Master Theorem

In the analysis of algorithms, the master theorem for divide-and-conquer recurrences provides an asymptotic analysis (using Big O notation) for recurrence relations of types that occur in the analysis of many divide and conquer algorithms. The approach was first presented by Jon Bentley, Dorothea Haken, and James B. Saxe in 1980, where it was described as a "unifying method" for solving such recurrences. The name "master theorem" was popularized by the widely used algorithms textbook Introduction to Algorithms by Cormen, Leiserson, Rivest, and Stein.

Not all recurrence relations can be solved with the use of this theorem; its generalizations include the Akra–Bazzi method.

Master Theorem

General format

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

where $a \geq 1$, $b > 1$ and $f(n)$ should +ve always.

Cases:

1> $f(n) < n^{\log_b a}$

$T(n) = \Theta(n^{\log_b a})$

2> $f(n) = n^{\log_b a}$

$T(n) = \Theta(n^{\log_b a} . \log n)$

3> $f(n) > n^{\log_b a}$

$T(n) = \Theta f(n)$

1) $T(n) = 4T(n/2) + n$

Solun Given :

$a = 4$

$b = 2$

$f(n) = n$

$\therefore n^{\log_b a} = n^{\log_2 4} = n^2$

$\therefore f(n) < n^{\log_b a}$ case I

$T(n) = \Theta(n^{\log_b a})$

$$\boxed{T(n) = \Theta(n^2)}$$

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

Solun: Given =

$a = 2$

$b = 2$

$f(n) = n$

$\therefore n^{\log_b a} = n^{\log_2 2} = n$

$\therefore f(n) = n^{\log_b a}$ case II

$T(n) = \Theta(n^{\log_b a} \cdot \log n)$

$$\boxed{T(n) = \Theta(n \cdot \log n)}$$

Page No.

1) $T(n) = 2T(n/2) + n^2$

Solun Given $a = 2$

$b = 2$

$f(n) = n^2$

$n^{\log_b a} = n^{\log_2 2} = n$

$\therefore f(n) > n^{\log_b a}$ case III

$T(n) = \Theta f(n)$

$T(n) = \Theta(n^2)$

2) $T(n) = 4T(n/2) + n^3$

Solun Given. $a = 4$

$b = 2$

$f(n) = n^3$

$n^{\log_b a} = n^{\log_2 4} = n^2$

$\therefore f(n) > n^{\log_b a}$ case III

$T(n) = \Theta f(n)$

$T(n) = \Theta(n^3)$

Date: / /
Page No.

3> $T(n) = T(9n/10) + n$

Solun:- $T\left(\frac{9n/9}{10/9}\right) + n$

$T(n) = T\frac{n}{1.1} + n$

$a = 1 \qquad f(n) = n$

$b = 1.1$

$n^{\log_b a} = n^{\log_{1.1} 1} = n^0$

$\therefore f(n) > n^{\log_b a} \qquad T(n) = \Theta f(n)$

$T(n) = \Theta(n)$

4> $T(n) = 8T(n/4) + n^2$

Solun given $a = 8$

$b = 4$

$f(n) = n^2$

$n^{\log_b a} = n^{\log_4 8} = n^{3/2}$

$\therefore f(n) > n^{\log_b a}$

$T(n) = \Theta f(n)$

$T(n) = \Theta(n^2)$

1> $T(n) = 4T(n/2) + n^3$

Solun: Given $a = 4$

$b = 2$

$f(n) = n^3$

$n^{\log_b a} = n^{\log_2 4} = n^2$

$\therefore f(n) > n^{\log_b a}$ case III

$T(n) = \Theta f(n)$

$T(n) = \Theta(n^3)$

2> $T(n) = 3T\left(\frac{2n}{6}\right) + n$

Soln $3T\left(\frac{2n/2}{6/2}\right) + n$

$3T(n/3) + n$

$a = 3$

$b = 3$

$f(n) = n$

$n^{\log_3 a} = n^{\log_3 3} = n$

$\therefore f(n) = n^{\log_b a}$ case II

$T(n) = \Theta(n^{\log_b a} \cdot \log n)$

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

Date: / /
Page No.

3) $T(n) = 8T(n/4) + n$

Solun:

given $a = 8$
$b = 4$
$f(n) = n$.

$n^{\log_b a} = n^{\log_4 8} = n^{3/2}$

$\therefore f(n) < n^{\log_b a}$ case I

$T(n) = \Theta(n^{\log_b a})$
$T(n) = \Theta(n^{3/2})$

.

4) $T(n) = 16T(n/8) + n^2$

Soln:-

given $a = 16$
$b = 8$
$f(n) = n^2$

$n^{\log_b a} = n^{\log_8 16} = n^{4/3}$

$\therefore f(n) > n^{\log_b a}$ case III

$T(n) = \Theta f(n)$
$T(n) = \Theta(n^2)$