傅申 PB20000051
2022 秋

算法基础 作业 2

4.1–4. 假定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. 你应该如何修改现有算法, 使它们能允许空子数组为最终结果?

解: 在 FIND-MAXIMUM-SUBARRAY 过程中, 如果需要返回的子数组和为负, 则改为返回空子数组.

4.1–5. 使用如下思想为最大子数组问题设计一个非递归的, 线性时间的算法. 从数组的左边开始, 从左至右处理, 记录到目前为止以及处理过的最大子数组. 若已知 $A[1..j]$ 的最大子数组, 基于如下性质将解扩展为 $A[1..j+1]$ 的最大子数组: $A[1..j+1]$ 的最大子数组要么是 $A[1..j]$ 的最大子数组, 要么是某个子数组 $A[i..j+1](1 \leqslant i \leqslant j+1)$. 在已知 $A[1..j]$ 的最大子数组的情况下, 可以在线性时间内找出形如 $A[i..j+1]$ 的最大子数组.

解: 如下

Algorithm 1: LINEAR-TIME-MAX-SUBARRAY(A)

```
max-sum = -∞
max-tail-sum = 0
tail-left = 1
for j = 1 to A.length
    max-tail-sum = max-tail-sum + A[j]
    if max-tail-sum > max-sum
        max-sum = max-tail-sum
        max-left = tail-left
        max-right = j
    if max-tail-sum < 0
        max-tail-sum = 0
        tail-left = j + 1
return max-left, max-right, max-sum
```

4.3–3. 我们看到 $T(n) = 2T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + n$ 的解为 $O(n \lg n)$. 证明 $\Omega(n \lg n)$ 也是这个递归式的解. 从而得出结论: 解为 $\Theta(n \lg n)$.

解: 恰当选择常数 $\frac{1}{3} \geqslant c > 0$, 有 $T(n) \geqslant cn \lg n$. 假定此上界对所有正数 $m < n$ 都成立, 特别对于 $m = \lfloor \frac{n}{2} \rfloor$, 有 $T\left(\lfloor \frac{n}{2} \rfloor\right) \geqslant c \lfloor \frac{n}{2} \rfloor \lg\left(\lfloor \frac{n}{2} \rfloor\right)$, 将此代入递归式, 有

$$\begin{aligned}
T(n) &\geqslant 2\left(c\left\lfloor \frac{n}{2} \right\rfloor \lg\left(\left\lfloor \frac{n}{2} \right\rfloor\right)\right) + n \geqslant c(n-1)\lg\frac{n-1}{2} + n \\
&= c(n-1)\left(\lg n - 1 - \lg\frac{n}{n-1}\right) + n \\
&= cn\left(\lg n - 1 - \lg\frac{n}{n-1} + \frac{1}{c}\right) - c\left(\lg(n-1) - 1\right) \\
&\geqslant cn\left(\lg n - 2 + \frac{1}{c} - \frac{\lg(n-1)-1}{n}\right) \\
&\geqslant cn\left(\lg n - 3 + \frac{1}{c}\right) \\
&\geqslant cn \lg n
\end{aligned} \tag{3.1}$$

因此 $\Omega(n \lg n)$ 也是这个递归式的解. 得到解为 $\Theta(n \lg n)$.

4.3–5. 证明: 归并排序的 "严格" 递归式 (4.3) 的解为 $\Theta(n \lg n)$.

$$T(n) = \begin{cases} \Theta(1) & \text{若 } n = 1 \\ T\left(\left\lceil \frac{n}{2} \right\rceil\right) + T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \Theta(n) & \text{若 } n > 1 \end{cases} \tag{5.1}$$

解: 不妨令递归式中的 $\Theta(n)$ 满足 $d_1 n \leqslant \Theta(n) \leqslant d_2 n$.

先证明 $O(n \lg n)$ 是这个递归式的解. 恰当选择常数 $c \geqslant \frac{3}{2} d_2$, 有 $T(n) \leqslant cn \lg n$. 假定此上界对所有正数 $m < n$ 都成立, 将此代入递归式, 若 n 为偶数, 有

$$\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + \Theta(n) \\
&\leqslant cn \lg\frac{n}{2} + d_2 n \\
&= cn \lg n + (d_2 - c)\, n \\
&\leqslant cn \lg n
\end{aligned} \tag{5.2}$$

