Some recurrence relations are of the form :-

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

where; $a \rightarrow$ No. of subproblems.

$\left(\frac{n}{b}\right) \rightarrow$ size of each subproblem

$f(n) \rightarrow$ function which denotes the time taken to combine the result of subproblems.

$T(n) \rightarrow$ time taken to solve a problem of size n.

Recurrences of these forms can be solved by using :-

I) Recursion Tree Method.

II) Master Method.

III) Change of Variable Method.

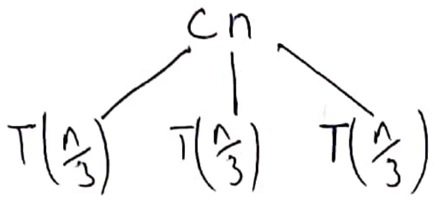
(I) Recursion Tree Method

→ It is a pictorial representation of an iteration method, which is in the form of a tree, where at each level nodes are expanded.

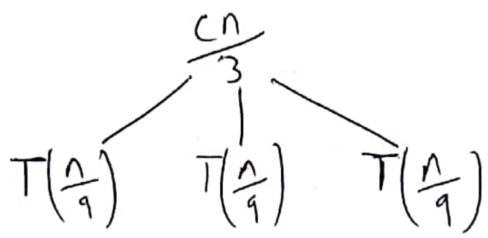
→ Generally, the second term in recurrence is considered as "root"

→ It is useful when the divide-and-conquer algorithm is used.

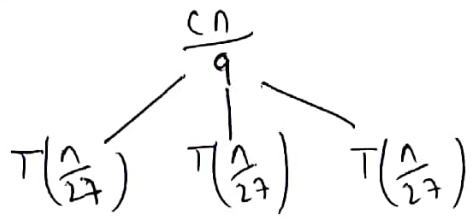
1) $T(n) = 3T\left(\frac{n}{3}\right) + cn$



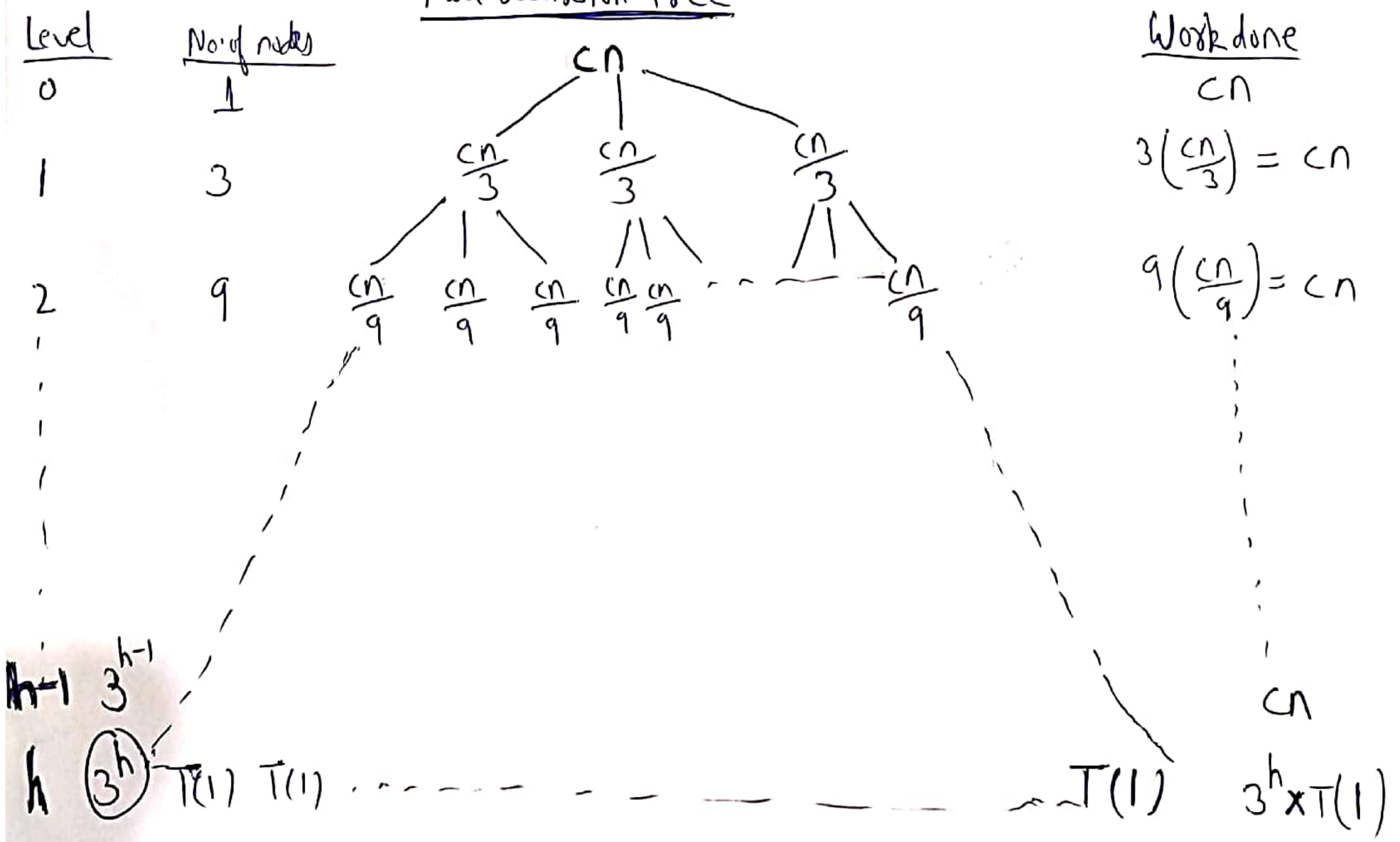
$T\left(\frac{n}{3}\right) = 3T\left(\frac{n}{9}\right) + \frac{cn}{3}$



