The Master Method is a formula for solving recurrence relation of form

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

where,

n = size of input

a = number of subproblems in the recursion

$\left(\frac{n}{b}\right)$ = size of each subproblem. All subproblems are assumed to have the same size.

$f(n) =$ cost of the work done outside the recursive call, which includes the cost of dividing the problem & cost of merging solution.

Here, $\boxed{a \geq 1}$ & $\boxed{b > 1}$ are constants & $f(n)$ is an asymptotically positive function.

An asymptotically positive function means that for a sufficiently large value of n we have $\boxed{f(n) > 0}$.

This master theorem is used to calculate divide & conquer problems time complexity in an easy way.

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

where, $T(n)$ has the following asymptotic bounds:

1. If $f(n) = O(n^{\log_b a - \epsilon})$,

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

2. If $f(n) = \Theta(n^{\log_b a} \log^k n)$

$\boxed{K \geq 0}$ $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$

3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$

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

1 & 3 for some $\boxed{\epsilon > 0}$

$\boxed{\epsilon \to \text{constant}}$

3 condition is only possible with regularity condition.

$a f\left(\frac{n}{b}\right) \leq c f(n)$ for $\boxed{c < 1}$

& all sufficiently large n

Examples:-

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

$a = 1$, $b = 2$, $f(n) = c$

Master theorem conditions

$a \geq 1$ & $b > 1$ & $f(n)$ is asym. positive

It satisfies all three

So, $\Rightarrow \log_b a \Rightarrow \log_2 1 \Rightarrow 0$

$n^{\log_b a} \Rightarrow n^0 \Rightarrow \boxed{1}$

$f(n) = C$ & $n^{\log_2 a} = 1$

It satisfies 2nd condition

$$f(n) = \Theta(n^{\log_b a} \log^k n)$$

here $K = 0$

$$f(n) = \Theta(1 \times 1) = C$$

So, $T(n) = \Theta(n^{\log_b a} \times \log^{k+1} n)$

$$\boxed{T(n) = \Theta(1 \times \log n)}$$

(ii) $T(n) = 4T(\frac{n}{2}) + cn$

$a = 4, b = 2$ & $f(n) = cn$

So, satisfies Master Theorem condition

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

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

It satisfies first condition

$f(n) = O(n^{\log_b a - \epsilon})$

for $\boxed{\epsilon = 2}$

$cn = f(n) = O(n^{\log_2 (4-2)})$

$cn = f(n) = O(n^{\log_2 2})$

$\Rightarrow O(n)$

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

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

$a = 3$, $b = 2$ & $f(n) = n^2$

$n^{\log_b a} \Rightarrow n^{\log_2 3}$

It satisfies third condition

$f(n) = \Theta(n^{\log_b a + \epsilon})$

$n^2 = f(n) \Rightarrow \Theta(n^{\log_2 3 + 1})$

$n^2 \Rightarrow \Theta(n^2)$ for $\boxed{\epsilon = 1}$

Check for regularity

$a f\left(\frac{n}{b}\right) \leq c f(n)$

put a & b

$3 f\left(\frac{n}{2}\right) \leq c f(n)$

put $f(n) = n^2$

$3 \times \frac{n^2}{(2)^2} \leq c n^2$

$\frac{3}{4} n^2 \leq c n^2$

we can take $\boxed{c = 0.8}$

then this is true.

So,

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

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