2204042

1) Algorithm

1) Recursive relation: A problem is described by a mathematical relation, where a function is called again and again.

1) Substitution Method:

Q. Solve $T(n) = \begin{cases} 1 & \text{if } n=1 \\ n*T(n-1) & \text{if } n>1 \end{cases}$

Soln:

$T(n) = n*T(n-1)$ ------ (1)

$T(n-1) = (n-1)*T(n-1-1)$

$= (n-1)*T(n-2)$ --- (2)

$T(n-2) = (n-2)*T(n-2-1)$

$= (n-2)*T(n-3)$ --- (3)

From (1), (2) and (3), we get

$T(n) = n*(n-1)*(n-2)*T(n-3)$

$\vdots$

$= n(n-1)(n-2)(n-3) \cdots T(n-(n-1))$

$= n(n-1)(n-2)(n-3) \cdots 3.2.1$

$= n.n(1-\frac{1}{n}).n(1-\frac{2}{n}).n(1-\frac{3}{n}) \cdots n(\frac{3}{n}).n(\frac{2}{n}).n(\frac{1}{n})$

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

$\therefore T(n) = O(n^n)$.

Q. Solve $T(n) = \begin{cases} 1 & \text{if } n=1 \\ 2T\left(\frac{n}{2}\right)+n & ; n>1 \end{cases}$

Solⁿ:

$$T(n) = 2T\left(\frac{n}{2}\right)+n \quad \text{--- (1)}$$

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

$$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right)+\frac{n}{4} \quad \text{--- (3)}$$

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

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

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

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

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

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

$$\vdots$$

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

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

$\Rightarrow n = 2^k$

$\Rightarrow \log_2 n = \log_2 2^k$

$\Rightarrow k = \log_2 n$

$\therefore T(n) = n\,T\left(\frac{n}{n}\right) + n\log_2 n$

$= n \cdot 1 + n\log_2 n$

$= n + n\log_2 n$

$\therefore T(n) = O(n\log n)$.

Q. Solve $T(n) = \begin{cases} 1 & ; n = 1 \\ T(n-1) + \log n & ; n > 1 \end{cases}$

Soln:

$T(n) = T(n-1) + \log n$ --- (1)

$T(n-1) = T(n-2) + \log(n-1)$ --- (2)

$T(n-2) = T(n-3) + \log(n-2)$ --- (3)

From (1) and (2),

$T(n) = T(n-2) + \log(n-1) + \log n$

$= T(n-3) + \log(n-2) + \log(n-1) + \log n$

$\vdots$

$= T(n-k) + \log(n-(k-1)) + \log(n-(k-2)) + \dots + \log n$

CamScanner

Now, $n-k=1$

$\Rightarrow k = n-1$

$\approx n$

$\therefore T(n) = T(1) + \log 1 + \log 2 + \cdots + \log n$

$= 1 + \log(1.2.3.\cdots n)$

$= 1 + \log(n!)$

$= 1 + \log n^n \quad [n! \approx n^n]$

$= 1 + n\log n$

$\therefore T(n) = O(n\log n)$.

Q. Solve

$$T(n) = \begin{cases} 1, & \text{if } n=1 \\ T\left(\frac{n}{2}\right)+c \end{cases}$$

(Binary search equation)

Soln:

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

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

$T\left(\frac{n}{4}\right) = T\left(\frac{n}{8}\right) + c$ ---(3)

Substituting (2) in (1),

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

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

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

$\vdots$

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

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

$\Rightarrow n = 2^k$

$\Rightarrow \log_2 n = \log_2 2^k$

$\Rightarrow k = \log_2 n$

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

$= 1 + c(\log_2 n)$

$\therefore T(n) = O(\log_2 n)$.

1) Master Method-

$$T(n) = aT\left(\frac{n}{b}\right) + \Theta(n^k \log^p n)$$

$a \geq 1, b \geq 1$, $k \geq 0$ and p is a real number.

(1) If $a > b^k$, then $T(n) = \Theta(n^{\log_b a})$

(2) If $a = b^k$,

(a) $p > -1$, $T(n) = \Theta(n^{\log_b a} . \log^{p+1} n)$

(b) $p = -1$, $T(n) = \Theta(n^{\log_b a} . \log \log n)$

(c) $p < -1$, $T(n) = \Theta(n^{\log_b a})$

(3) If $a < b^k$,

(a) $p \geq 0$, then $T(n) = \Theta(n^k \log^p n)$

(b) $p < 0$, then $T(n) = \Theta(n^k)$

1) <u>Ex:</u>

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

$a = 4, b = 2, k = 1, p = 0$

Here, $a > b^k$

$\Rightarrow 4 > 2^1$

$\therefore T(n) = \Theta(n^{\log_2 4}) = \Theta(n^2)$.