$T\left(\frac{n}{9}\right) = 3T\left(\frac{n}{27}\right) + \frac{cn}{9}$



<u>Full recursion tree</u>



Total work done (time taken) :-

$$T(n) = \sum_{i=0}^{h-1} cn + 3^h \times T(1) \longrightarrow (1)$$

~~*or such other certificate by whatever name called issued by the competent authority.~~

[$\because$ From level 0 to level $(h-1)$; work done $= cn$
and at the level h; work done $= 3^h(T(1))$.]

$$= cn \sum_{i=0}^{h-1} 1 + 3^h \times T(1)$$

$$\boxed{T(n) = cnh + 3^h \times T(1)} \longrightarrow (2)$$

We can analyse from the recursion tree that,

$$\boxed{\frac{n}{3^h} = 1} \Rightarrow n = 3^h$$

$$\Rightarrow \boxed{h = \log_3 n} \longrightarrow (3)$$

Putting eqn (3) in eqn (2) we get,

$$T(n) = cn\log_3 n + n \times T(1)$$

$$\boxed{T(n) = \Theta(n\log_3 n)}$$

(As $n \times T(1)$ is non-dominant term as compared to $cn\log_3 n$; so $n \times T(1)$ can be ignored).

---

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

$$n^2$$

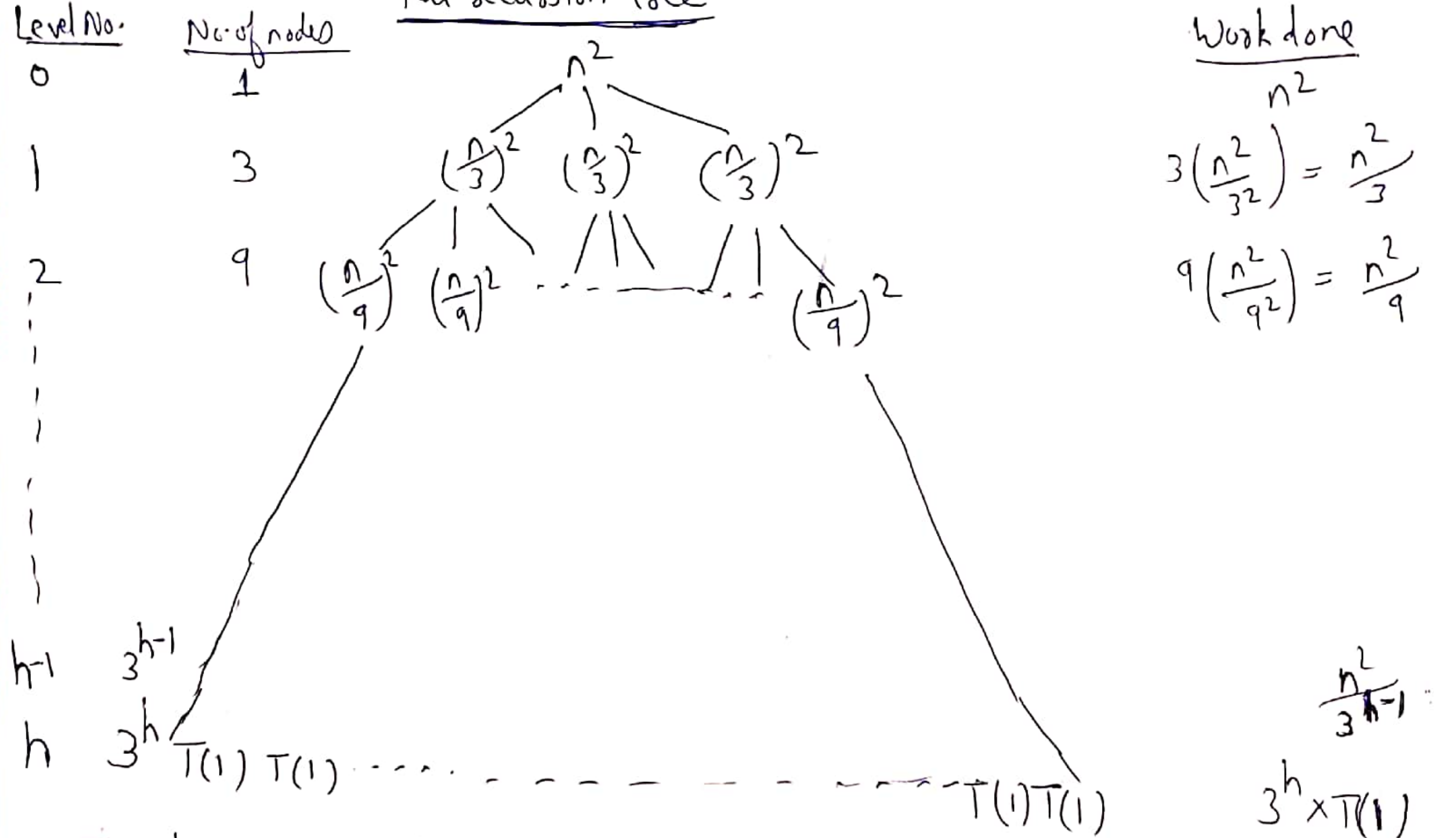
$$T\left(\frac{n}{3}\right) \quad T\left(\frac{n}{3}\right) \quad T\left(\frac{n}{3}\right)$$

