

CH2、Recurrence

遞迴的介紹

考題重點(目錄)

(如何選擇適合的方法)

- 一、Substitution Method
- 二、Recursion Tree
- 三、Master Theorem
- 四、數學解(迭代、變數變換)

Substitution Method

一、Concept

1. 先猜一個 Bound(經驗法則)
2. 假設子問題符合此 Bound，證母問題亦符合此 Bound

二、適用時機

分多、單邊、題目要求

三、例

$T(n) = 2T(\lfloor n/2 \rfloor) + n$ ，問 $T(n) = O(?)$

用 Substitution Method

猜 $T(n) = O(n \log n)$

(想法：已知 $g(n) = 2g(n/2) + n = O(n \log n)$ ，覺得有 floor 和無 floor 不會對 order 產生太大影響)

欲證： $T(n) = O(n \log n)$ ，即證：存在 $c, n_0 > 0$ ，使得當 $n \geq n_0$ ， $T(n) \leq c n \log n$

假設 $T(\lfloor n/2 \rfloor) \leq c \lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor$ ，考慮： $T(n) = 2T(\lfloor n/2 \rfloor) + n \leq 2c \lfloor n/2 \rfloor \lg \lfloor n/2 \rfloor$ (假設)

$\leq 2c n/2 \log n/2 + n = cn \log n - (1-c)n$ (取 $c=2$) $\leq c n \log n$

$\Rightarrow T(n) = O(n \log n)$

例(97 成大)：

$T(n) = 2T(\sqrt{n}) + \log n$ ，用 O 表示

用 Substitution Method

Recursion Tree

一、用到的 term：

$$T(n) = T(n/3) + T(2n/3) + n$$

母問題 子問題 成本

子問題 n 系數相加

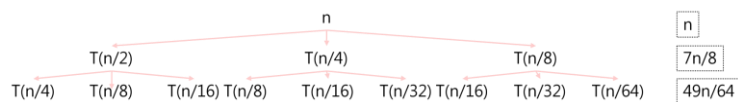
二、適用時機

子問題個數 ≥ 2 個，子問題： $T(n/a)$

三、 $r < 1$ 型

例： $T(n) = T(n/2) + T(n/4) + T(n/8) + n$ ，求 Θ

1. 建立 Recursion Tree



$$T(n) = n + 7n/8 + 49n/64 + \dots + c$$

2. 先求 Upper Bound(O)： $T(n) = n + 7n/8 + 49n/64 + \dots + c \leq n + 7n/8 + \dots = n/(1-7/8) = 8n$

$$T(n) = O(n) \quad // \text{小於無窮等比}$$

3. 再求 Lower Bound(Ω)： $T(n) = n + 7n/8 + 49n/64 + \dots + c \geq n \Rightarrow T(n) = \Omega(n)$

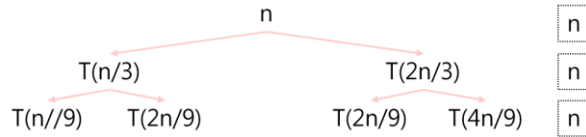
$\Rightarrow T(n) = \Theta(n)$

四、 $r=1$ 型

例(100 政大)(90、91 台大)：

$$T(n) = T(n/3) + T(2n/3) + n$$

1. 建立 Recursion Tree



$$T(n) = n + n + \dots + c$$

2. 求 bound

(1) 先求 upper bound(O)

$$T(n) = n + n + \dots + c \leq n + n + \dots + n = n^*h = n(\log_{3/2} n + 1) \Rightarrow O(n \lg n)$$

$$(2/3)kn = 1 \Rightarrow n(3/2)^k = 1 \Rightarrow k = \log_{3/2} n \Rightarrow h = k + 1 = \log_{3/2} n + 1$$

(2) 再求 lower bound(Ω)

$$T(n) = n + n + \dots + c \geq n^*h' = n(\log_3 n + 1) \Rightarrow T(n) = \Omega(n \lg n)$$

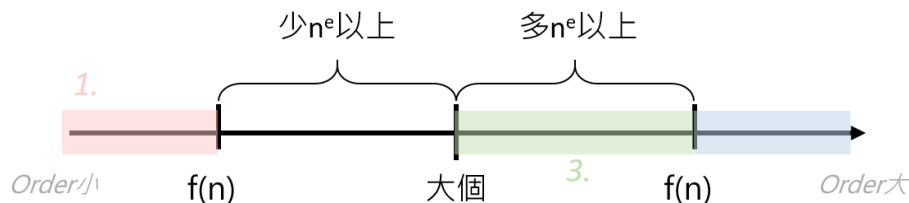
$$\Rightarrow T(n) = \Theta(n \lg n)$$

Master Theorem

($n^{\log_b a}$ 稱其為大個，而 Master Theorem 即與大個比較)

一、易錯題目

- 若存在 $\epsilon > 0$ ，使得 $f(n) = O(n^{\log_b a - \epsilon})$ ，則 $T(n) = \Theta(n^{\log_b a})$
- 若存在 $\epsilon > 0$ ， $0 < c < 1$ ，使得 $f(n) = \Omega(n^{\log_b a + \epsilon})$ ，且 $af(n/b) < cf(n)$ ，則 $T(n) = \Theta(f(n))$
- 若 $f(n) = \Theta(n^{\log_b a} \lg^k n)$ ， $k \geq 0$ ，則 $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$



例：

- (100 政大)： $T(n) = 7T(n/2) + n^2$
- (99 交大)： $T(n) = 3T(n/4) + n \lg n$
- (98 交大)： $T(n) = 3T(n/2) + n \lg n$
- $T(n) = 4T(n/2) + n^2 / \lg n$

- $n^{\lg 7}$, $f(n) = n^2$ ，存在 $\epsilon = \lg 7 - 2$ ，使得 $n^2 = O(n^{\lg 7 - \epsilon})$ ，則 $T(n) = \Theta(n^{\lg 7})$
- $n^{\lg 43}$, $f(n) = n \lg n$ ，存在 $\epsilon = 1 - \lg_4 3$ ， $c = 3/4$ ，使得 $n \lg n = \Omega(n^{\lg 43 + \epsilon})$ ，且 $3(n/4) \lg(n/4) < n \lg n$ ，則 $T(n) = \Theta(n \lg n)$
- $n^{\lg 3}$, $f(n) = n \lg n$ ，存在 $\epsilon = \lg 3 - 1.0001$ ，使得 $n \lg n = O(n^{\lg 3 - \epsilon})$ ，則 $T(n) = \Theta(n^{\lg 3})$
- (不可用 Master Theorem)