MASTERS METHOD

Master's theorem is one of the many methods that are applied to calculate time complexities of algorithms. In analysis, time complexities are calculated to find out the best optimal logic of an algorithm. Master's theorem is applied on recurrence relations.

But before we get deep into the master's theorem, let us first revise what recurrence relations are –

Recurrence relations are equations that define the sequence of elements in which a term is a function of its preceding term. In algorithm analysis, the recurrence relations are usually formed when loops are present in an algorithm.

Problem Statement

Master's theorem can only be applied on decreasing and dividing recurring functions. If the relation is not decreasing or dividing, master's theorem must not be applied.

Master's Theorem for Dividing Functions

Consider a relation of type −

$$T(n) = aT(n/b) + f(n)$$

where, **a >= 1** and **b > 1**,

n − size of the problem

a − number of sub-problems in the recursion

n/b − size of the sub problems based on the assumption that all sub-problems are of the same size.

f(n) − represents the cost of work done outside the recursion -> Θ(nk logn p) ,where k >= 0 and p is a real number;

If the recurrence relation is in the above given form, then there are three cases in the master theorem to determine the asymptotic notations −

- If $a > b^k$, then $T(n)= \Theta (n^{\log_b a})$ [$\log_b a = \log a / \log b$.]
- If $a = b^k$

- If $p > -1$, then $T(n) = \Theta (n^{\log_b a} \log^{p+1} n)$
- If $p = -1$, then $T(n) = \Theta (n^{\log_b a} \log \log n)$
- If $p < -1$, then $T(n) = \Theta (n^{\log_b a})$

- If $a < b^k$,
  - If $p >= 0$, then $T(n) = \Theta (n^k \log^p n)$.
  - If $p < 0$, then $T(n) = \Theta (n^k)$

Master's Theorem for Decreasing Functions

Consider a relation of type –

$T(n)=aT(n-b)+f(n)$

Where $a>1$ and $b>1$,$f(n)$ is asymptotically positive

Here,

n − size of the problem

a − number of sub-problems in the recursion

n-b − size of the sub problems based on the assumption that all sub-problems are of the same size.

f(n) − represents the cost of work done outside the recursion -> $\Theta(n^k)$, where $k >= 0$.

If the recurrence relation is in the above given form, then there are three cases in the master theorem to determine the asymptotic notations −

- if $a = 1$, $T(n) = O (n^{k+1})$
- if $a > 1$, $T(n) = O (a^{n/b} * n^k)$
- if $a < 1$, $T(n) = O (n^k)$