Master Theorem

- Solving recurrence of the form: $T(n) = aT(n/b) + f(n)$
  where $a \geq 1$ anb $b > 1$

1、If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$

2、If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$

3、If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$

* For analysis, we limit n to be the powers of b. I.e. $n = b^i, \forall i = 0, 1, 2, \cdots$
We break the proof into 3 parts.

1、將原本解遞迴關係的問題化簡為一個解 summation 的問題.

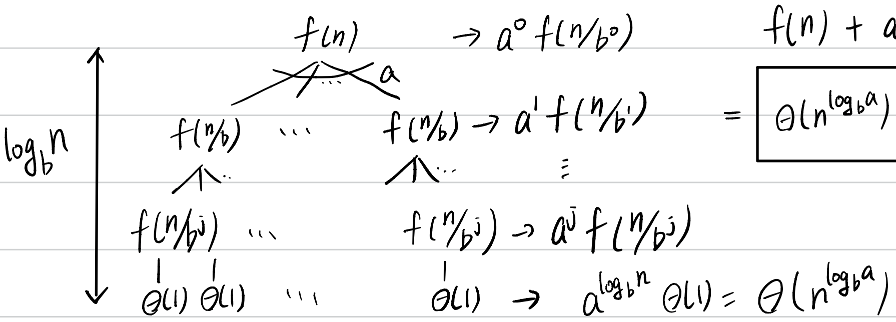
2、針對那個 summation 求出它的 upper bound.

3、把 (1)、(2) 合起來，證明當 $n = b^i$ 的時候，Master Theorem 會成立.

1、根據 Master Theorem 的關係式：$T(n) = \begin{cases} \Theta(1) & , n = 1 \\ aT(n/b) + f(n), & n = b^i \end{cases}$

目標求得 $T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$

proof：畫出 recursion tree：



$\Rightarrow$ Total cost $=$

$$f(n) + af(n/b) + \cdots + a^{\log_b n - 1} f(n/b^{\log_b n - 1}) + \Theta(n^{\log_b a})$$

$$= \boxed{\Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)}$$

#

- Total cost $= \Theta(n^{\log_b a}) + \sum_{j=0}^{\log_b n - 1} a^j f(n/b^j)$ = Leaf node + internal node
  = Cost of leaves + Cost of dividing and recombining subproblems.

- Corresponding to 3 cases:
  - Cost is dominated by leaves.
  - Cost is evenly distributed among the levels of the tree.
  - Cost is dominated by root.

2、令 $a \geq 1$ & $b > 1$，且 $g(n) = \sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$，$n = b^i$，則

1. If $f(n) = O(n^{\log_b a-\varepsilon})$ for some constant $\varepsilon > 0$, then $g(n) = O(n^{\log_b a})$
2. If $f(n) = \Theta(n^{\log_b a})$, then $g(n) = \Theta(n^{\log_b a}\lg n)$
3. If $af(n/b) \leq cf(n)$ for some constant $c < 1$ and for all sufficiently large n, then $g(n) = \Theta(f(n))$

proof: Case 1, 把 $f(n) = O(n^{\log_b a-\varepsilon})$ 代入

$$\Rightarrow g(n) = O\left(\sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a-\varepsilon}\right)$$

$$\Rightarrow \sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a-\varepsilon} = n^{\log_b a-\varepsilon}\sum_{j=0}^{\log_b n-1}\left(\frac{ab^\varepsilon}{b^{\log_b a}}\right)^j = n^{\log_b a-\varepsilon}\sum_{j=0}^{\log_b n-1}(b^\varepsilon)^j$$

$$= n^{\log_b a-\varepsilon}\left(\frac{b^{\varepsilon\log_b n}-1}{b^\varepsilon-1}\right) = n^{\log_b a-\varepsilon}\left(\frac{n^\varepsilon-1}{b^\varepsilon-1}\right)$$

$$= n^{\log_b a-\varepsilon}\,O(n^\varepsilon) = O(n^{\log_b a})$$

Case 2、一樣的作法 $\Rightarrow g(n) = \Theta\left(\sum_{j=0}^{\log_b n-1} a^j \left(\frac{n}{b^j}\right)^{\log_b a}\right)$

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

$$\Rightarrow \Theta(n^{\log_b a}\log_b n) = \Theta(n^{\log_b a}\lg n) \ \#$$

Case 3. ① 因為 $g(n)$ 每項都有 $f(n)$，所以 $g(n) = \Omega(f(n))$

② 根據假設 $af(n/b) \leq cf(n)$ 可得 $f(n/b^j) \leq (c/a)^j f(n)$，如果 n 夠大.

假設 n/b^{j-1} 夠大.滿足↑，則：$g(n) = \sum_{j=0}^{\log_b n-1} a^j f(n/b^j) \leq \sum_{j=0}^{\log_b n-1} c^j f(n) + O(1)$

（$O(1)$ 代表那些 n 不夠大的項）

$$\Rightarrow \sum_{j=0}^{\log_b n-1} c^j f(n) + O(1) \leq \sum_{j=0}^{\infty} c^j f(n) + O(1)$$

$$= f(n)\sum_{j=0}^{\infty} c^j + O(1)$$

$$= f(n)\left(\frac{1}{1-c}\right) + O(1)$$

$$= O(f(n))$$

$\Rightarrow g(n) = \Omega(f(n))$ && $g(n) = O(f(n)) \rightarrow g(n) = \Theta(f(n))$

3、(Master Theorem) Let $a \geq 1$, $b > 1$ be constants, and let $f(n) \geq 0 \ \forall \ n = b^i, i \in \mathbb{N}$

define $T(n) = \begin{cases} \Theta(1) & \text{, if } n = 1 \\ aT(n/b) + f(n), & \text{if } n = b^i \end{cases}$

a、If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = \Theta(n^{\log_b a})$

b、If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$

c、If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n, then $T(n) = \Theta(f(n))$.

proof: 已知 Total cost $= \Theta(n^{\log_b a}) + g(n)$

a.) $g(n) = O(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a}) + O(n^{\log_b a}) = \Theta(n^{\log_b a})$ #

b.) $T(n) = \Theta(n^{\log_b a}) + \Theta(n^{\log_b a} \lg n) = \Theta(n^{\log_b a} \lg n)$ #

c.) $T(n) = \Theta(n^{\log_b a}) + \Theta(f(n)) = \Theta(f(n))$ because $f(n) = \Omega(n^{\log_b a + \varepsilon})$ #

(把第2步 $g(n)$ 的值代入即可)