算法分析与设计: 第二次作业

PB20061372　朱云沁　Mar. 28, 2023

4.1-3 修改最大子数组问题的定义, 允许结果为空子数组, 其和为 0. 该如何修改现有算法, 使它们能允许空子数组为最终结果?（最大子数组问题的定义请参考推荐教材 P38-4.1）

解: 伪代码如下:

Algorithm 1: 4.1-3

```
Input: an array A[1..n]
Output: a subarray A[i..j] with the maximum sum, returned as a tuple (i, j, ∑A[i..j])
by FIND-MAX-SUBARRAY(A, 1, n). The subarray may be empty if i > j and ∑A[i..j] = 0.
function FIND-MAX-SUBARRAY(A, low, high):
    if low > high:
        return (low, high, 0)
    mid ← ⌊(low+high)/2⌋
    (leftLow, leftHigh, leftSum) ←
                        FIND-MAX-SUBARRAY(A, low, mid)
    (rightLow, rightHigh, rightSum) ←
                        FIND-MAX-SUBARRAY(A, mid + 1, high)
    (crossLow, crossHigh, crossSum) ←
                        FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high)
    if leftSum ≥ rightSum and leftSum ≥ crossSum:
        return (leftLow, leftHigh, leftSum)
    else if rightSum ≥ leftSum and rightSum ≥ crossSum:
        return (rightLow, rightHigh, rightSum)
    else:
        return (crossLow, crossHigh, crossSum)
function FIND-MAX-CROSSING-SUBARRAY(A, low, mid, high):
    maxLeft ← maxRight ← mid
    maxSum ← sum ← A[mid]
    for i = mid − 1 downto low do:
        sum = sum + A[i]
        if sum > maxSum:
            maxLeft = i
            maxSum = sum
    sum = maxSum
    for j = mid + 1 to high do:
        sum = sum + A[j]
        if sum > maxSum:
            maxRight = i
            maxSum = sum
return (maxLeft, maxRight, maxSum)
```

4.2-7 设计算法, 仅使用 3 次实数乘法即可完成复数 $a+bi$ 和 $c+di$ 相乘. 算法需要接收 a, b, c 和 d 为输入, 分别生成实部 $ac-bd$ 和虚部 $ad+bc$.

解: 伪代码如下:

Algorithm 2: 4.1-7

```
Input: 4 real numbers a, b, c and d.
Output: 2 real numbers ac − bd and ad + bc.
    let x = (a + b)c, y = b(c + d), z = a(c − d)
    return (x − y, x − z)
```

4.3-6 证明: $T(n)=2T\left(\left\lfloor\frac{n}{2}\right\rfloor+17\right)+n$ 的解为 $O(n\lg n)$.

解: 假设存在 $n,a,c>0$, 使得当 $k<n$ 时, $T(k)\le c(k-a)\lg k$. 故

$$T\left(\left\lfloor\frac{n}{2}\right\rfloor+17\right)\le c\left(\left\lfloor\frac{n}{2}\right\rfloor+17-a\right)\lg\left(\left\lfloor\frac{n}{2}\right\rfloor+17\right) \qquad n\ge 35$$

当 $k=n$ 时, 有

$$\begin{aligned}
T(n)&=2T\left(\left\lfloor\frac{n}{2}\right\rfloor+17\right)+n\\
&\le 2c\left(\left\lfloor\frac{n}{2}\right\rfloor+17-a\right)\lg\left(\frac{n}{2}+17\right)+n && a\le 34\\
&=2c\left(\left\lfloor\frac{n}{2}\right\rfloor+17-a\right)\left[\lg\left(\frac{2n+68}{3}\right)-\lg\frac{4}{3}\right]+n\\
&=2c\left(\left\lfloor\frac{n}{2}\right\rfloor+17-a\right)\lg\left(\frac{2n+68}{3}\right)-2c\lg\frac{4}{3}\left(\left\lfloor\frac{n}{2}\right\rfloor+17-a\right)+n\\
&\le c(n+34-2a)\lg n-c\lg\frac{4}{3}(n+32-2a)+n && n\ge 68\\
&=c(n-a)\lg n-\left(c\lg\frac{4}{3}-1\right)n+36c\lg\frac{4}{3} && a=34\\
&\le c(n-a)\lg n && c>\tfrac{17}{16-8\lg 3}
\end{aligned}$$

取 $n_0=68,a=34,c=6$. 归纳得 $n\ge 68$ 时, $T(n)\le c(n-a)\lg n<cn\lg n$. 故 $T(n)=O(n\lg n)$.

4.3-9 利用改变变量的方法来求解递归式 $T(n)=3T(\sqrt{n})+\lg n$. 你的解应该是渐进紧确的. 不必担心数值是否整数.