$$T\left(\frac{n}{3}\right) = 3T\left(\frac{n}{9}\right) + \left(\frac{n}{3}\right)^2$$

$$\left(\frac{n}{3}\right)^2$$

$$T\left(\frac{n}{9}\right) \quad T\left(\frac{n}{9}\right) \quad T\left(\frac{n}{9}\right)$$

Full recursion tree



Total work done (time taken) :-

$$T(n) = \sum_{i=0}^{h-1} \frac{n^2}{3^i} + 3^h \times T(1)$$

$$= n^2 \sum_{i=0}^{h-1} \left(\frac{1}{3^i}\right) + 3^h \times T(1)$$

$$= n^2 \left[1 + \frac{1}{3} + \frac{1}{3^2} + \cdots + \frac{1}{3^{h-1}}\right] + 3^h \times T(1)$$

$$= n^2 \left[\frac{1\left(1-\left(\frac{1}{3}\right)^h\right)}{\left(1-\frac{1}{3}\right)}\right] + 3^h T(1)$$

$$= n^2 \left[\frac{3^h - 1}{3^h} \times \frac{3}{2}\right] + 3^h T(1)$$

$$T(n) = \frac{3}{2} n^2 \left(\frac{3^h - 1}{3^h}\right) + 3^h T(1) \longrightarrow (1)$$

We can analyse from the recursion tree that,

$$\boxed{\frac{n}{3^h} = 1} \implies n = 3^h$$

$$\implies \boxed{h = \log_3 n} \longrightarrow (2)$$

Putting eq^n (2) in eq^n(1) we get,

$$T(n) = \frac{3}{2} n^2 \left(\frac{n-1}{n}\right) + n\,T(1)$$

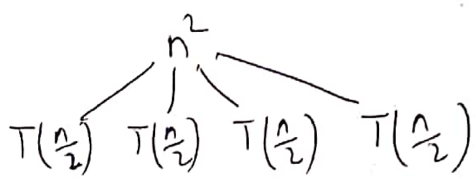
$$= \frac{3}{2} n(n-1) + n\,T(1)$$

$$= \frac{3}{2} n^2 - \frac{3}{2} n + n\,T(1)$$

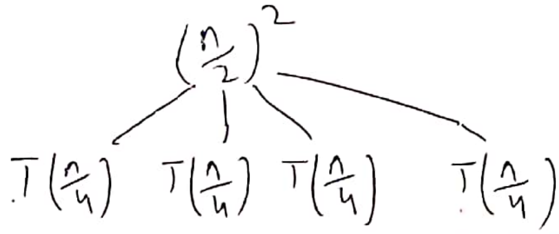
$$\boxed{T(n) = \Theta(n^2)}$$ (Ignoring $-\frac{3}{2}n + n\,T(1)$ with respect to $\frac{3}{2}n^2$)