若 n 为奇数 (显然 $n \geqslant 3$), 有

$$\begin{aligned}
T(n) &= T\left(\frac{n-1}{2}\right) + T\left(\frac{n+1}{2}\right) + \Theta(n) \\
&\leqslant c\frac{n-1}{2}\lg\frac{n-1}{2} + c\frac{n+1}{2}\lg\frac{n+1}{2} + d_2 n \\
&= \frac{cn}{2}\lg\frac{n^2-1}{4} + \frac{c}{2}\lg\frac{n+1}{n-1} + d_2 n \\
&\leqslant cn\lg n + c\frac{2}{n-1} - cn + d_2 n \\
&= cn\lg n + d_2 n - c\frac{(n-2)(n+1)}{(n-1)} \\
&\leqslant cn\lg n - d_2 n\left(\frac{3}{2}\cdot\frac{(n-2)(n+1)}{(n-1)n} - 1\right) \\
&\leqslant cn\lg n
\end{aligned} \tag{5.3}$$

其中当 $n \geqslant 3$ 时, $\dfrac{(n-2)(n+1)}{(n-1)n} \geqslant \dfrac{3}{2}$.

再证明 $\Omega(n\lg n)$ 是这个递归式的解. 恰当选择常数 $0 < c \leqslant \frac{16}{17}d_1$, 有 $T(n) \geqslant cn\lg n$. 假定此下界对所有正数 $m < n$ 都成立, 将此代入递归式, 若 n 为偶数, 有

$$\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + \Theta(n) \\
&\geqslant cn\lg\frac{n}{2} + d_1 n \\
&= cn\lg n + (d_1 - c)\,n \\
&\geqslant cn\lg n
\end{aligned} \tag{5.4}$$

若 n 为奇数 (显然 $n \geqslant 3$), 有

$$\begin{aligned}
T(n) &= T\left(\frac{n-1}{2}\right) + T\left(\frac{n+1}{2}\right) + \Theta(n) \\
&\geqslant c\frac{n-1}{2}\lg\frac{n-1}{2} + c\frac{n+1}{2}\lg\frac{n+1}{2} + d_1 n \\
&= \frac{cn}{2}\lg\frac{n^2-1}{4} + \frac{c}{2}\lg\frac{n+1}{n-1} + d_1 n \\
&\geqslant \frac{cn}{2}\left(\lg n^2 - \lg\frac{n^2}{n^2-1}\right) - cn + d_1 n \\
&\geqslant cn\lg n - cn\left(1 + \frac{1}{2\left(n^2-1\right)}\right) + d_1 n \\
&\geqslant cn\lg n
\end{aligned} \tag{5.5}$$

其中当 $n \geqslant 3$ 时, $\dfrac{1}{2\left(n^2-1\right)} \leqslant \dfrac{1}{16}$, 所以有最后一步的大于等于.

综上所述, $O(n\lg n)$ 和 $\Omega(n\lg n)$ 都是递归式的解, 因此递归式的解为 $\Theta(n\lg n)$

4.3–7. 使用 4.5 节中的主方法, 可以证明 $T(n) = 4T\left(\frac{n}{3}\right) + n$ 的的解为 $T(n) = \Theta\left(n^{\log_3 4}\right)$. 说明基于假设 $T(n) \leqslant cn^{\log_3 4}$ 的代入法不能证明这一结论. 然后说明如何通过减去一个低阶项完成代入法证明.

解: 基于假设 $T(n) \leqslant cn^{\log_3 4}$, 假定此上界对所有 $m < n$ 都成立, 代入递归式, 得到

$$\begin{aligned} T(n) &\leqslant 4c\Big(\frac{n}{3}\Big)^{\log_3 4} + n \\ &= cn^{\log_3 4} \cdot \frac{4}{3^{\log_3 4}} + n \\ &= cn^{\log_3 4} + n \end{aligned} \tag{7.1}$$

其中除了第一部是基于假设的直接放缩, 其他两步都是严格相等, 但是最后得到的结果为 $cn^{\log_3 4} + n > cn^{\log_3 4}$, 因此基于该假设无法证明该结论.

不妨假设 $T(n) \leqslant cn^{\log_3 4} - 3n$, 只要 c 取的足够大就能满足基本情况. 假定此上界对所有正数 $m < n$ 都成立, 代入递归式, 得到

$$\begin{aligned} T(n) &\leqslant 4\left(c\Big(\frac{n}{3}\Big)^{\log_3 4} - n\right) + n \\ &= cn^{\log_3 4} - 4n + n \\ &= cn^{\log_3 4} - 3n \end{aligned} \tag{7.2}$$

因为 $cn^{\log_3 4} - 3n = O\left(n^{\log_3 4}\right)$, 所以 $O\left(n^{\log_3 4}\right)$ 为递归式的解.

