Master-theorem (老大定理)

O, Θ, Ω

(比較遞迴式中的 $f(n)$ 是否符合 3個條件中的一項)

遞迴式：

$$T(n) = aT\left(\frac{n}{b}\right) + f(n) \quad , \quad a \geq 1,\ b > 1 \quad f: f(n)$$

1. $f(n) = O(n^{(\log_b a) - \varepsilon})$

且 $\exists \varepsilon \, (\varepsilon > 0)$ ，則 $T(n) = \Theta(n^{\log_b a})$

2. $f(n) = \Theta(n^{\log_b a} \lg^k n)$

且 $\forall k \, (k \geq 0)$ ，則 $T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$

3. $f(n) = \Omega(n^{(\log_b a) + \varepsilon})$

且 $af\left(\frac{n}{b}\right) \leq cf(n)$ ，則 $T(n) = \Theta(f(n))$

且 $\exists \varepsilon \, (\varepsilon > 0)$

$\exists c \, (c < 1)$

Example:

We will evaluate the time complexity of a recursive algorithm A with input of n items. Algorithm A works as follow: when input size is m, the algorithm will first use $\Theta(\sqrt{m})$ steps to prepare and divide the input into 4 roughly equal-size subsets; for each size $\frac{m}{4}$ subset, recursively call A; finally it use $\Theta(\sqrt{m})$ steps to merge all 4 partial results to get the final solution. What is the time complexity for this algorithm?

Ans. 輸入m個後,程式執行次數:

$$T(m) = \sqrt{m} + T\left(\frac{m}{4}\right) \times 4 + \sqrt{m}$$

$\sqrt{m}$ ↓ 分解次數

$T\left(\frac{m}{4}\right) \times 4$ ↓ 分成$\frac{n}{4}$並呼叫自己,然後有4塊,所以乘4

$\sqrt{m}$ ↓ 合成次數

整理

$\Rightarrow T(m) = 4T\left(\frac{m}{4}\right) + 2\sqrt{m} \xrightarrow{\text{轉成共通式}} T(n) = 4T\left(\frac{n}{4}\right) + \Theta(n^{\frac{1}{2}})$

和 $\Theta(2n^{\frac{1}{2}})$ 相等,時間複雜度是不看係數的。

使用 master-theorem:

$a=4,\ b=4,\ f(n) = \Theta(n^{\frac{1}{2}})$ 取 $\varepsilon = \frac{1}{2}$,即

$\Rightarrow f(n) = O(n^{\log_4 4 - \varepsilon}) = O(n^{1-\frac{1}{2}}) = O(n^{\frac{1}{2}})$

$f(n) = O(n^{\frac{1}{2}})$ 符合,

$\therefore T(n) = \Theta(n^{\log_b a}) = \Theta(n^{\log_4 4}) = \Theta(n)$ #