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

$a = 1,\ b = 2,\ k = 0,\ p = 0$

Here, $a = b^k$

$\Rightarrow 1 = 2^0$

and $p > -1$

$\therefore T(n) = \Theta(n^{\log_2 1} \cdot \log n)$

$= \Theta(\log n)$

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

$a = 2,\ b = 2,\ k = 1,\ p = -1$

Here, $a = b^k$

$\Rightarrow 2 = 2^1$

and $p = -1$

$\therefore T(n) = \Theta(n^{\log_2 2} \cdot \log \log n)$

$= \Theta(n \log \log n)$

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

$a = 2;\ b = 2;\ k = 1;\ p = -2$

Here, $a = b^k$

$\Rightarrow 2 = 2^1$

and $p < -1$

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

$= \Theta(n^{\log_2 2})$

$= \Theta(n)$

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

$a = 2,\ b = 2,\ k = 2,\ p = 0$

Here, $a < b^k$

$\Rightarrow 2 < 2^2$

and $p \geq 0$

$\therefore T(n) = \Theta(n^2 \log^0 n)$

$= \Theta(n^2)$

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

$a=2,\ b=2,\ k=2,\ p=-1$

Here, $a < b^k$

$\Rightarrow 2 < 2^2$

and $p<0$

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

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

$a \geq 1,\ b > 1$

$T(n) = n^{\log_b a}[U(n)]$

$U(n)$ depends on $h(n)$

$$h(n) = \frac{f(n)}{n^{\log_b a}}$$

If $h(n)$	$U(n)$
$n^r,\ r>0$	$O(n^r)$
$n^r,\ r<0$	$O(1)$
$(\log n)^i,\ i \geq 0$	$\frac{(\log_2 n)^{i+1}}{i+1}$

Ex: (1) $T(n) = T(n-1) + 1$

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

hence $b \not> 1$

Not soluable by Master method.

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

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

$T(n) = n^{\log_b a}[U(n)]$

$= n^{\log_2 8}[U(n)]$

$= n^3 U(n)$

$h(n) = \frac{f(n)}{n^{\log_b a}} = \frac{n^2}{n^3} = \frac{1}{n} = n^{-1}$

$\therefore U(n) = O(1)$

Hence, $T(n) = n^3 \cdot O(1)$

$\approx O(n^3)$.

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

Here, $a = 1, b = 2, f(n) = c$

$T(n) = n^{\log_b a}[U(n)]$

$= n^{\log_2 1}[U(n)] = U(n)$

$$h(n) = \frac{f(n)}{n^{\log_b a}} = \frac{c}{n^0} = c = (\log_2 n)^0 \cdot c$$

$$U(n) = \frac{(\log_2 n)^{0+1}}{0+1}$$

$$= \log_2 n$$

$\therefore T(n) = O(\log_2 n).$

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

a can't be function of n.

(2) $T(n) = 64T\left(\frac{n}{2}\right) - n^2$

f(n) can't be negative.

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

f(n) can't be sin, cos etc..

(4) $T(n) = 2T\left(\frac{n}{2}\right) + \frac{1}{n}$

k must be greater or equal zero $(k \geq 0)$

(5) $T(n) = 0.5T\left(\frac{n}{2}\right) + n$

a must be greater or equal one $(a \geq 1)$.

Not solvable by Master method.

(6) $T(n) = \sqrt{2}\, T\left(\frac{n}{2}\right) + \log n$

Here, $a = \sqrt{2}, b = 2, k = 0, p = 1$

$a > b^k$

$\Rightarrow \sqrt{2} > 2^0$

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

$= \Theta(n^{\log_2 \sqrt{2}})$

$= \Theta(n^{1/2})$

$= \Theta(\sqrt{n})$.

