Master Theorem in Another Form

Master Theorem

- The order of growth of its solution *T(n)* depends on the values of the constants *a* and *b* and *the order of growth of the* function *f (n)*.

$$T(n) = aT(n/b) + f(n) \quad \text{where } f(n) \in \Theta(n^d), \quad d \geq 0$$

- The efficiency analysis of many divide-and-conquer algorithms is greatly simplified by the following theorem:

Master Theorem: If $a < b^d$, $T(n) \in \Theta(n^d)$

If $a = b^d$, $T(n) \in \Theta(n^d \log n)$

If $a > b^d$, $T(n) \in \Theta(n^{\log_b a})$

Note: The same results hold with O instead of Θ.

Master Theorem

$T(n) = aT(n/b) + f(n)$ where $f(n) \in \Theta(n^d)$, $d \geq 0$

If $a < b^d$, $T(n) \in \Theta(n^d)$

If $a = b^d$, $T(n) \in \Theta(n^d \log n)$

If $a > b^d$, $T(n) \in \Theta(n^{\log_b a})$

Solve these recurrence:

- $T(n) = 4T(n/2) + n^2$

- $T(n) = 16T(n/4) + n$

Master Theorem

$$T(n) = aT(n/b) + f(n) \quad \text{where } f(n) \in \Theta(n^d), \quad d \geq 0$$

If $a < b^d$, $\quad T(n) \in \Theta(n^d)$

If $a = b^d$, $\quad T(n) \in \Theta(n^d \log n)$

If $a > b^d$, $\quad T(n) \in \Theta(n^{\log_b a})$

Solve these recurrence:

- $T(n) = 4T(n/2) + n^2$ $\qquad$ $T(n) = 4T(n/2) + n^2 \Longrightarrow T(n) = \Theta(n^2 \log n)$
- $T(n) = 16T(n/4) + n$ $\qquad$ $T(n) = 16T(n/4) + n \Longrightarrow T(n) = \Theta(n^2)$