= c \lg(n+1) - c + 1 \\
&\leqslant c \lg n
\end{aligned}$$
$$\Longleftrightarrow c \lg(n+1) - c \lg 2 - c \lg n + 1 \leqslant 0$$
$$\Longleftrightarrow c \lg\left(\frac{n+1}{2n}\right) + 1 \leqslant 0$$

取$n_0 = 2$，当$n \geqslant n_0$时（取适当的n减低边界困难）：

$\dfrac{n+1}{2n} = \dfrac{1+\frac{1}{n}}{2}$，在正的区间上，是单调递增的，故

$$c \lg\left(\frac{n+1}{2n}\right) + 1 \leqslant c \lg\left(\frac{3}{4}\right) + 1 \leqslant 0$$

$\lg\left(\dfrac{3}{4}\right) \approx -0.4$，故取$c = 3$即可。证毕。

4.1-2：

设$T(n) \leqslant c_1 n \lg n$，当$T(\lfloor n/2 \rfloor) \leqslant c_1 \lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)$，

$$
\begin{aligned}
T(n) &\leqslant 2c_1\lfloor n/2\rfloor \lg(\lfloor n/2\rfloor)+n\\
&\leqslant 2c_1\cdot\frac{n}{2}\cdot\lg(\frac{n}{2})+n\\
&=c_1 n\lg(\frac{n}{2})+n\\
&\leqslant c_1 n\lg n
\end{aligned}
$$

$$\Longleftrightarrow c_1 n\lg(\frac{n}{2})-c_1 n\lg n+n\leqslant 0$$

$$\Longrightarrow c_1 n\lg\left(\frac{\frac{n}{2}}{n}\right)+n\leqslant 0$$

$$\Longrightarrow c_1 n\lg\left(\frac{1}{2}\right)+n\leqslant 0$$

$$\Longrightarrow -c_1 n+n\leqslant 0$$

$$\Longrightarrow c_1\geqslant 1$$

取$c_1=1$即可满足。所以$T(n)=O(n\lg n)$。

设$T(n)\geqslant c_2 n\lg n$，当$T(\lfloor n/2\rfloor)\geqslant c_2\lfloor n/2\rfloor\lg(\lfloor n/2\rfloor)$，

$$
\begin{aligned}
T(n) &\geqslant 2c_2\lfloor n/2\rfloor \lg(\lfloor n/2\rfloor)+n\\
&\geqslant 2c_2\frac{n-1}{2}\lg\left(\frac{n-1}{2}\right)+n\\
&=c_2(n-1)\lg\left(\frac{n-1}{2}\right)+n\\
&\geqslant c_2 n\lg n
\end{aligned}
$$

$$\Longleftrightarrow c_2(n-1)\lg(n-1)-c_2(n-1)+n\geqslant c_2 n\lg n\geqslant c_2 n\lg(n-1)$$

$$\Longrightarrow c_2\lg(n-1)+c_2(n-1)\leqslant n$$

$$\Longrightarrow c_2\leqslant\frac{n}{\lg(n-1)+(n-1)}<\frac{n}{(n-1)+(n-1)}<\frac{n-1}{2n-2}=\frac{1}{2}$$

取$c_2=\frac{1}{4}$即可满足。所以$T(n)=\Omega(n\lg n)$

因此得到$T(n)=\Theta(n\lg n)$

4.1-3：

从猜测项中加上一个低阶项，猜测$T(n)\leqslant cn\lg n+n$，则当$T(\lfloor n/2\rfloor)\leqslant c\lfloor n/2\rfloor\lg(\lfloor n/2\rfloor)+\lfloor n\rfloor$时：

$$
\begin{aligned}
T(n) &\leqslant 2\left(c\lfloor n/2\rfloor \lg(\lfloor n/2\rfloor) + \lfloor n/2\rfloor\right) + n \\
&\leqslant 2c\cdot\frac{n}{2}\cdot\lg(\frac{n}{2}) + 2\cdot\frac{n}{2} + n \\
&= cn\lg(\frac{n}{2}) + 2n \\
&\leqslant cn\lg n + n
\end{aligned}
$$

$$
\begin{aligned}
&\Longleftrightarrow cn\lg n + n - cn\lg(\frac{n}{2}) - 2n \geqslant 0 \\
&\Longrightarrow cn\lg n + n - cn\lg n + cn - 2n \geqslant 0 \\
&\Longrightarrow cn - n \geqslant 0 \\
&\Longrightarrow c \geqslant 1
\end{aligned}
$$