解: 令 $m=\lg n$, $S(m)=T(n)$ 则有 $S(m)=3S(\frac{m}{2})+m$. 由于 $a=3$, $b=2$, $f(m)=m$, 取 $\epsilon=\log_2 3-1$, 有 $f(m)=O(m^{\log_b a-\epsilon})$, 满足主方法第一种情况, 故 $S(m)=\Theta(m^{\log_2 3})$. 换元得 $T(n)=\Theta((\lg n)^{\log_2 3})$.

4.4-7 对递归式 $T(n)=4T\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+cn$ (c 为常数), 画出递归树, 并给出其解的一个渐进紧确界. 用代入法进行验证.

解: 递归树如下

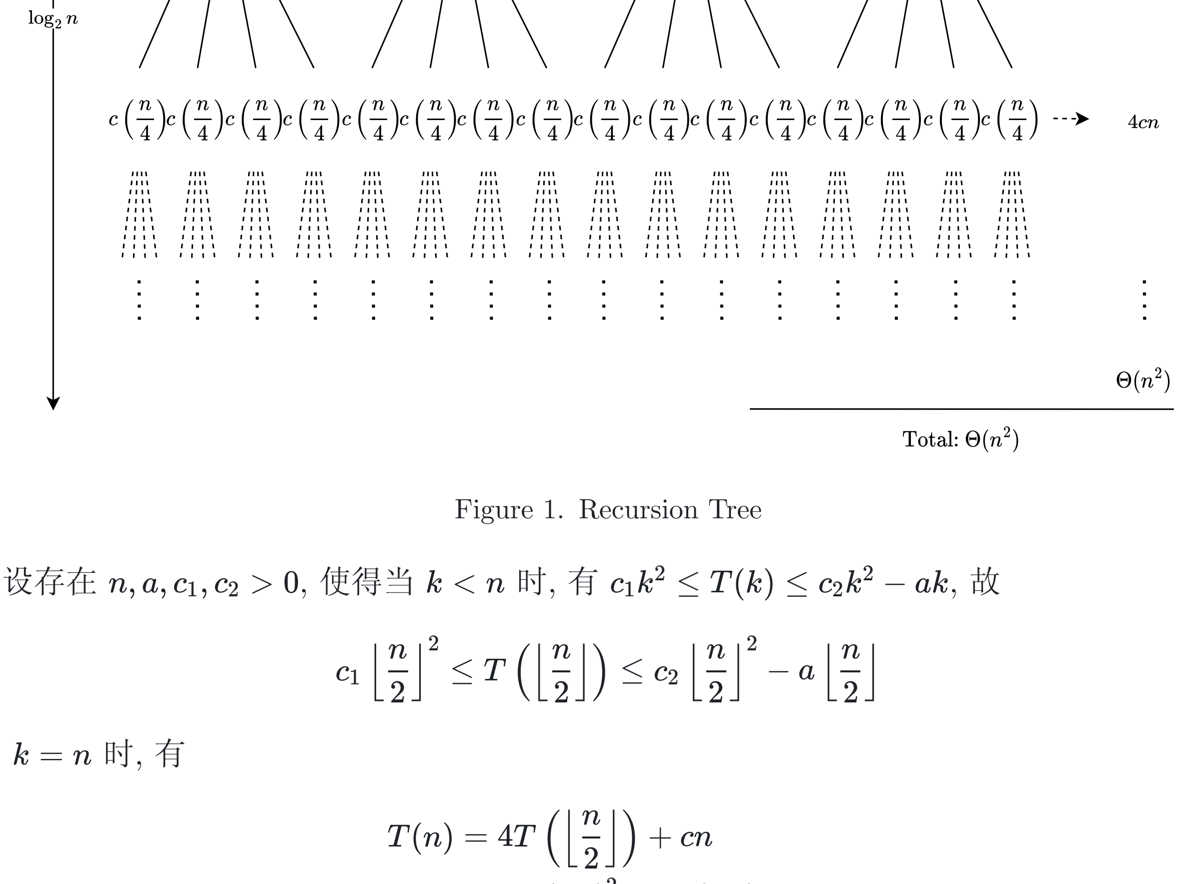

Figure 1. Recursion Tree

假设存在 $n,a,c_1,c_2>0$, 使得当 $k<n$ 时, 有 $c_1k^2\le T(k)\le c_2k^2-ak$, 故

$$c_1\left\lfloor\frac{n}{2}\right\rfloor^2\le T\left(\left\lfloor\frac{n}{2}\right\rfloor\right)\le c_2\left\lfloor\frac{n}{2}\right\rfloor^2-a\left\lfloor\frac{n}{2}\right\rfloor$$