1Q. Solve $T(n) = \begin{cases} T(\sqrt{n}) + \log n & \text{if } n \geq 2 \\ O(1) & \text{else} \end{cases}$

<u>Solⁿ:</u> Let, $n = 2^m$

$\Rightarrow \log n = m \log_2 2 = m$

$\therefore T(2^m) = T(2^{m/2}) + m$

Let, $T(2^m) = S(m)$

$\therefore S(m) = S\left(\frac{m}{2}\right) + m$

$T(n) = a\, T\left(\frac{n}{b}\right) + n^k \log^p n$

Here, $a = 1$, $b = 2$, $k = 1$, $p = 0$

$a < b^k$

$\Rightarrow 1 < 2^1$

and $p = 0$

$\therefore S(m) = O(m^1 \log^0 m)$

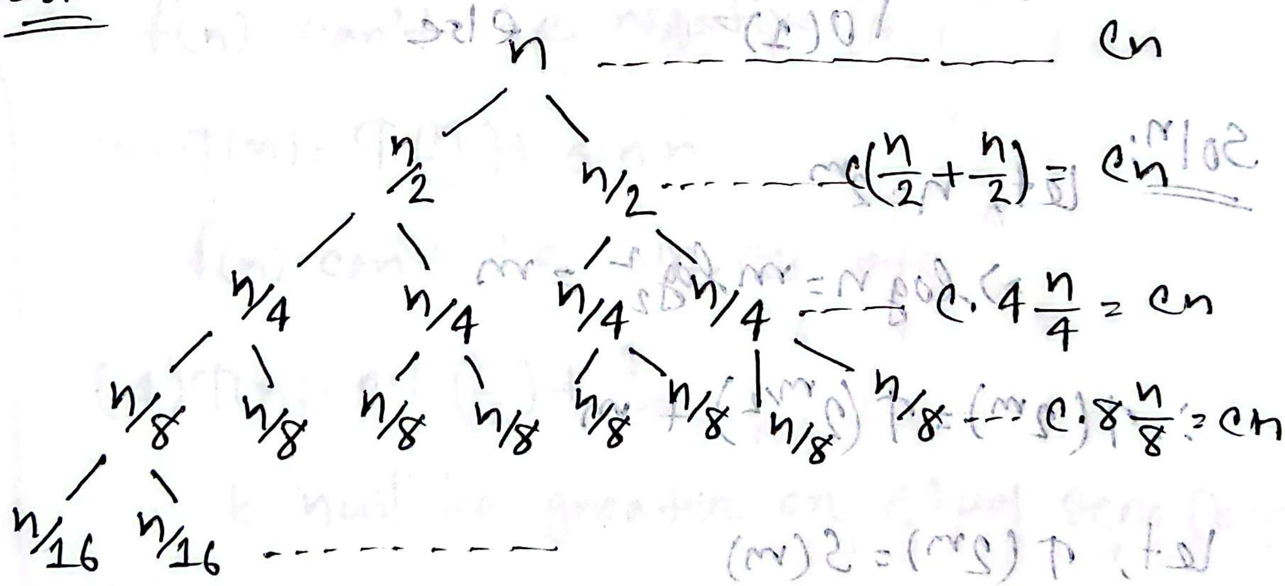
$= O(m)$

$\therefore T(n) = O(\log n)$.

II) Recursive Tree Method:

Q. Solve $T(n) = 2T\left(\frac{n}{2}\right) + cn$

Soln:

