高级算法设计与分析作业2
数学基础

傅彦璋 23S003008

2024.5.3

1

反证，假设不为空集。任取$f(n) \in o(g(n)) \cap \omega(g(n))$.

对任意常数c，都有：

因为$f(n) \in o(g(n))$，所以$\exists n_0$，使得$n > n_0$时，$f(n) < c \cdot g(n)$成立。因为$f(n) \in \omega(g(n))$，所以$\exists n_1$，使得$n > n_1$时，$f(n) > c \cdot g(n)$成立。

取$n_2 = max\{n_0, n_1\}$，当$n > n_2$时，$c \cdot g(n) < f(n) < c \cdot g(n)$，这是不可能的。所以$o(g(n)) \cap \omega(g(n))$是空集。

2

记$h(n) = \max\{f(n), g(n)\}$.根据$h(n)$的定义，我们知道，$\exists n_0$，当$n > n_0$时，有$0 \le g(n) \le h(n)$, $0 \le f(n) \le h(n)$，又有$h(n) \le (f(n) + g(n))$.

总之$\frac{1}{2}(f(n) + g(n)) \le h(n) \le f(n)+g(n)$.所以$h(n)$与$f(n)+g(n)$是同阶的，即$\max\{f(n), g(n)\} = \Theta(f(n) + g(n))$.

3

先证明$\log(n!) = O(n \log n)$：

$\log(n!) = \log 1 + \log 2 + ... + \log n \le \sum_{i=1}^{n} \log n = n \log n = O(nlogn)$.

再证明$\log(n!) = \Omega(n \log n)$：

$$\log(n!) = \sum_{i=1}^{n} \log i \ge \sum_{i=n/2}^{n} \log i \ge \sum_{i=n/2}^{n} \log \frac{n}{2} = \frac{n}{2} \log \frac{n}{2} = \frac{n}{2} \log n - \frac{n}{2} \log 2$$

当$n > 4$时，$\frac{n}{2} \log n - \frac{n}{2} \log 2 \ge \frac{n \log n}{4}$，所以$\log(n!) = \Omega(n \log n)$。

总之，$\log(n!) = \Theta(n \log n)$.

4

$$
\begin{aligned}
T(n) &= T\left(\frac{3}{10}n\right) + 5n \\
&= T\left(\left(\frac{3}{10}\right)^2 n\right) + 5n\left(1 + \frac{3}{10}\right) \\
&= T\left(\frac{3^k}{10^k}n\right) + 5n\left(1 + \frac{3}{10} + ... + \left(\frac{3}{10}\right)^{k-1}\right)
\end{aligned}
$$

令$\frac{3^k}{10^k}n = 1$，则$n = \left(\frac{10}{3}\right)^k$.

$$
\begin{aligned}
T(n) &= T(1) + 5 \cdot \left(\frac{10}{3}\right)^k \cdot \left(\frac{1 - \left(\frac{3}{10}\right)^k}{1 - \frac{3}{10}}\right) \\
&= T(1) + \frac{50}{7}\left(\left(\frac{10}{3}\right)^k - 1\right) \\
&= T(1) + \frac{50(n-1)}{7} \\
&= O(n)
\end{aligned}
$$

5

$$
T(n) = T(\lceil n/2 \rceil) + 1 = T(\lceil \frac{n}{2^k} \rceil) + k
$$

令$\frac{n}{2^k} = 1$，则$k = \log_2 n$，则

$$
T(n) = T(1) + \log_2 n = O(\log n)
$$

6

$T(n) = 3T(n^{1/2}) + \log n$.令$n = 2^m$，则$T(2^m) = 3T(2^{m/2}) + m$. 令$S(m) = T(2^m)$，则$S(m) = 3S(\frac{m}{2}) + m$.

由Master定理，$S(m) = \Theta(m^{\log_2 3})$.

$T(n) = S(\log_2 n) = \Theta\left((\log_2 n)^{\log_2 3}\right)$.

7

$T(n)=3T(n-1)+2^n$，令$m=2^n$，则$T(\log_2 m)=3T(\log_2 \frac{m}{2})+m.$
令$S(m)=T(\log_2 m)$，则$S(m)=3S(m/2)+m.$
由Master定理，$S(m)=\Theta\left(m^{\log_2 3}\right)$.故$T(n)=S(2^n)=\Theta\left(2^{n\log_2 3}\right)$.

8

$T(n)\le 2T\left(\lfloor 3n/4\rfloor\right)+n$，由Master定理，$T(n)\in O\left(n^{\log_{4/3}2}\right)$.
$T(n)\ge 2T\left(\lfloor n/2\rfloor\right)+n$，由Master定理，$T(n)\in \Omega(n\log n)$.

9

$T(n)=2T(n/4)+n^{1/2}$.由Master定理，$T(n)=\Theta(\sqrt{n}\log n)$.

10

$T(n)=T(\lfloor n/2\rfloor)+n^3$.由Master定理，$3>\log_2 1$，$T(n)=\Theta(n^3)$.

11

$T(n)=5T(n/3)+n$.由Master定理，$\log_3 5>1$，$T(n)=\Theta(n^{log_3 5})$.

12

$T(n)=4T(n/2)+n^2\log n$.根据Master，记$b=2,a=4,f(n)=n^2\log n$，则$f(n)=\Theta(n^{log_b a}\log n)$，所以$T(n)=\Theta(n^{log_b a}\log^2 n)=\Theta(n^2\log^2 n)$.

13

$T(n)=\sqrt{n}T(\sqrt{n})+n$，则$T(n)/n=T(\sqrt{n})/\sqrt{n}+1$.
令$S(n)=\frac{T(n)}{n}$，则$S(n)=S(\sqrt{n})+1$.
令$n=2^m$，则$S(2^m)=S(2^{m/2})+1$.令$R(m)=S(2^m)$，则$R(m)=R(m/2)+1$.
由Master定理，$R(m)=\Theta(\log m)$.则$S(n)=\Theta(\log\log n)$.则$T(n)=\Theta(n\log\log n)$.