此时已经克服了在$T(1) = 1$处的边界困难。

4.1-4：

$$
\begin{aligned}
T(n) &\leqslant c_2\lceil n/2\rceil \lg(\lceil n/2\rceil) + c_2\lfloor n/2\rfloor \lg(\lfloor n/2\rfloor) + \Theta(n) \\
&\leqslant c_2\cdot\frac{n+1}{2}\lg\left(\frac{n+1}{2}\right) + c_2\cdot\frac{n}{2}\lg\left(\frac{n}{2}\right) + \Theta(n) \\
&= \frac{n+1}{2}c_2\lg(n+1) - \frac{n+1}{2}c_2 + \frac{n}{2}c_2\lg n - \frac{n}{2}c_2 + \Theta(n) \\
&\leqslant c_2 n\lg n
\end{aligned}
$$

$$
\begin{aligned}
&\Longleftrightarrow \frac{n}{2}c_2\lg n - \frac{n+1}{2}c_2\lg(n+1) + \frac{n+1}{2}c_2 + \frac{n}{2}c_2 - \Theta(n) \geqslant 0 \\
&\quad \because \frac{n}{2}c_2\lg n - \frac{n+1}{2}c_2\lg(n+1) \leqslant 0 \\
&\quad \therefore \frac{n+1}{2}c_2 + \frac{n}{2}c_2 - \Theta(n) \geqslant nc_2 - \Theta(n) \geqslant 0 \Longrightarrow\ c_2 \geqslant \Theta(1)
\end{aligned}
$$

取$c_2 = \Theta(1)$,所以$T(n) = O(n\lg n)$

$$
\begin{aligned}
T(n) &\geqslant c_1\lceil n/2\rceil \lg(\lceil n/2\rceil) + c_1\lfloor n/2\rfloor \lg(\lfloor n/2\rfloor) + \Theta(n) \\
&\geqslant c_1\frac{n}{2}\lg(\frac{n}{2}) + c_1\frac{n-1}{2}\lg(\frac{n-1}{2}) + \Theta(n) \\
&= \frac{n}{2}c_1\lg n - \frac{n}{2}c_1 + \frac{n-1}{2}c_1\lg(n-1) - \frac{n-1}{2}c_1 + \Theta(n) \\
&\geqslant c_1 n\lg n
\end{aligned}
$$

$$
\begin{aligned}
&\Longleftrightarrow \frac{n-1}{2}c_1\lg(n-1) - \frac{n}{2}c_1\lg n - \frac{n}{2}c_1 - \frac{n-1}{2}c_1 + \Theta(n) \geqslant 0 \\
&\quad \because \frac{n-1}{2}c_1\lg(n-1) - \frac{n}{2}c_1\lg n \leqslant 0 \\
&\Longrightarrow \Theta(n) \geqslant \frac{n}{2}c_1 + \frac{n-1}{2}c_1 \geqslant nc_1 \Longrightarrow\ c_1 \leqslant \Theta(1)
\end{aligned}
$$

取$c_1 = \Theta(1)$,所以$T(n) = \Omega(n\lg n)$

综上,$T(n) = \Theta(n\lg n)$

4.1-5：

如果$T(n)=O(n\lg n)$,则$T(n+17)=O(n\lg n)$,故若$T(\lfloor n/2\rfloor)=O(\lfloor n/2\rfloor\lg(\lfloor n/2\rfloor))$的话也有$T(\lfloor n/2\rfloor+17)=O(\lfloor n/2\rfloor\lg(\lfloor n/2\rfloor))$,即$T(\lfloor n/2\rfloor)$和$T(\lfloor n/2\rfloor+17)$的效率是渐近一致的。

因此，若$T(n)\leqslant cn\lg n$，则$T(\lfloor n/2\rfloor+17)=T(n/2+17)=T(n/2)=c\cdot\frac{n}{2}\lg(\frac{n}{2})$

$$
\begin{aligned}
T(n)\leqslant&2\times c\cdot\frac{n}{2}\lg\left(\frac{n}{2}\right)+n\\
=&cn\lg\left(\frac{n}{2}\right)+n\\
=&cn\lg n-cn+n\\
\leqslant&cn\lg n\\
\iff cn-n\geqslant 0 \iff& c-1\geqslant 0 \iff c\geqslant 1
\end{aligned}
$$

