Complexity of Divide and Conquer

證明複雜度

有 2 個方法來證明：

1. 用數學歸納法：猜一個複雜度，然後用數學歸納法證明滿足各種複雜度關係的定義。因為當中牽扯到把 n 比較小的 induction hypothesis 帶入，所以叫做「Substitution Method」。如果猜不到，可以用樹狀圖來輔助，叫做遞迴樹。
2. 大師定理：是一個有用的結論。照特定公式代進去。

在 I2A 這本書中實際上把遞迴樹列作另外一個方法，但該書在裡面提到遞迴樹只是一種「輔助找出複雜度形式」的一種方法，最後還是要用數學歸納法證明，因此併在一起。

Mathematic Induction

技巧 1：基本用法

$$T(n) = 2T(\lfloor n/2 \rfloor) + n$$

要找這東西的大 O。

直覺的來看：

$$\begin{aligned} T(n) &= 2T(n/2) + n \\ &= 2^2T(n/2^2) + 2 \cdot \frac{n}{2} + n \\ &= 2^3T(n/2^3) + 2^2 \cdot \frac{n}{2^2} + 2 \cdot \frac{n}{2} + n \\ &= \ldots \end{aligned}$$

這些步驟重複 $\lg(n)$ 次之後就會停，所以猜演算法複雜度是：

$$T(n) = O(n \lg(n))$$

要用歸納法證明這件事，假定：

$$\exists c, n_0 . \forall n > n_0 . T(\lfloor n \rfloor) \leq c \lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)$$

成立，接著用這個東西看看：

$$\begin{aligned}
\forall n > n_0 . T(n) &= 2T(\lfloor n/2 \rfloor) + n \\
&\leq 2\left(c\lfloor n/2 \rfloor \lg(\lfloor n/2 \rfloor)\right) + n && \text{帶入歸納法前提} \\
&\leq cn\lg(n/2) + n && \lfloor n/2 \rfloor < n/2 \text{且 } \lg \text{ 遞增} \\
&\leq cn\lg(n) - cn + n \\
&\leq cn\lg(n) && \forall c \geq 1
\end{aligned}$$

所以這邊找到了一個常數 c 的條件：

$$c \geq 1$$

接著看看 base case。可以發現 n = 1 時，$1\lg 1 = 0$ （真是一件尷尬的事）不過，大 O 的定義中，只有規定「$n > n_0$ 之後，都要小於 c 倍」，而 n_0 和 c 都是我們可以自己選的！所以只要訂出一個夠用的 n_0 就好。

可以觀察到：$n \geq 2$ 之後，所有的 $T(n)$ 最後都會被拆成只跟 $T(2)$ 跟 $T(3)$ 有關的組合。所以只要進一步限制常數，使 $T(2)$、$T(3)$ 成立就好。

以 $T(2)$ 為例，比較一下這時 $T(n)$ 跟 $n\lg n$ 的大小：

$$\begin{cases}
T(2) = 2 \cdot T(1) + 2 \\
\\
n\lg n|_{n=2} = 2 \cdot \lg 2 = 2
\end{cases}$$

所以比較一下，如果希望：

$$T(2) = 2 \cdot T(1) + 2 \leq c \cdot 2\lg 2$$

那麼這個 c 必須要滿足：

$$c \geq \frac{T(1) + 1}{\lg 2}$$

因此我們有了常數 c 的第二個條件。類似的道理：

$$T(3) = 2 \cdot T(1) + 3 \leq c \cdot 3\lg 3 \qquad \forall c \geq \frac{2T(1) + 3}{3\lg 3}$$

綜合以上兩個對 c 的限制：

$$\begin{cases} c \geq \frac{T(1)+1}{\lg 2} \\ \\ c \geq \frac{2T(1)+3}{3\lg 3} \\ \\ c \geq 1 \end{cases}$$

