Q 1) find T(2)

$T(n) = 3T(n-1) + 12n$

$T(0) = 5$

if $n=1$

$T(1) = 3T(1-1) + 12 \times 1$

$= 3T(0) + 12$

$\because T(0) = 5$

$\therefore \Rightarrow 3 \times 5 + 12$

$= 15 + 12$

$T(1) = 27$

$T(2) = 3T(2-1) + 12 \times 2$

$= 3T(1) + 24$

$\because T(1) = 27$

$\therefore 3 \times 27 + 24$

$\Rightarrow 81 + 24$

$\Rightarrow 105$

$T(2) = 105$ Ans

---

2) Substitution method

(a) $T(n) = T(n-1) + c$ —— (1)

if $n = n-1$

then $T(n-1) = T((n-1)-1) + c$

$T(n-1) = T(n-2) + c$ —— (2)

again,

if $n = n-2$

$T(n-2) = T((n-2)-1) + c$

$T(n-2) = T(n-3) + c$ —— (3)

if eq (2) substitution in eq (1)

$T(n) = T(n-1) + c$

$T(n) = [T(n-2) + c] + c$

$T(n) = T(n-2) + 2c$ ——④

Now put eq ③ in eq ④

$T(n) = [T(n-3) + c] + 2c$

$= [T(n-3) + 3c]$

$T(n) = T(n-3) + 3c$

if we do this for k th terms
then we find Pattern

$$\boxed{T(n) = T(n-k) + k.c}$$

Use Base Condition

if base condition $T(0) = 1$

then $T(n-k) = 1$

$n - k = 1$

$\boxed{n = k}$

$T(n) = T(n-n) + n.c$

$= T(0) + n.c$

$= 1 + n.c$

$= nc$

$= n$

$T(n) = n$ Ans

$\boxed{\text{Time Complexity } T(n) = O(n)}$ Ans (b.) $T(n) = 2T(n/2) + n$ —— (1)

if $n = n/2$

then $T(\frac{n}{2}) = 2T(\frac{n}{2} \times \frac{1}{2}) + \frac{n}{2}$

$\Rightarrow 2T\left(\frac{n}{2^2}\right) + \frac{n}{2}$ —— (2)

again put $n = \frac{n}{2^2}$

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

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

put eq (2) in eq (1)

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

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

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

put eq (3) in eq (4)

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

$\Rightarrow 8T\left(\frac{n}{2^3}\right) + \frac{\cancel{4}n}{\cancel{2^2}} + 2n \Rightarrow 2^3\left(\frac{n}{2^3}\right) + 3n$

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

if we do this ~~for~~ k^{th} term

find a pattern

$$\boxed{T(n) = 2^k T\left(\frac{n}{2^k}\right) + kn}$$

if Base Case is give

$T(0) = 1$

then

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

Using log both side

$\log n = \log 2^k \quad (\because \log n^m = m \log n)$

$\log_2 n = k \log_2 2 \quad (\because \log_n n = 1)$

$\log n = k \times 1$

$\boxed{\log n = k}$

So $T(n) = 2^k T\left(\frac{n}{2^k}\right) + kn$

$\boxed{\because n = 2^k} \Rightarrow n\, T\left(\frac{n}{n}\right) + \log n * n$

$\because T(0) = 1 \quad \Rightarrow n \times T(0) + n \log n$

$= n + n \log n$

$\Rightarrow n \log n$ Ans

$$\boxed{T(n) = O(n \log n)}$$

(a) $T(n) = 2T(n/2) + c$ — (1)

if $n = n/2$

then $T(n) = 2T\left(\frac{n}{2} * \frac{1}{2}\right) + c$

$T(n/2) \Rightarrow 2T\left(\frac{n}{2^2}\right) + c$ — (2)

if put $n = \frac{n}{2^2}$ in eq (1)

$T\left(\frac{n}{2^2}\right) = 2T\left(\frac{n}{2^2} \cdot \frac{1}{2}\right) + c$

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

Now put eq (2) in eq (1)

eq (1) $\Rightarrow$ $T(n) = 2T(n/2) + c$

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

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

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

again eq (3) put in eq (4)