故取$c=1$即可。

4.1-6：

令$m=\lg n \iff n=2^m$

原问题变成$T\left(2^m\right)=2T\left(\frac{1}{2}\times 2^m\right)+1$

再令$T\left(2^m\right)=S(m)$，则问题变成$S(m)=2S(m/2)+1$，可知$S(m)=\Theta(m\lg m)$

$\therefore T\left(n\right)=T\left(2^m\right)=S\left(m\right)=\Theta\left(m\lg m\right)=\Theta\left(\lg n\lg\lg n\right)$

递归树方法是作出良好猜测的有效方法。一般来说，经常使用递归树方法做出好的猜测，然后再利用代换法加以证明。

例：为递归式$T(n)=3T(\lfloor n/4\rfloor)+\Theta(n^2)$提供良好的猜测

考虑递归式$T(n)=3T(n/4)+cn^2$的递归树。

（请参照书本）深度为i的结点，其子问题大小为$n/4^i$，那么当$n/4^i=1 \iff i=\log_4 n$时，子问题大小达到 1。这棵树有$\log_4 n+1$层。在深度i时，结点数目是3^i，每个结点代价是$c(n/4^i)^2$，则第$i(i=0,1,2\cdots)$层总代价是$3^i c(n/4)^2=(3/16)^i cn^2$。在最后一层，

即深度为$\log_4 n$时，有$3^{\log_4 n} = n^{\log_4 3}$个结点，每个节点代价是$T(1)$,故总代价$\Theta(n^{\log_4 3})$。

将各层大家相加得到递归树总代价：

$$\begin{aligned}
T(n) =& cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots + \left(\frac{3}{16}\right)^{\log_4 n-1} cn^2 + \Theta(n^{\log_4 3}) \\
=& \sum_{i=0}^{\log_4 n-1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
=& \frac{(3/16)^{\log_4 n} - 1}{(3/16) - 1} cn^2 + \Theta(n^{\log_4 3})
\end{aligned}$$

用一个无穷等比递减级数作为上界：

$$\begin{aligned}
T(n) =& \sum_{i=0}^{\log_4 n-1} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
<& \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + \Theta(n^{\log_4 3}) \\
=& \frac{1}{1-(3/16)} cn^2 + \Theta(n^{\log_4 3}) \\
=& \frac{16}{13} cn^2 + \Theta(n^{\log_4 3}) \\
=& O(n^2)
\end{aligned}$$

（由于第一个递归调用的代价是$\Theta(n^2)$，所以$\Omega(n^2)$一定是整个递归式的下界，故这里得到的界其实是渐近紧确的。）

然后利用代换法可以得到对这个猜测的证明。

一个复杂一点的例子：$T(n) = T(n/3) + T(2n/3) + O(n)$

从根部到叶子的最长路径是$n \to (2/3)n \to (2/3)^2 n \to \cdots \to 1$，因此树的深度满足$(2/3)^i n = 1 \iff i = \log_{3/2} n$，由于在变量成线性的递归式中，每层代价是相同的(其实越往下越小，有的点在消失，这里说的是中间部分。)，因此作出预期：递归式的解至多是层数乘以每层的代价，亦即$O(cn \log_{3/2} n) = O(n \lg n)$。这里容忍了一些误差，如不断消失的结点，这样做出猜测后同样可以用代换法进行证明。

练习

4.2-1：

从根部到叶子的最长路径是$n \to (1/2)n \to (1/2)^2 n \to \cdots \to 1$，当$(1/2)^i n = 1$时，算得树的深度是$i = \lg n$，每层的总代价是$n$，因此猜测渐近上界$O(n \lg n)$。

$$\begin{aligned}
T(n) \leqslant& 3c \times n/2 \lg(n/2) + n \\
=& \frac{3}{2} cn \lg n - \frac{3}{2} cn + n \\
\leqslant& cn \lg n \\
\Longleftrightarrow& \frac{1}{2} cn \lg n + \frac{3}{2} cn - n \geqslant 0 \\
\Longrightarrow& c \lg n + 3c - 1 \geqslant 0 \\
\Longrightarrow& 3c \geqslant 1 \Longrightarrow c \geqslant \frac{1}{3}
\end{aligned}$$

