1. Show That, $T(n) = T(\lfloor n/2 \rfloor) + 1 = O(\log n)$ - using substitution method.

Solution:-

We have,

$$T(n) = T(\lfloor n/2 \rfloor) + 1$$

$$= [\,T(\,n/4 + 1)\,] + 1$$

$$= T(\,n/4) + 1 + 1$$

$$= [\,T(\,n/8)\ + 1\,] + 1 + 1$$

$$= T(\,n/8)\,] + 1 + 1 + 1$$

$$= [T(\,n/16 + 1)] + 1 + 1 + 1$$

$$= T(n/16) + 1 + 1 + 1 + 1$$

So, at step i, $T(n) = T(n/2^i) + \text{i}.1 + \cdots\cdots\cdots\cdots$ (i)

For the base case to occur -

$$n/2^i = 1$$

$$=> \quad 2^i = n$$

$$=> \quad \log_2 2^i = \log_2 n$$

$$=> \quad \text{i} = \log_2 n$$

Putting the value if i in eq(i), we get

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

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

$$= T(n/n) + \log_2 n$$

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

$$= \theta(\log_2 n)$$

$$= O(\log_2 n)$$

---

2. Show That, $T(n) = 2T(\lfloor n/2 \rfloor) + n = \Omega(n \log n)$

<u>Solution:-</u>

We have,

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

$= 2[\ 2T(n/2^2) + n/2\ ] + n$

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

$= 2^2\ [\ 2T(n/2^3) + (n/2^2)\ ] + n + n$

$= 2^3\ T(n/2^3) + n + n + n$

$= 2^3\ [\ 2T\ (n/2^4) + (n/2^3)\ ] + n + n + n$

$= 2^4\ T\ (n/2^4) + n + n + n + n$

So, at step i ,

$T(n) = 2^i\ T(n/2^i) + n.i \ldots \ldots \ldots$ (i)

