



# Chapter 1

## 课后练习

---

## 1

- 假设一个递归算法的程序步数可抽象为：  
 $T(n)=2T(n/2) + n$ ，假设  $T(1)=1$ ， $n = 2^k$ 。请分析该算法的程序步数。
- 注：更一般的情况是  $T(n)=T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + O(n)$ ， $T(1)=1$ ，不影响最终结果，简化为：  
 $T(n)=2T(n/2) + n$ ， $T(1)=1$

# 解答

## ○ 解法1：

- $T(n) = 2T(n/2) + n = 2(2T(n/4) + n/2) + n = 8T(n/8) + n + n + n$   
 $= 2^k T(n/2^k) + k * n$
- 由于  $k = \log n$ ,  $n = 2^k$
- 得到结果:  $T(n) = n + n \log n = O(n \log n)$

## ○ 解法2：

- $T(n)/n = T(n/2)/(n/2) + 1$ ,
- $T(n/2)/(n/2) = T(n/4)/(n/4) + 1, \dots$ ,
- $T(2)/2 = T(1)/1 + 1$
- 把上述各式左边相加, 右边相加, 消去同类项, 得到:
- $T(n)/n = T(1)/1 + \log n$ ,
- 最后得到与解法1相同的结果

## 2

- 判断 $f(n)$ 和 $g(n)$ , 当 $n \rightarrow \infty$ 时, 哪个函数增长更快?
  - $f(n) = (\ln(n!) + 5)^3$ ,  $g(n) = 13n^{2.5}$
  - $f(n) = n^{2.1} + \sqrt{n^4 + 1}$ ,  $g(n) = (\ln(n!))^2 + n$
  - $f(n) = 2^{(n*n*n)} + (2^n)^2$ ,  $g(n) = n^{(n*n)} + n^5$

# 解答

- $f(n) = (\ln(n!) + 5)^3$ ,  $g(n) = 13n^{2.5}$ 
  - 答：f(n)快。因为  $(\ln(n!) + 5)^3 > (n \ln 3)^3 > n^3 > 13n^{2.5}$ 。（这里的大于小于号表示增长速度的快慢）。
- $f(n) = n^{2.1} + \sqrt{n^4 + 1}$ ,  $g(n) = (\ln(n!))^2 + n$ 
  - 答：f(n)快， $(\ln(n!))^2 + n < (n \ln(n))^2$ ，所以就是比较  $(\ln(n))^2$  与  $n^{0.1}$ ，假设  $n = e^{10x}$ ，也就是比较  $100x^2$  与  $e^x$ ，显然  $e^x$  快，所以结论正确。
- $f(n) = 2^{(n*n*n)} + (2^n)^2$ ,  $g(n) = n^{(n*n)} + n^5$ 
  - 答：f(n)增长速度快