再假设 $T(n) \geqslant cn^{\log_3 4}$. 假定此下界对所有正数 $m < n$ 都成立, 代入递归式, 得到

$$\begin{aligned} T(n) &\geqslant 4c\Big(\frac{n}{3}\Big)^{\log_3 4} + n \\ &= cn^{\log_3 4} \cdot \frac{4}{3^{\log_3 4}} + n \\ &= cn^{\log_3 4} + n \\ &\geqslant cn^{\log_3 4} \end{aligned} \tag{7.3}$$

因此 $\Omega\left(n^{\log_3 4}\right)$ 是递归式的解.

综上所述, $\Theta\left(n^{\log_3 4}\right)$ 是递归式的解.

4.4–3. 对递归式 $T(n) = 4T\left(\frac{n}{2} + 2\right) + n$, 利用递归树确定一个好的渐进上界, 用代入法进行验证.

解: 递归树的深度约为 $\lg n$, 深度为 i 的层共有 4^i 个结点, 每个结点代价约为 $\frac{n}{2^i}$. 求和得到总代价约为 $2n^2$, 因此假设渐进上界为 $O\left(n^2\right)$. 使用代入法, 恰当选择常数 $c \geqslant 1$, 有 $T(n) \leqslant c\left(n^2 - 9n\right)$. 假定此上界对所有正数 $m < n$ 都成立, 代入递归式, 得到

$$\begin{aligned} T(n) &\leqslant 4c\left(\left(\frac{n}{2} + 2\right)^2 - 9\left(\frac{n}{2} + 2\right)\right) + n \\ &= c\left(n^2 - 10n - 56\right) + n \\ &= c\left(n^2 - 9n\right) - (c-1)n - 56c \\ &\leqslant c\left(n^2 - 9n\right) \end{aligned} \tag{3.1}$$

因为 $n^2-9n=O\left(n^2\right)$, 所以 $O\left(n^2\right)$ 是渐进上界.

4.4–5. 对递归式 $T(n)=T(n-1)+T\left(\frac{n}{2}\right)+n$, 利用递归树确定一个好的渐进上界, 用代入法进行验证.

解: 递归树有一部分类似于 $T(n)=2T(n-1)+n$ 的递归树, 但是有另一部分类似于 $T(n)=2\left(\frac{n}{2}\right)+n$ 的递归树. 前者的总代价是指数级别的 $O\left(2^n\right)$, 所以不难得到解的一个渐进上界为 $O\left(2^n\right)$.

使用代入法, 恰当选择常数 $c\geqslant 1$, 有 $T(n)\leqslant c2^n-3$. 假定此上界对所有正数 $m<n$ 都成立, 代入递归式, 得到

$$
\begin{aligned}
T(n) &\leqslant c2^{n-1}-3+c2^{\frac{n}{2}}-3+n \\
&= c\left(2^{n-1}+2^{\frac{n}{2}}\right)-3+(n-3) \\
&\begin{cases} \leqslant c\left(2^{n-1}+2^{\frac{n}{2}}\right)-3 & 0<n\leqslant 3 \\ \leqslant c2^{n-1}+\left(c2^{\frac{n}{2}}+n\right)-3\leqslant c\left(2^{n-1}+2^{\frac{n}{2}+1}\right)-3 & n\geqslant 4 \end{cases} \\
&\leqslant c2^n-3
\end{aligned}
\tag{5.1}
$$

因为 $c2^n-3=O\left(2^n\right)$, 所以 $O\left(2^n\right)$ 是 $T(n)$ 的递归上界.

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确界. 用代入法进行验证.

解: 递归树如下图 7.1所示. 得到渐进确界为 $\Theta\left(n^2\right)$, 下面用代入法证明.

