master method → (For recurrance relation) →

- Cons → We cannot solve all kinds of recurrance relation with this method.

- we can only solve relations in the form of →

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

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

Then the solution is →

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

a, b are some constants f(n) is some +ve function.

Rule for master th^m →
The given equation must be polynomial equation.

$\frac{n^3}{n^2} = n^1$ → polynomial form (minimum power 1)

Q $2T\left(\frac{n}{2}\right) + n\log n$

$a = 2, b = 2$

$n + n\log n$ — master th^m not applicable as $\frac{n}{n\log n}$ not a polynomial equation.

Q $T(n) = 8T\left(\frac{n}{2}\right) + n^2$

$a = 8,\ b = 2$

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

$\Rightarrow n^{\log_2 8}$

$= n^3 + n^2$

$T(n) = O(n^3) + O(n^2) \Rightarrow T(n) = O(n^3)$

Worst Case

Q $4T\left(\frac{n}{2}\right) + n^5$

$T(n) = n^{\log_2 4}$

$= n^{2\log_2 2}$

$= n^{2(1)} + n^5$

$= O(n^5)$ A

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

$a = 2,\ b = 2$

$n^{\log_2 2}$

$\Rightarrow n + n$

since they are same, so add $\log n$

$O(n \log n)$

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

$\Rightarrow a = 1,\ b = 2$

$= n^{\log_2 1} + c$ $\qquad \log 1 = 0$
$= n^0$
$= 1$

$= 1 + c$
$= 1 + c$
so, $O(\log n)$

Q <u>Efficiency of Binary search -></u>

- <u>Binary search</u> -> It is based on divide & conquer

$n,\ n/2,\ n/4 \ldots$
$\frac{n}{2^k} = 1$ (Because at last 1 element is there)

$n = 2^k$
Take log on both sides
$\log n = k \log 2$
$\boxed{\log n = k}$

Best case (when $n = 1$)
$\Rightarrow O(1)$
Worst case (when element is not present)
$\Rightarrow O(\log n)$