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

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

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

if do this for k times the pattern is $T(n) = 2^k \cdot T\left(\frac{n}{2^k}\right) + (k+1)c$ — (5)

if the basecase $T(1) = 1$

then $T\left(\frac{n}{2^k}\right) = 1$

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

$$\boxed{n = 2^k}$$

taking log both side

$$\log n = \log 2^k$$

$$\log n = k \log_2 2 \quad (\because \log_b b = 1)$$

$$\boxed{\log n = k}$$

Put k & $n = 2^k$ in eq (5)

$$T(n) = n \cdot T\left(\frac{n}{n}\right) + (\log n + 1)c$$

$$= n \cdot T(1) + (\log n + 1)c$$

$\Rightarrow n + c \log n + 1$

$\Rightarrow$ So take big one

$\Rightarrow c \log n$

$\Rightarrow c =$ constant

$\Rightarrow \log n$

$$\boxed{T(n) = O(\log n)}$$ Ans

(d) $T(n) = T(n/2) + c$ — ① —— 1^{st} term

Now put $n = n/2$ in eq ①

$$T(n/2) = T\left(\frac{n}{2^2}\right) + c \quad -② \qquad — 2^{nd} \text{ term}$$

Now again put $n = \frac{n}{2^2}$ in eq ①

$$T\left(\frac{n}{2^2}\right) = T\left(\frac{n}{2^3}\right) + c \quad -③ \qquad — 3^{rd} \text{ term}$$

Now put eq ② in eq ①

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

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

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

Now put eq ③ in eq ④

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

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

if do for kth term the Pattern is

$$T(n) = T\left(\frac{n}{2^k}\right) + kc \qquad —— k^{th} \text{ Term}$$

If Base Case ~~T(2) = 1~~

$T(1) = 1$

so $\frac{n}{2^k} = 1$

$$\boxed{n = 2^k}$$

put log both side

$\log n = \log 2^k$

$$\boxed{\log n = k}$$

$$\therefore T(n) = T\left(\frac{n}{2^k}\right) + kc$$

$$\therefore T(n) = T\left(\frac{n}{n}\right) + \log n \, c$$

$= T(1) + \log n \, c$

$= 1 + c \log n$

$\Rightarrow c \log n$

$$\boxed{T(n) = O(\log n) \text{ Ans}}$$

③ · Tree Approach

⑨ $T(n) = 2T(n-1) + 1$

Let Basecase $T(0) = 1$ when $n = 0$

$\Rightarrow$ 2T $\Rightarrow$ means each recursive Step involves making $\overset{2T}{\boxed{\text{two recursive}}}$ calls so do like

$T(n)$

$T(n-1)$ $T(n-1)$

$\boxed{\text{Levels}}$ $\boxed{\text{Tree}}$

L-1 $T(n)$ ← each step has a $O(1)$ work

L-2 $T(n-1)$ $T(n-1)$

L-3 $T(n-2)$ $T(n-2)$ $T(n-2)$ $T(n-2)$

L.... - - - - - -

$T(1)$ $T(1)$

$\boxed{Ln}$ $T(0)$ → Base case hit

Analyze Tree Structure

→ there is n levels → each levels decrease by 1.

→ At each level the Node ($T(n-1)$) is $\boxed{2^k \text{ nodes}}$ k = level number

→ ↳ Node double

Time Complexity

- each level has $O(1)$ work
- n levels so TC is $O(n)$.

$T.C = O(n) + O(1)$

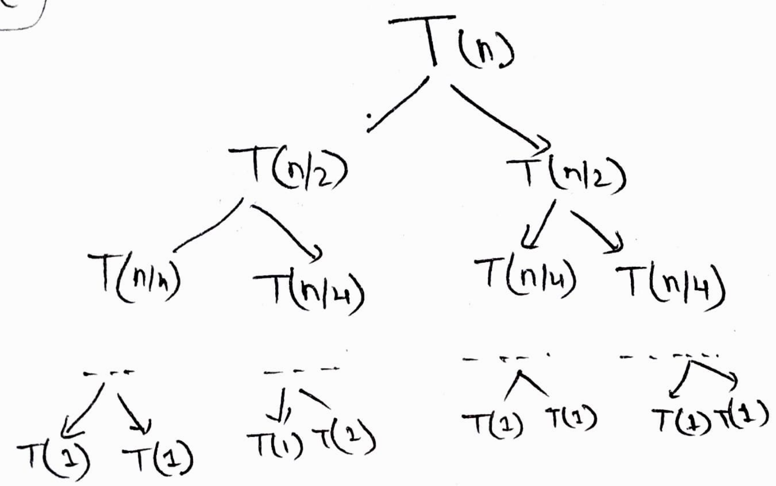
$= O(n)$ Ans

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

Ans Let Base Case $T(1) = 1$, $n = 1$

Recursive Steps $(n>1)$ Make 2 recursive call of $n/2$ and add n

Tree



Analyze tree :

we find :

- Levels of tree is $\log n$ and each level having value of n
- Each level No of Nodes $= 2^k$ $k = $ level.
- & Value of n decrease by half with each level.

Time Complexity :

- Each level work has $O(n)$.
- $\log n$ levels so. Total T.C is $O(n \log n)$

$\boxed{T.C \Rightarrow O(n.\log n)}$ Ans