For the base case to occur,

$n/2^i = 1$

$=> \quad 2^i = n$

$=> \quad \log_2 2^i = \log_2 n$

$=> \quad i = \log_2 n$

Putting the value of i, in eq(i), we get –

$$T(n) = 2^{\log_2 n}\ T({}^{n}/_{2^{\log_2 n}}) + \text{n} \log_2 n$$

$$= n^{\log_2 2}\ T({}^{n}/_{n^{\log_2 2}}) + \text{n} \log_2 n$$

$$= \text{n}\ T(1) + \text{n} \log_2 n$$

$$= \theta(\text{n} \log_2 n)$$

$$= \Omega(\text{n log n})$$

---

4. Solve $T(n) = 2T(\sqrt{n}) + 1$.

<u>Solution:-</u>

$$T(n) = 2T(\sqrt{n}) + 1$$

$$= 2\left[2 \cdot T\left(n^{1/2^2}\right) + 1\right] + 1$$

$$= 2^2\ T\left(n^{1/2^2}\right) + 2 + 1$$

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

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

$$= 2^3\left[2T\left(n^{1/2^4}\right) + 1\right] + 2^2 + 2 + 1$$

$$= 2^4\ T\left(n^{1/2^4}\right) + 2^3 + 2^2 + 2 + 1$$

At step i,

$$T(n) = 2^i\ T\left(n^{\left(1/2\right)^i}\right) + \sum_{k=0}^{i-1} 2^k$$

So, for the best case to occur

$$\left(n^{1/2}\right)^i = 1$$

$\Rightarrow \log_{n^{1/2}} 1 = i$

$\Rightarrow \quad i = 0$

So, $T(n) = 2^i \, T\left(n^{(1/2)^i}\right) + \frac{1\cdot(2^i-1)}{2-1}$

$= 2^0 \cdot T\left(n^{(1/2)^0}\right) + 2^0 - 1$

$= 2^0 \cdot T(n^1) + 2^0 - 1$

$= T(n)$

---

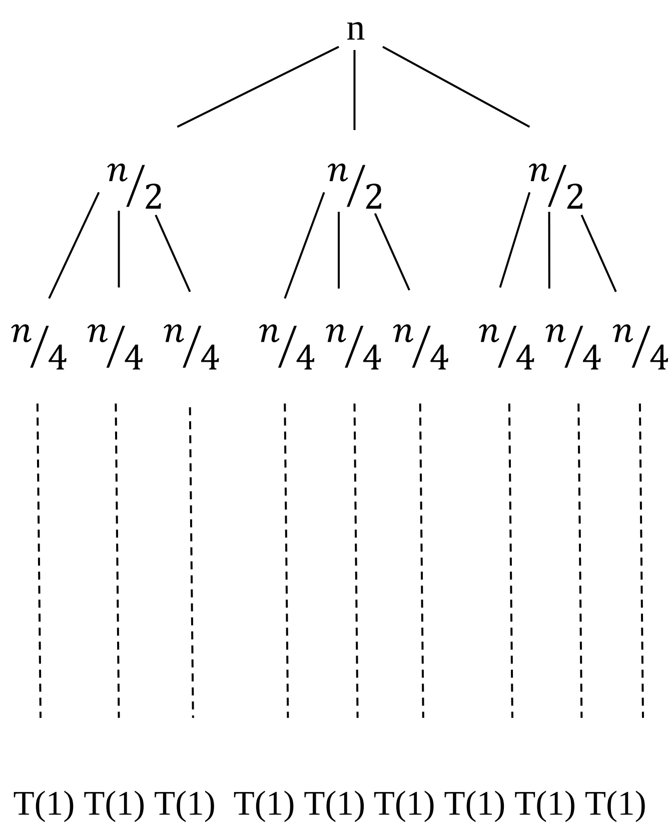
5. Find O using recursion tree

i) $T(n) = 3T\left(\lfloor n/2 \rfloor\right) + n$, verify the guess using substitution method.

ii) $T(n) = 4T\left(\lfloor n/4 \rfloor\right) + cn$

<u>Ans:-</u>

i) We have, $T(n) = 3T\left(n/2\right) + n$



Let, total number of levels $= h + 1$

Number of nodes at each level $= 3^i, \;\; 0 \le i \le h$

Cost of each node at each level $= \frac{n}{2^i}\,, \;\; 0 \le i \le h$

Given, the cost of each node at last level $= T(1)$

So,

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

$$\Longrightarrow 2^h = n$$

$$\Longrightarrow h = \log_2 n$$

So,

T(n)=Sum of all the nodes upto (**h-1**) level + Sum of all the nodes at **h** level

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

$$= n \sum_{i=0}^{h-1} \left(\frac{3}{2}\right)^i + 3^{\log_2 n} \; T(n)$$

Since the first term of the above equation is an increasing G.P. Series, so, the time complexity of the recurrence relation will be the term of cost of all the nodes at the last level, which is —

$$T(1) \;\; n^{\log_2 3}$$

$$= \; \theta\left(n^{\log_2 3}\right)$$

----

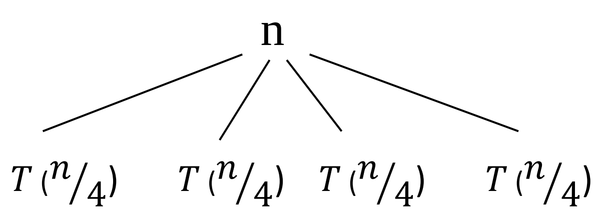
ii) $T(n) = 4T(\lfloor n/4 \rfloor) + cn$

Sol:

We have,

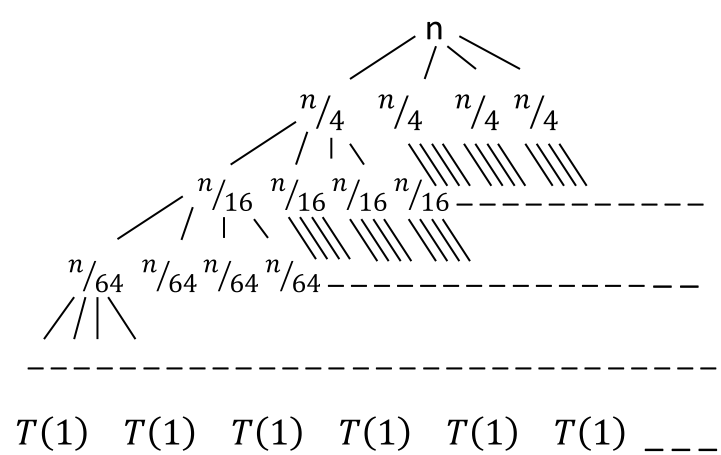
$T(n) = 4T(\lfloor n/4 \rfloor) + cn$

T (n) -



$T(n/4) \quad = \quad 4\,T(n/16) + (n/4)$

Levels	#nodes
0	1
1	4
2	4^2
3	4^3
i	4^i
h	4^h



Let, total number of levels = h+1

Number of nodes at each level = 4^i , Where, $0 \leq i \leq h$

Cost of each nodes at each level = 4^i, Where, $0 \leq i \leq h$

The cost of each node at the last level = $T(1)$

So, $n/4^h = 1$

- ⇨ $4^h = n$
- ⇨ $\log_4 4^h = \log_4 \mathrm{n}$
- ⇨ $\mathrm{h} = \log_4 n$

So, T(n) = Sum of cost of all the nodes upto (h+1) level + Sum of cost of all the levels at 'h' level .

= h.n + $4^h\ T(1)$

= $\log_4 n$. n + $4^{\log_4 n}$. c

= θ($n^{\log_4 n}$)