Master's Method (Master's theorem)

The recurrence relation is solved is master's theorem.

Substitution method can be used for solving any recurrence relation. but it is slow.

But this not the case about master's method.
It can solve only ~~recurrence~~ a few recurrence relations which are in the format given below,

① $$\boxed{T(n) = a \cdot T\left(\frac{n}{b}\right) + f(n)}$$

where $\boxed{a \geq 1}$ and $\boxed{b > 1}$

Also $f(n)$ must be a positive funⁿ

e.g. $T(n) = T(n-1) + 1$

Here $T(n) = 1 \cdot T\left(\frac{n-1}{1}\right) + 1$

so $a = 1$ & $b = 1$

But for master's theorem we require $b > 1$

so this equation can not be solved using master's method

Page :
Date :

③ ~~Solution~~ is $T(n) = 8T\left(\frac{n}{2}\right) + n^2$

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

These 2 eqn's can be solved using master's method.

There are 3 cases for master's theorem

$\boxed{\text{Case I:}}$ If $T(n) = O(n^{\log_b a - \epsilon})$
for constant $\epsilon > 0$ then
$T(n) = \Theta(n^{\log_b a})$
(avg case)

For Big O notation the fun^n which we have is
$f(n) \leq c \cdot g(n)$
we ~~we~~ need to show $T(n) = \Theta(n^{\log_b a})$
we need to make both terms ~~wit~~ as equal
$n^1 \leq n^2$
$n^1 = n^{2-1}$ ← we need substract ~~it~~ ~~with~~ some constant

$\boxed{\text{Case II:}}$ If $T(n) = \Omega(n^{\log_b a + \epsilon})$
for constant $\epsilon > 0$ then
$T(n) = \cancel{\Omega}\ \Theta(f(n))$

For Ω notation the fun^n is
$f(n) \geq c \cdot g(n)$
We need to show $T(n) = \Theta(f(n))$
we need to make both the terms as equal
$\cancel{n^2}$
$n^3 \geq n^1$

$n^3 = n^{1+2}$ ← we need to add some constant

Case III :

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

As both the terms are equal ~~not~~ ~~≥~~ neither ≥ nor ≤ in this we do not have to add or substract the constant ϵ.

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

Just multiply the term with $\log n$

Example 1 based on case I

Give recurrence relation is

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

Pattern of recurrence relⁿ for master's method is

$$T(n) = aT(n/b) + f(n)$$
$a \geq 1$, $b > 1$, $f(n)$ must be +ve funⁿ

As the given recurrence relation is in the form of above equation let's find value for a, b & f(n)

We have $a = 8$, $b = 2$, $f(n) = n^2$

Find, $n^{\log_b a} = n^{\log_2 8}$ This gives us value of g(n)

$$\boxed{n^{\log_b a} = n^3}$$

(compare n^2 & $n^{\log_b a}$ i.e. n^3)

As per case I of master's theorem we have
$$f(n) \leq c \cdot g(n)$$
$$n^2 \leq c \cdot n^3 \text{ (True)}$$

For this case
$$T(n) = \Theta(n^{\log_b a - \epsilon})$$

$$\boxed{T(n) = \Theta(n^3)}$$ consider the larger term amongst n^3 & n^2 as we need upper bound

Example 2 based on case ~~II~~ I

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

So, $a = 9$, $b = 3$, $f(n) = n$

$$n^{\log_b a} = n^{\log_3 9} = n^2$$

As per case I

$$f(n) \leq c \cdot g(n)$$
$$n \leq c \cdot n^2 \quad \text{(True)}$$

So final answer of Time complexity of given recurrence relation is,

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

consider the term which is largest amongst n^2 & n as we need upper bound

Example 3	based on case II

$T(n) = 64\,T(n/2) + n^7$

Solve the given recurrence relation.

First check if it is in the given format or not.

$$T(n) = a\,T(n/b) + f(n) \quad \left.\right\} \begin{array}{l} a \geq 1 \\ b > 1 \\ f(n) \text{ +ve fun}^n \end{array}$$

~~Fr~~ As the given ~~req~~ equation is in the above format we have,

$$a = 64, \quad b = 2, \quad f(n) = n^7$$

First calculate $n^{\log_b a}$ to get value of $g(n)$

$$n^{\log_b a} = n^{\log_2 64}$$

$$n^{\log_b a} = n^{\log_2 2^6}$$

$$\boxed{n^{\log_b a} = n^6} \quad \text{as } \log_2 2 = 1$$

~~This~~ This is the value of $g(n)$

So we have $f(n) = n^7$ & $g(n) = n^6$

as per case II of master's theorem we have,

$$f(n) \geq c \cdot g(n)$$
$$n^7 \geq c \cdot n^6 \quad (\text{True})$$

So time complexity of given recurrence relation is.

$\boxed{T(n) = \Theta(n^7)}$ (consider largest term from n^6 & n^7)