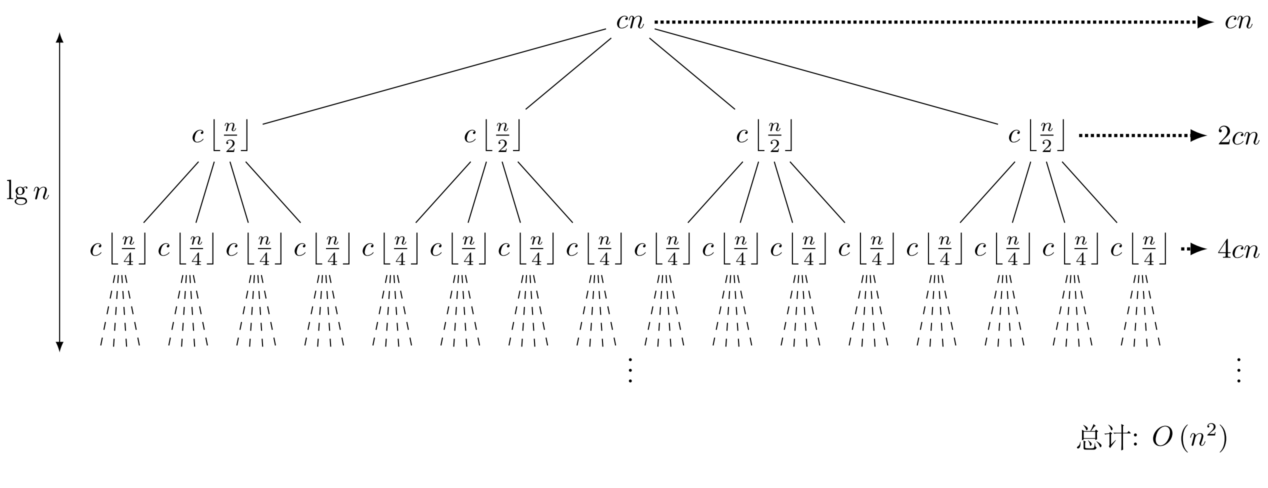

图 7.1: 题中递归式对应的递归树

先证明 $T(n)=O\left(n^2\right)$. 恰当选择常数 $d>c$, 有 $T(n)\leqslant dn^2-cn$. 假定此上界对所有正数 $m<n$ 都成立, 将其代入递归式得到

$$\begin{aligned}T(n)&\leqslant 4d\left\lfloor\frac{n}{2}\right\rfloor^2-4c\left\lfloor\frac{n}{2}\right\rfloor+cn\\&\leqslant 4d\left(\frac{n}{2}\right)^2-cn\\&=dn^2-cn\end{aligned}\tag{7.1}$$

再证明 $T(n)=\Omega\left(n^2\right)$. 恰当选择常数 $\frac{c}{5}\geqslant d>0$, 有 $T(n)\geqslant d\left(n^2+4n\right)$. 假定此下界对所有正数 $m<n$ 都成立, 将其代入递归式得到

$$\begin{aligned}T(n)&\geqslant 4d\left(\left\lfloor\frac{n}{2}\right\rfloor^2+4\left\lfloor\frac{n}{2}\right\rfloor\right)+cn\\&\geqslant 4d\left(\frac{n-1}{2}\right)^2+8d(n-1)+cn\\&=d\left(n^2+4n\right)+2dn+cn-7d\\&\geqslant d\left(n^2+4n\right)+cn-5d\\&\geqslant d\left(n^2+4n\right)\end{aligned}\tag{7.2}$$

4.5–2. Caesar 教授项设计一个渐进快于 Strassen 算法的矩阵相乘算法. 他的算法使用分治方法, 将每个矩阵分解为 $\frac{n}{4}\times\frac{n}{4}$ 的子矩阵, 分解和合并步骤共花费 $\Theta\left(n^2\right)$ 时间. 他需要确定, 他的算法需要创建多少个子问题, 才能击败 Strassen 算法. 如果他的算法创建 a 个子问题, 则描述运行时间 $T(n)$ 的递归式为 $T(n)=aT\left(\frac{n}{4}\right)+\Theta(n^2)$. Caesar 教授的算法如果要渐进快于 Strassen 算法, a 的最大整数值应是多少?

解: 当 a 取最大值时, 递归式显然适用于主方法的情况一, 递归式的解为

$$T(n)=\Theta\left(n^{\frac{1}{2}\lg n}\right)<\Theta\left(n^{\lg 7}\right)\tag{2.1}$$

解得 $a<49$, 最大值整数值为 48.

当 a 为 48 时, 对于常数 $\varepsilon=\frac{1}{2}\lg 48-2>0$, 有 $f(n)=\Theta\left(n^2\right)=\Theta\left(n^{\log_4 48-\varepsilon}\right)$, 满足主方法情况一的条件.

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进上界.

解: 可以应用 (如果是顾老师 PPT 上的主方法). 递归式满足 $f(n)=\Theta\left(n^{\log_2 4}\lg^1 n\right)$, 由主方法情况二可得 $T(n)=\Theta\left(n^2\lg^2 n\right)$.