# 数据结构与算法 II 作业 (9.15)

中国人民大学　信息学院　崔冠宇　2018202147

**P47, T4.2-1**　使用 Strassen 算法计算如下矩阵乘法:

$$\begin{bmatrix} 1 & 3 \\ 7 & 5 \end{bmatrix} \begin{bmatrix} 6 & 8 \\ 4 & 2 \end{bmatrix}$$

给出计算过程。

**解:** 在此例子中, $A_{11} = [1], A_{12} = [3], A_{21} = [7], A_{22} = [5]; B_{11} = [6], B_{12} = [8], B_{21} = [4], B_{22} = [2]$.

$S_1 = B_{12} - B_{22} = [6], S_2 = A_{11} + A_{12} = [4], S_3 = A_{21} + A_{22} = [12], S_4 = B_{21} - B_{11} = [-2], S_5 = A_{11} + A_{22} = [6]$, $S_6 = B_{11} + B_{22} = [8], S_7 = A_{12} - A_{22} = [-2], S_8 = B_{21} + B_{22} = [6], S_9 = A_{11} - A_{21} = [-6], S_{10} = B_{11} + B_{12} = [14]$.

$P_1 = A_{11}S_1 = [6], P_2 = S_2B_{22} = [8], P_3 = S_3B_{11} = [72], P_4 = A_{22}S_4 = [-10], P_5 = S_5S_6 = [48], P_6 = S_7S_8 = [-12]$, $P_7 = S_9S_{10} = [-84]$.

$C_{11} = P_5 + P_4 - P_2 + P_6 = [18], C_{12} = P_1 + P_2 = [14], C_{21} = P_3 + P_4 = [62], C_{22} = P_5 + P_1 - P_3 - P_7 = [66]$.

所以 $\begin{bmatrix} 1 & 3 \\ 7 & 5 \end{bmatrix} \begin{bmatrix} 6 & 8 \\ 4 & 2 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} 18 & 14 \\ 62 & 66 \end{bmatrix}$.

**P50, T4.3-1**　证明: $T(n) = T(n-1) + n$ 的解是 $O(n^2)$ 的.

**证明:** ① 假设对于任意 $k < n$ 有 $T(k) \leq ck^2$, 则

$$\begin{aligned} T(n) &= T(n-1) + n \\ &\leq c(n-1)^2 + n \\ &= cn^2 - 2cn + 1 + n \\ &= cn^2 - ((2c-1)n + 1) \end{aligned}$$

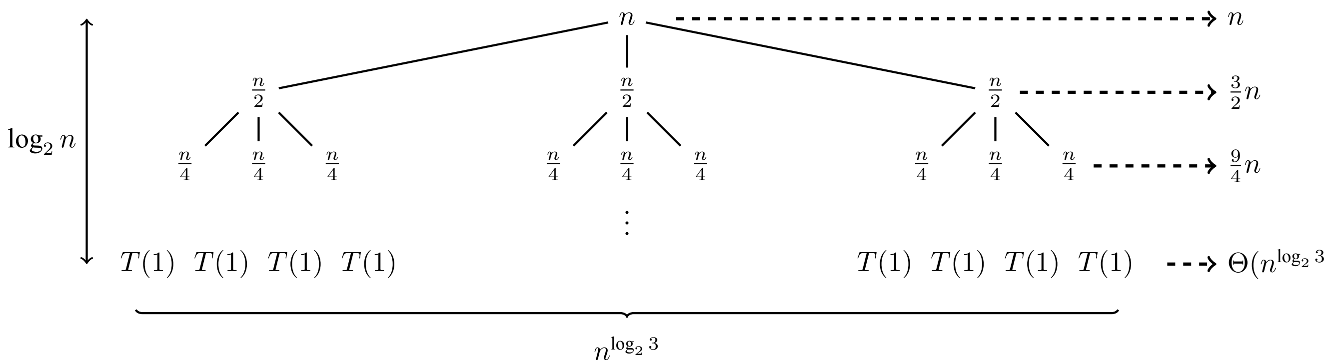
当 $c \geq \frac{1}{2}$ 时, 上式第二项为正数, 故 $T(n) \leq cn^2$, 归纳成立.

② 验证边界条件. 因为 $T(1) = \Theta(1)$, 取 $c \geq T(1)$ 时, $T(1) \leq c$, 边界条件成立.

