Divide-and-Conquer: Master Theorem

Neil Rhodes

Department of Computer Science and Engineering
University of California, San Diego

Data Structures and Algorithms
Algorithmic Toolbox

Outline

$$T(n) = T\left(\frac{n}{2}\right) + O(1)$$

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

$$T(n) = 4T\left(\frac{n}{2}\right) + O(n)$$

$$\downarrow$$

$$T(n) = O(n^2)$$

$$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$$

$$\downarrow$$

$$T(n) = O(n^{\log_2 3})$$

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

$$\downarrow$$

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

Master Theorem

Theorem

If $T(n) = aT\left(\left\lceil \frac{n}{b} \right\rceil\right) + O(n^d)$ (for constants $a > 0, b > 1, d \geq 0$), then:

$$T(n) = \begin{cases} O(n^d) & \text{if } d > \log_b a \\ O(n^d \log n) & \text{if } d = \log_b a \\ O(n^{\log_b a}) & \text{if } d < \log_b a \end{cases}$$

Master Theorem Example 1

$$T(n) = 4T\left(\frac{n}{2}\right) + O(n)$$

$$a = 4$$

$$b = 2$$

$$d = 1$$

Since $d < \log_b a$, $T(n) = O(n^{\log_b a}) = O(n^2)$

Master Theorem Example 2

$$T(n) = 3T\left(\frac{n}{2}\right) + O(n)$$
$$a = 3$$
$$b = 2$$
$$d = 1$$

Since $d < \log_b a$,
$T(n) = O(n^{\log_b a}) = O(n^{\log_2 3})$

Master Theorem Example 3

$$
\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + O(n) \\
a &= 2 \\
b &= 2 \\
d &= 1
\end{aligned}
$$

Since $d = \log_b a$,
$T(n) = O(n^d \log n) = O(n \log n)$

Master Theorem Example 4

$$T(n) = T\left(\frac{n}{2}\right) + O(1)$$

$$a = 1$$

$$b = 2$$

$$d = 0$$

Since $d = \log_b a$, $T(n) = O(n^d \log n) = O(n^0 \log n) = O(\log n)$

Master Theorem Example 5

$$
\begin{aligned}
T(n) &= 2T\left(\frac{n}{2}\right) + O(n^2) \\
a &= 2 \\
b &= 2 \\
d &= 2
\end{aligned}
$$

Since $d > \log_b a$, $T(n) = O(n^d) = O(n^2)$

Outline

1. What is the Master Theorem

2. Proof of Master Theorem

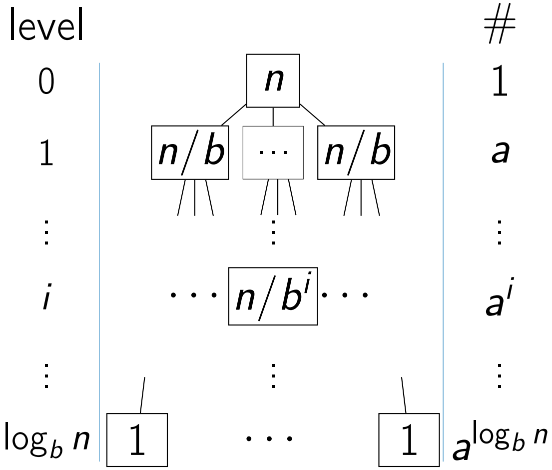
Master Theorem

Theorem

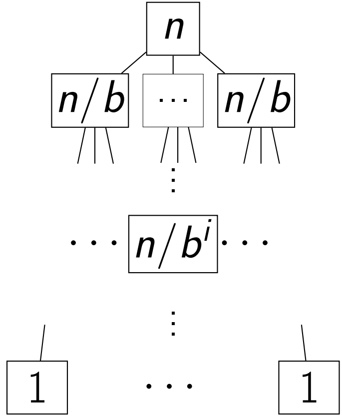
If $T(n) = aT\left(\left\lceil \frac{n}{b} \right\rceil\right) + O(n^d)$ (for constants $a > 0, b > 1, d \geq 0$), then:

$$T(n) = \begin{cases} O(n^d) & \text{if } d > \log_b a \\ O(n^d \log n) & \text{if } d = \log_b a \\ O(n^{\log_b a}) & \text{if } d < \log_b a \end{cases}$$

$$T(n) = aT\left(\left\lceil \frac{n}{b} \right\rceil\right) + O(n^d)$$



$$T(n) = aT\left(\left\lceil \frac{n}{b} \right\rceil\right) + O(n^d)$$



level	work
0	$O(n^d)$
1	$aO(\frac{n}{b})^d = O(n^d)\frac{a}{b^d}$
$\vdots$	$\vdots$
i	$a^i O\left(\frac{n}{b^i}\right)^d = O(n^d)\left(\frac{a}{b^d}\right)^i$
$\vdots$	$\vdots$
$\log_b n$	$a^{\log_b n} = O(n^{\log_b a})$

Total: $\sum_{i=0}^{\log_b n} O(n^d)\left(\frac{a}{b^d}\right)^i$

Geometric Series

For $r \neq 1$:

$$
\begin{aligned}
& a + ar + ar^2 + ar^3 + \cdots + ar^{n-1} \\
=& a\frac{1 - r^n}{1 - r} \\
=& \begin{cases} O(a) & \text{if } r < 1 \\ O(ar^{n-1}) & \text{if } r > 1 \end{cases}
\end{aligned}
$$

Case 1: $\frac{a}{b^d} < 1$ ($d > log_b a$)

$$\sum_{i=0}^{\log_b n} O(n^d)\left(\frac{a}{b^d}\right)^i$$

$$= O(n^d)$$

Case 2: $\frac{a}{b^d} = 1$ ($d = log_b a$)

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

Case 3: $\frac{a}{b^d} > 1$ ($d < log_b a$)

$$
\begin{aligned}
&\sum_{i=0}^{\log_b n} O(n^d)\left(\frac{a}{b^d}\right)^i \\
=&O\left(O(n^d)\left(\frac{a}{b^d}\right)^{\log_b n}\right) \\
=&O\left(O(n^d)\frac{a^{\log_b n}}{b^{d\log_b n}}\right) \\
=&O\left(O(n^d)\frac{n^{\log_b a}}{n^d}\right) \\
=&O(n^{\log_b a})
\end{aligned}
$$

Summary

Master theorem is a shortcut:

Theorem

If $T(n) = aT\left(\left\lceil \frac{n}{b} \right\rceil\right) + O(n^d)$ (for constants $a > 0, b > 1, d \geq 0$), then:

$$T(n) = \begin{cases} O(n^d) & \text{if } d > \log_b a \\ O(n^d \log n) & \text{if } d = \log_b a \\ O(n^{\log_b a}) & \text{if } d < \log_b a \end{cases}$$