故取$c = \frac{1}{3}$即可以满足猜测条件。

4.2-2：

按这棵子树：$n \to (1/3)n \to (1/3)^2 n \to \cdots \to 1$就可以得出$\Omega(n \lg n)$

4.2-3：

$\Theta(n \lg n)$

4.2-4：

从根部到叶子的最长路径是$n \to n-a \to n-a^2 \cdots \to 1$，$n - a^i = 1$，容忍误差计算得$i = \log_a n$，即该树的深度是$\log_a n$，每层总代价$cn$，因此得到$O(n \log_a n)$。在每个分支的二叉树里去掉$T(a)$，即只计算$n \to n-a \to n-a^2 \cdots \to 1$这个路径上的结点，得到$\Omega(n \log_a n)$（因为$T(n-a^i)$和$T(n)$的界一致），故得到渐近确界$\Theta(n \log_a n)$。

4.2-5：

两个子树的深度分别是$\log_{\frac{1}{\alpha}} n$和$\log_{\frac{1}{1-\alpha}} n$，所以上下界分别是$O(n \log_{\frac{1}{\alpha}} n)$和$\Omega(n \log_{\frac{1}{1-\alpha}} n)$（$\frac{1}{\alpha} > \frac{1}{1-\alpha}$）或者$O(n \log_{\frac{1}{1-\alpha}} n)$和$\Omega(n \log_{\frac{1}{\alpha}} n)$（$\frac{1}{\alpha} < \frac{1}{1-\alpha}$）

当$\frac{1}{\alpha} > \frac{1}{1-\alpha}$时，$\frac{1}{\alpha} > 2$，因此降低上界$O(n \log_{\frac{1}{\alpha}} n) = O(n \lg n)$，$\frac{1}{1-\alpha} < 2$，因此提

高下界$\Omega(\log_{\frac{1}{1-\alpha}} n) = \Omega(n \lg n)$，故得确界$\Theta(n \lg n)$

当$\frac{1}{\alpha} < \frac{1}{1-\alpha}$时可类似求证。

主定理：设$a \geqslant 1$和$b > 1$为常数，设$f(n)$为一函数，$T(n)$由递归式

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

对非负整数定义，其中n/b指$\lceil n/b \rceil$或$\lfloor n/b \rfloor$。那么$T(n)$可能有如下的渐近界：

1） 若对常数$\varepsilon > 0$，有$f(n) = O\left(n^{\log_b a - \varepsilon}\right)$，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） 若对常数$\varepsilon > 0$，有$f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right)$，且对常数$c > 1$与所有足够大的$n$，有

$af(n/b) \leqslant cf(n)$，则$T(n) = \Theta(f(n))$

练习

4.3-1：

a)$f(n) = n$,$n^{\log_b a} = n^2$,$f(n) = O(n^{\log_b a - \varepsilon})$,$\varepsilon = 1$,$\therefore T(n) = \Theta(n^2)$

b)$f(n) = n^2$,$n^{\log_b a} = n^2$,$f(n) = \Theta(n^{\log_b a})$,$\therefore T(n) = \Theta(n^2 \lg n)$

c)$f(n) = n^3$,$n^{\log_b a} = n^2$,$f(n) = \Omega(n^{\log_b a + \varepsilon})$,$af(n/b) \leqslant cf(n) \implies c \geqslant \frac{1}{2}$,取$c = \frac{1}{2}$即可。
$\therefore T(n) = \Theta(n^3)$

4.3-2：

A中$n^{\log_b a} = n^{\lg 7} \approx n^{2.81}$，$f(n) = n^2$，$f(n) = O(n^{2.81-0.81})$，$\varepsilon = 0.81 > 0$，所以 A 的渐近确界是 $\Theta(n^2)$

A'显然有个$\Omega(n^2)$，因此A'肯定不比A快，至多一样快。

4.3-3：

$n^{\log_b a} = 1$，$f(n) = \Theta(1) = \Theta(n^{\log_b a})$，满足主定理情况二，因此：

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

4.3-4：

$f(n) = n^2 \lg n$，$n^{\log_b a} = n^{\log_2 4} = n^2$，$n^2 \lg n$渐近大于$n^2$，$f(n)/n^{\log_b a} = \lg n$渐近小于$n^\varepsilon$（期望的是渐进紧确）,因此不可以使用主定理。利用构造递归树后再用代换法证明，给出渐近上界$O(n^2 \lg^2 n)$。

4.3-5：

假设$T(n) = T(n/2) + F(n)$，其中：

$$F(s) = \begin{cases} n & s = n \\ n & s \neq n \end{cases}$$