综上, 当 $c \geq \frac{1}{2}$ 且 $c \geq T(1)$ 时, $T(n) \leq cn^2$ 恒成立, 所以 $T(n) = O(n^2)$.

**P53, T4.4-1**　对递归式 $T(n) = 3T(\lfloor n/2 \rfloor) + n$, 利用递归树确定一个好的渐进上界, 用代入法进行验证.

**解:** 为简单起见, 假设 $n$ 是 2 的幂. 画出递归树.




计算整棵树的代价:

$$
\begin{aligned}
T(n) &= n + \frac{3}{2}n + (\frac{3}{2})^2 n + \cdots + (\frac{3}{2})^{\log_2 n - 1} n + \Theta(n^{\log_2 3}) \\
&= n \sum_{i=0}^{\log_2 n - 1} (\frac{3}{2})^i + \Theta(n^{\log_2 3}) \\
&= (3 \cdot (\frac{3}{2})^{\log_2 n - 1} - 2)n + \Theta(n^{\log_2 3}) \\
&= (2 \cdot (\frac{3}{2})^{\log_2 n} - 2)n + \Theta(n^{\log_2 3}) \\
&= 2n^{\log_2 3} - 2n + \Theta(n^{\log_2 3}) \\
&= \Theta(n^{\log_2 3})
\end{aligned}
$$

下面用代入法验证 $T(n) = O(n^{\log_2 3})$.

① 假设对任意 $k < n$ 有 $T(k) \le c_1 k^{\log_2 3} - c_2 k$, 则

$$
\begin{aligned}
T(n) &= 3T(\lfloor n/2 \rfloor) + n \\
&\le 3(c_1(\lfloor n/2 \rfloor)^{\log_2 3} - c_2 \lfloor n/2 \rfloor) + n \\
&\le 3c_1(n/2)^{\log_2 3} - 3c_2(n/2) + 3c_2 + n \\
&= (c_1 n^{\log_2 3} - c_2 n) - (\frac{c_2 - 2}{2} n - 3c_2)
\end{aligned}
$$

可见当 $c_2 > 2, n$ 充分大时, 第二项为正, $T(n) \le (c_1 n^{\log_2 3} - c_2 n)$, 归纳成立.

② 验证边界条件: 因为 $T(1) = \Theta(1)$, 当取 $c_1 \ge c_2 + T(1)$ 时, $T(1) \le c_1 - c_2$, 也成立.

综上, 当 $c_1 \ge c_2 + T(1), c_2 > 2$ 时, 对充分大的 $n$ 有 $T(n) \le c_1 n^{\log_2 3} - c_2 n \le c_1 n^{\log_2 3}$, 所以 $T(n) = O(n^{\log_2 3})$.

**P55, T4.5-1** 对下列递归式, 使用主方法求出渐进紧确界。

**a.** $T(n) = 2T(n/4) + 1$

**b.** $T(n) = 2T(n/4) + \sqrt{n}$

**c.** $T(n) = 2T(n/4) + n$

**d.** $T(n) = 2T(n/4) + n^2$

**解:** 以下诸题均有 $a = 2, b = 4$ 以及 $\log_b a = 1/2$.

**a.** 取 $\varepsilon = 1/2$, 则 $f(n) = 1 = O(n^{\log_b a - \varepsilon}) = O(1)$. 根据主定理, $T(n) = \Theta(\sqrt{n})$.

**b.** $f(n) = \sqrt{n} = \Theta(n^{\log_b a})$. 根据主定理, $T(n) = \Theta(\sqrt{n}\log n)$.

**c.** 取 $\varepsilon = 1/2$, 则 $f(n) = n = \Omega(n^{\log_b a+\varepsilon})$, 且 $\exists c = 1/2 < 1$, 对充分大的 $n$ 有 $af(n/b) = n/2 \leq cf(n)$. 根据主定理, $T(n) = \Theta(n)$.

**d.** 取 $\varepsilon = 1/2$, 则 $f(n) = n^2 = \Omega(n^{\log_b a+\varepsilon})$, 且 $\exists c = 1/8 < 1$, 对充分大的 $n$ 有 $af(n/b) = n^2/8 \leq cf(n)$. 根据主定理, $T(n) = \Theta(n^2)$.