交集中的任何 c 都可以讓想要證的東西成立。比如說取：

$$c = max\{\frac{T(1)+1}{\lg 2}, \frac{2T(1)+3}{3\lg 3}, 1\}$$

並且令：

$$n_0 = 2$$

則可使：

$$\forall n > n_0 . T(n) \leq cn \lg n$$

因此原命題成立。

> *實際上，完全相同的方法可以證明 $T(n) = \Omega(n \lg n)$。只要取：*
>
> $$c = min\{\frac{T(1)+1}{\lg 2}, \frac{2T(1)+3}{3\lg 3}, 1\}$$
>
> *以及：*
>
> $$n_0 = 2$$
>
> *就可以了。*

技巧 2：假設比較強的條件

$$T(n) = T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1$$

因為感覺:

$$T(n) \approx 2^0 + 2 + 2^2 + \ldots + 2^{\lg n} \approx n$$

所以猜:

$$T(n) = O(n)$$

使用類似的方法，假定:

$$T(n/2) \leq c \cdot n$$

則:

$$\begin{aligned} T(n) &= T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1 \\ &\leq c \cdot \lfloor n/2 \rfloor + c \cdot \lceil n/2 \rceil + 1 \\ &= c \cdot n + 1 \end{aligned}$$

到這邊看起來像是證完了，但是沒有。因為 $T(n) \leq cn + 1$ 並不保證 $T(n) \leq cn$，這樣一來歸納法就沒辦法跑下去。碰到這種狀況可以假設一個比較強的條件。比如假定:

$$T(n/2) \leq c \cdot n - d$$

其中:

$$d \geq 0$$

這時:

$$\begin{aligned} T(n) &= T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1 \\ &\leq (c \cdot \lfloor n/2 \rfloor - d) + (c \cdot \lceil n/2 \rceil - d) + 1 \\ &= c \cdot n - 2d + 1 \\ &\leq c \cdot n - d && \forall c > 0. \forall d \geq 0 \end{aligned}$$

就會發現這個命題是會對的。更好康的是:

$$T(1) = 2 \cdot T(1) - T(1)$$

所以 $n=1$ 開始這個性質就會成立。因此取 $d=T(1)$、$c=2$ 就自動讓東西都成立了。接下來只是臨門一腳：因為上面已經證明對於對於任意 $n \geq 1$：

$$T(n) \leq c \cdot n - d \in O(n)$$

所以 $T(n) \in O(n)$，

> *可以證明 $T(n) \in \Omega(n)$ ，但這邊就不會有上面這個問題，因為顯然 $cn+1 \leq cn$。所以用歸納法前提時會是：*
>
> $$\begin{aligned} T(n) &= T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1 \\ &\geq c \cdot \lfloor n/2 \rfloor + c \cdot \lceil n/2 \rceil + 1 && \text{注意現在在證}\Omega \\ &= c \cdot n + 1 \\ &\geq cn \end{aligned}$$
>
> *所以不會出事。*

技巧 3：變數代換

$$T(n) = T(\lfloor \sqrt{n} \rfloor) + \lg n$$

感覺不能用剛剛的方法直接展開。不過如果把變數帶掉：

$$n = 2^m$$

則：

$$T(2^m) = T(2^{\frac{m}{2}}) + m$$

令：

$$S(m) = T(2^m)$$

可以發現：

$$S(m) = S(m/2) + m$$

但是上一個例子已經知道：

$$S(m) = O(m \lg m)$$

因此：

$$T(2^m) = T(n) = O((\lg n) \lg \lg n)$$

Master Theorem

$$T(n) = a \cdot T(n/b) + f(n)$$

若 $f(n)$ 的 order 跟 $\log_b a$ 的大小有以下關係：

1. $\exists \epsilon > 0 . f(n) \in O(n^{\log_n a - \epsilon}) \Rightarrow T(n) = \Theta(n^{log_b a})$
2. $f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{log_b a} \lg n)$
3. $$\begin{cases} \exists \epsilon > 0 . f(n) \in \Omega(n^{\epsilon + \log_n a}) \\ \exists c < 1, n_0 > 0 . \forall n > n_0 . \, af(n/b) \leq cf(n) \end{cases} \Rightarrow T(n) = \Theta(f(n))$$