当 $k=n$ 时, 有

$$\begin{aligned}
T(n)&=4T\left(\left\lfloor\frac{n}{2}\right\rfloor\right)+cn\\
&\le 4c_2\left\lfloor\frac{n}{2}\right\rfloor^2-4a\left\lfloor\frac{n}{2}\right\rfloor+cn\\
&\le c_2n^2+(c-2a)n+4a\\
&\le c_2n^2-an && (n-4)a\ge cn
\end{aligned}$$

$$\begin{aligned}
T(n)&\ge 4c_1\left\lfloor\frac{n}{2}\right\rfloor^2+cn\\
&\ge c_1(n-2)^2+cn\\
&=c_1n^2+(c-4c_1)n+4c_1\\
&\ge c_1n^2 && c_1\le\tfrac{c}{4}
\end{aligned}$$

取 $n_0=5,a=5c,c_1=\frac{c}{4},c_2=5c$, 归纳得 $n\ge 5$ 时, $\frac{c}{4}n^2\le T(n)\le 5c(n^2-n)\le 5cn^2$. 故 $T(n)=\Theta(n^2)$.

4.5-1 对下列递归式, 使用主方法求出渐近紧确界.

(1) $T(n)=2T\left(\frac{n}{4}\right)+1$
(2) $T(n)=2T\left(\frac{n}{4}\right)+\sqrt{n}$
(3) $T(n)=2T\left(\frac{n}{4}\right)+n$
(4) $T(n)=2T\left(\frac{n}{4}\right)+n^2$

解:

(1) 由于 $a=2$, $b=4$, $f(n)=1$, 取 $\epsilon=0.5$, 有 $f(n)=O(n^{\log_b a-\epsilon})$, 满足主方法第一种情况, 故 $T(n)=\Theta(n^{\log_4 2})=\Theta(\sqrt{n})$.

(2) 由于 $a=2$, $b=4$, $f(n)=n^{0.5}$, 有 $f(n)=\Theta(n^{\log_b a})$, 满足主方法第二种情况, 故 $T(n)=\Theta(n^{\log_4 2}\lg n)=\Theta(\sqrt{n}\lg n)$.

(3) 由于 $a=2$, $b=4$, $f(n)=n$, 取 $\epsilon=0.5$, 有 $f(n)=\Omega(n^{\log_b a+\epsilon})$, 且 $2f\left(\frac{n}{4}\right)=\frac{n}{2}\le cn$ 对 $0.5\le c<1$ 恒成立, 满足主方法第三种情况, 故 $T(n)=\Theta(f(n))=\Theta(n)$.

(4) 由于 $a=2$, $b=4$, $f(n)=n^2$, 取 $\epsilon=1.5$, 有 $f(n)=\Omega(n^{\log_b a+\epsilon})$, 且 $2f\left(\frac{n}{4}\right)=\frac{n^2}{8}\le cn^2$ 对 $0.125\le c<1$ 恒成立, 满足主方法第三种情况, 故 $T(n)=\Theta(f(n))=\Theta(n^2)$.

4.5-4 主方法能应用于递归式 $T(n)=4T\left(\frac{n}{2}\right)+n^2\lg n$ 吗? 请说明为什么可以或者为什么不可以. 给出这个递归式的一个渐近上界.

解: 由于 $a=4$, $b=2$, $f(n)=n^2\lg n=\Omega(n^{\log_b a})$, 考虑主方法第三种情况.

$$\frac{4f\left(\frac{n}{2}\right)}{f(n)}=\frac{n^2\lg n-n^2\lg 2}{n^2\lg n}\to 1 \qquad n\to\infty$$

不存在 $0<c<1$ 和 n_0 使得 $af\left(\frac{n}{b}\right)\le cf(n)$ 对所有 $n\ge n_0$ 恒成立, 无法应用主方法.
假设存在 $n,c>0$, 使得当 $k<n$ 时, 有 $T(k)\le ck^2\lg^2 k$, 故

$$T\left(\frac{n}{2}\right)\le c\frac{n^2}{4}\lg^2\frac{n}{2}$$

当 $k=n$ 时, 有

$$\begin{aligned}
T(n)&=4T\left(\frac{n}{2}\right)+n^2\lg n\\
&\le cn^2\lg^2\frac{n}{2}+n^2\lg n\\
&=cn^2(\lg n-1)^2+n^2\lg n\\
&=cn^2\lg^2 n+(1-2c)n^2\lg n+cn^2\\
&=n^2\lg^2 n-(\lg n-1)n^2 && c=1\\
&\le n^2\lg^2 n && n\ge 2
\end{aligned}$$

取 $n_0=2,c=1$, 归纳得 $n\ge 2$ 时, $T(n)=O(n^2\lg^2 n)$.