---

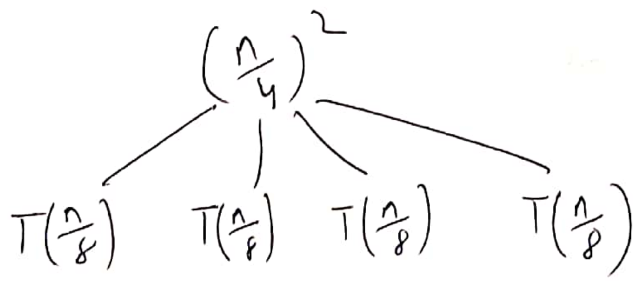
3) $T(n) = 4T\left(\frac{n}{2}\right) + n^2$



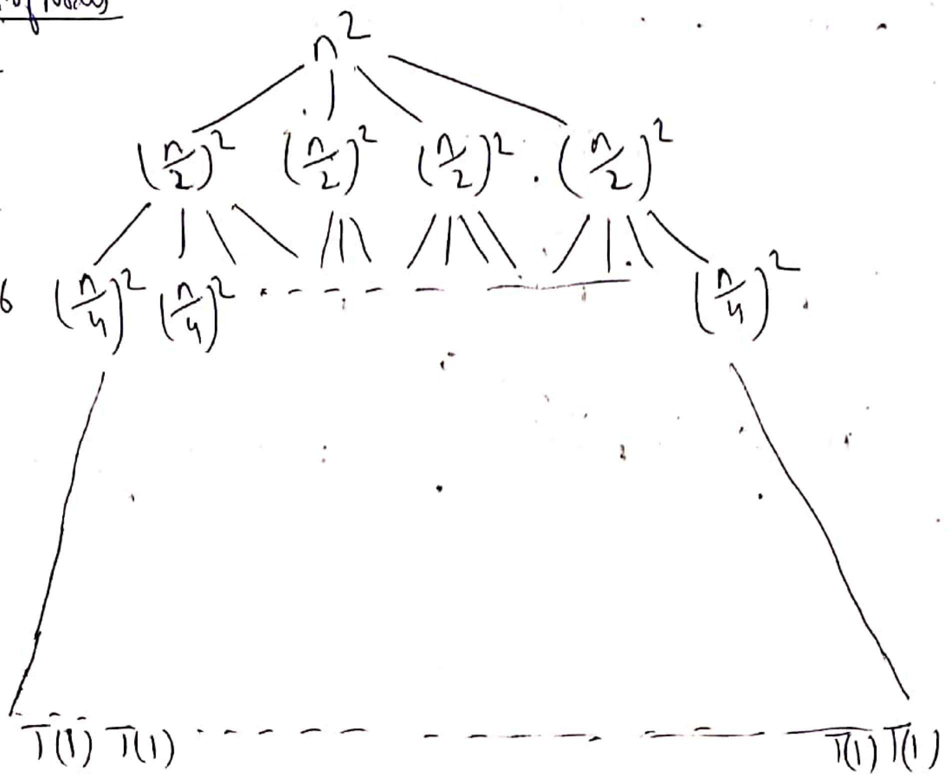
$$T\left(\frac{n}{2}\right) = 4T\left(\frac{n}{4}\right) + \left(\frac{n}{2}\right)^2$$



$$T\left(\frac{n}{4}\right) = 4T\left(\frac{n}{8}\right) + \left(\frac{n}{4}\right)^2$$



Full recursion tree



Level No.	No. of Nodes	Work done
0	1	n^2
1	4	$4\left(\frac{n^2}{4}\right) = n^2$
2	16	$16\left(\frac{n^2}{16}\right) = n^2$
⋮		
h-1	4^{h-1}	$4^{h-1}\left(\frac{n^2}{4^{h-1}}\right) = n^2$
h	4^h	$4^h \times T(1)$

Total work done (time taken) :-

$$T(n) = \sum_{i=0}^{h-1} n^2 + 4^h T(1)$$

$$= n^2 \sum_{i=0}^{h-1} (1) + 4^h T(1)$$

$$\boxed{T(n) = n^2 h + 4^h T(1)} \longrightarrow (1)$$

We can analyse from the recursion tree that,

$$\boxed{\frac{n}{2^h} = 1} \Rightarrow n = 2^h$$

$$\Rightarrow \boxed{h = \log_2 n} \longrightarrow (2)$$

Putting eq^n (2) in eq^n (1) we get,

$$T(n) = n^2 \log_2 n + 4^{\log_2 n} T(1)$$

$$= n^2 \log_2 n + n^{\log_2 4} T(1)$$

$$T(n) = n^2 \log_2 n + n^2 T(1)$$

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