這個定理是用 Polynomial Order 去比的。比如：

$$T(n) = 2T(n/2) + n \lg n$$

這時可以發現不存在能使 $n \lg n \in \Omega\left(n^{1+\epsilon}\right)$ 的 ϵ。因此沒辦法用大師定理得到什麼結論。也就是說 2. 跟 3. 的狀況之間有一些「漏掉」的狀況。

證明（不嚴謹版）

這是個特例的證明。假定「n 是 b 的整數次方」，而且遞迴式只在 n 是 b 的整數次方時有定義。即假定：

$$T(n) = a \cdot T(n/b) + f(n)$$

且:

$$\forall n. \exists k \in \mathbb{N}. n = b^k$$

Lemma 1

$$T(n) = \Theta\left(n^{log_b a}\right) + \sum_{i=0}^{\log_b n-1} a^i f(\frac{n}{b^i})$$

不斷對遞迴式做展開:

$$\begin{aligned} T(n) &= a \cdot T(n/b) + f(n) \\ &= a^2 \cdot T(n/b^2) + af\left(\frac{n}{b}\right) + f(n) \\ &\;\vdots \\ &= a^{\log_b n} T(1) + a^{\log_b a-1} f\left(\frac{n}{a^{\log_b a-1}}\right) + \ldots + af\left(\frac{n}{b}\right) + f(n) \\ &= \underbrace{a^{\log_b n}}_{n^{\log_b a}} T(1) + \sum_{i=0}^{\log_b n-1} a^i f(\frac{n}{b^i}) \\ &= \Theta\left(n^{\log_b a}\right) + \sum_{i=0}^{\log_b n-1} a^i f(\frac{n}{b^i}) \end{aligned}$$

因此:

$$T(n) = \Theta\left(n^{\log_b a}\right) + \sum_{i=0}^{\log_b n-1} a^i f(\frac{n}{b^i})$$

所以只要找 order 比較大的那一個就可以了。

> *Recall* : $max(f(n) + g(n)) = \Theta\left(f(n) + g(n)\right)$，*且* Θ *是 Symmetry* 。

Lemma 2

假定:

$$g(n) = \sum_{i=0}^{\log_b n - 1} a^i f\left(\frac{n}{b^i}\right)$$

則根據 $f(n)$ 的漸進性質，有以下的關係:

1. $\exists \epsilon > 0 . f(n) \in O\left(n^{\log_b a - \epsilon}\right) \Rightarrow g(n) \in O(n^{\log_b a})$
2. $f(n) \in \Theta(n^{\log_b a}) \Rightarrow g(n) \in \Theta(n^{\log_b a} \lg n)$
3. $\exists c < 1, n_0 . \forall n > n_0 . af(n/b) \leq cf(n) \Rightarrow g(n) \in \Theta(f(n))$

這邊先導一個之後會一直用的小東西:

$$\begin{aligned}\sum_{i=0}^{\log_b n - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a + \epsilon} &= n^{\log_b a + \epsilon} \sum_{i=0}^{\log_b n - 1} a^i \left(\frac{1}{b^i}\right)^{k+\epsilon} \\ &= n^{\log_b a + \epsilon} \sum_{i=0}^{\log_b n - 1} \left(\frac{a}{b^{\log_b a + \epsilon}}\right)^i \\ &= n^{\log_b a + \epsilon} \sum_{i=0}^{\log_b n - 1} \left(b^{-\epsilon}\right)^i\end{aligned}$$

Case 1

因為:

$$f(n) = O(n^{\log_b a - \epsilon})$$

故:

$$\exists c > 0, n_0 > 0. \forall n > n_0 . \, f(n) \leq cn^{\log_b a - \epsilon}$$

在這個前提之下：

$$
\begin{aligned}
g(n) = &\sum_{i=0}^{\log_b n - 1} a^i f\left(\frac{n}{b^i}\right) \\
&\leq c \cdot \sum_{i=0}^{\log_b n - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a - \epsilon} && \left(f(n) \in O(n^{\log_b a - \epsilon})\right) \\
&= c \cdot n^{\log_b a - \epsilon} \sum_{i=0}^{\log_b n - 1} (b^\epsilon)^i && \left(\text{一開始證的小東西}\right) \\
&= c \cdot n^{\log_b a - \epsilon} \left(\frac{b^{\epsilon \log_b n} - 1}{b^\epsilon - 1}\right) && \left(\text{套用等比級數公式}\right) \\
&= c \cdot n^{\log_b a - \epsilon} \left(\frac{n^\epsilon - 1}{b^\epsilon - 1}\right) && \left(b^{\log_b n} = n\right) \\
&\leq \left(\frac{c}{b^\epsilon - 1}\right) n^{\log_b a}
\end{aligned}
$$

因此：

$$g(n) = O(n^{\log_b a})$$

Case 2

因為：

$$f(n) = \Theta\left(n^{\log_b a}\right)$$

故：

$$\exists c_1 > 0, c_2 > 0, n_0 > 0. \forall n > n_0 . \, c_1 n^{\log_b a - 0} \leq f(n) \leq c_2 n^{\log_b a - 0}$$

分別證明 $g(n) \in O\left(n^{(\cdots)}\right)$ 與 $g(n) \in \Omega\left(n^{(\cdots)}\right)$實際上步驟跟剛剛一模一樣，只是步驟中令 $\epsilon = 0$：

$$g(n) = \sum_{i=0}^{\log_b n - 1} a^i f\left(\frac{n}{b^i}\right)$$

$$\leq_{(\geq)} c \cdot \sum_{i=0}^{\log_b n - 1} a^i \left(\frac{n}{b^i}\right)^{\log_b a - \epsilon} \qquad \text{(取 } c_2 \text{ 跟 } \leq \text{ 就是 } O \text{ 的證明;}$$

$$\text{取 } c_1 \text{ 跟 } \geq \text{ 就是 } \Omega \text{ 的證明)}$$

$$= c \cdot n^{\log_b a - \epsilon} \sum_{i=0}^{\log_b n - 1} (b^\epsilon)^i \qquad \text{(一開始證的小東西)}$$

$$= c \cdot n^{\log_b a - 0} \sum_{i=0}^{\log_b n - 1} \left(b^0\right)^i$$

$$= c \cdot n^{\log_b a} \cdot \log_b n$$

Case 3

假定:

$$\exists c < 1, n_0 . \forall n > n_0 . af(n/b) \leq cf(n)$$

首先顯然:

$$g(n) = \sum_{i=0}^{\log_b n - 1} a^i f\left(\frac{n}{b^i}\right) \geq f(n)$$

因此:

$$g(n) \in \Omega\left(f(n)\right)$$

又因為:

$$a^i f(n/b^i) \leq c \cdot \left(a^{i-1} f\left(\frac{a}{b^{i-1}}\right)\right)$$

一直遞推下去，可以知道：

$$
\begin{aligned}
a^i f(n/b^i) &\le c \cdot \left(a^{i-1} f\left(\frac{n}{b^{i-1}}\right)\right) \\
&\le c^2 \cdot \left(a^{i-2} f\left(\frac{n}{b^{i-2}}\right)\right) \\
&\;\;. \\
&\;\;. \\
&\;\;. \\
&\le c^i \cdot f(n)
\end{aligned}
$$

因此：

$$
\begin{aligned}
g(n) &= \sum_{i=0}^{\log_b n-1} a^i f\left(\frac{n}{b^i}\right) \\
&\le \sum_{i=0}^{\log_b n-1} c^i \cdot f(n) && (\text{上面遞推的結論}) \\
&\le \sum_{i=0}^{\infty} c^i \cdot f(n) && (\text{無窮項和} \ge \text{前幾項和}) \\
&= \left(\frac{1}{1-c}\right) f(n) && (c<1 \text{ 的等比級數和})
\end{aligned}
$$

因此：

$$
g(n) \in O(f(n))
$$

配合剛剛 $g(n) \in \Omega(f(n))$，可知：

$$
g(n) \in \Theta(f(n))
$$

參考資料

1. I2A：實際上大多數內容都是。
2. CMU 有另外一個大師定理的版本。但並沒有較嚴謹的證明。