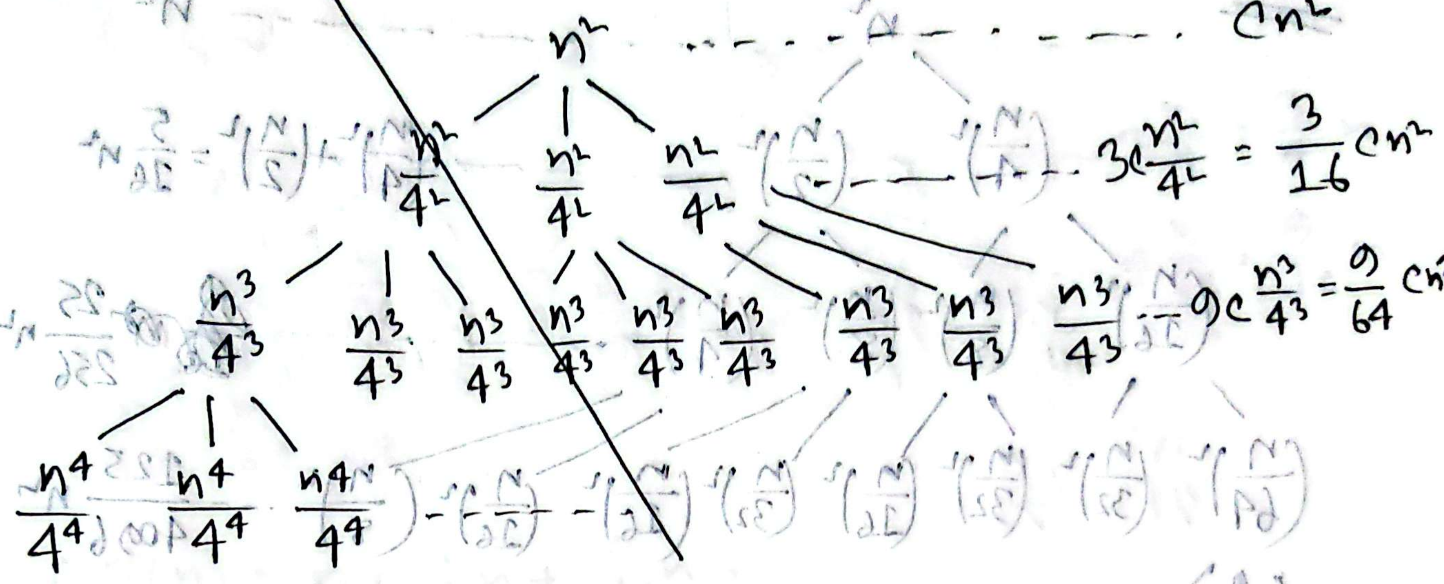


Total cost depends on the height

Height = $\log n$

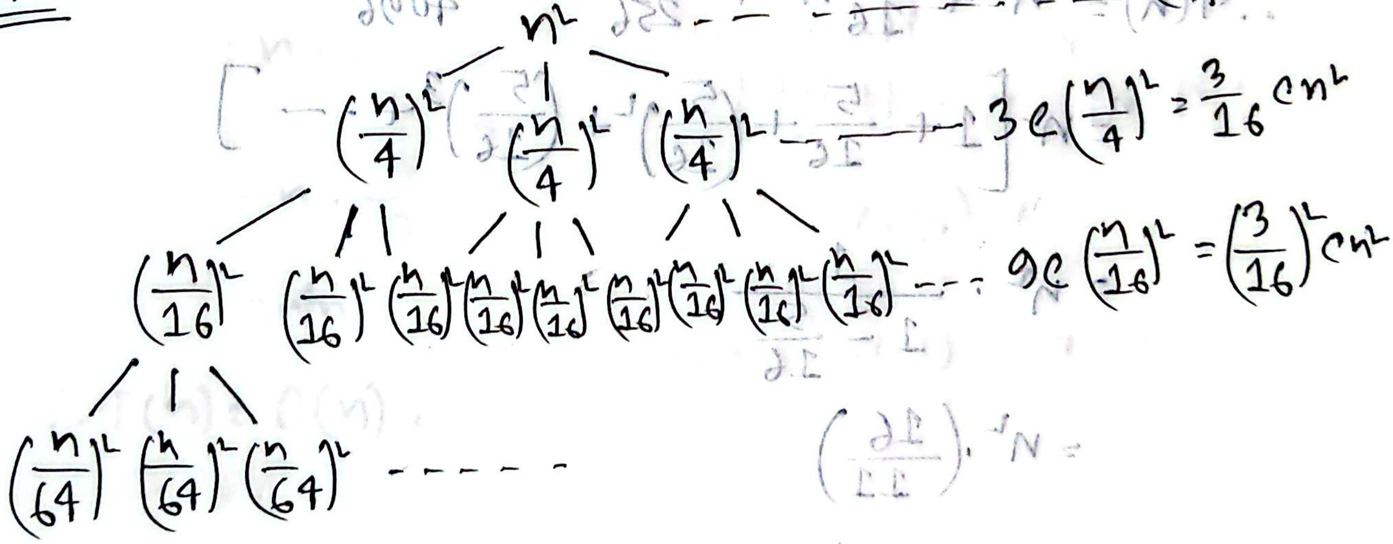
$\therefore T(n) = cn * (\text{Height}) = cn \log n$ $\therefore T(n) = O(n \log n)$.

Q. Solve $T(n) = 3T(\lfloor n/4 \rfloor) + cn^2$



Q. Solve $T(n) = 3T(\lfloor n/4 \rfloor) + cn^2$

Sol^n:



$$\therefore T(n) = cn^2 + \frac{3}{16}cn^2 + \left(\frac{3}{16}\right)^2 cn^2 + \cdots$$

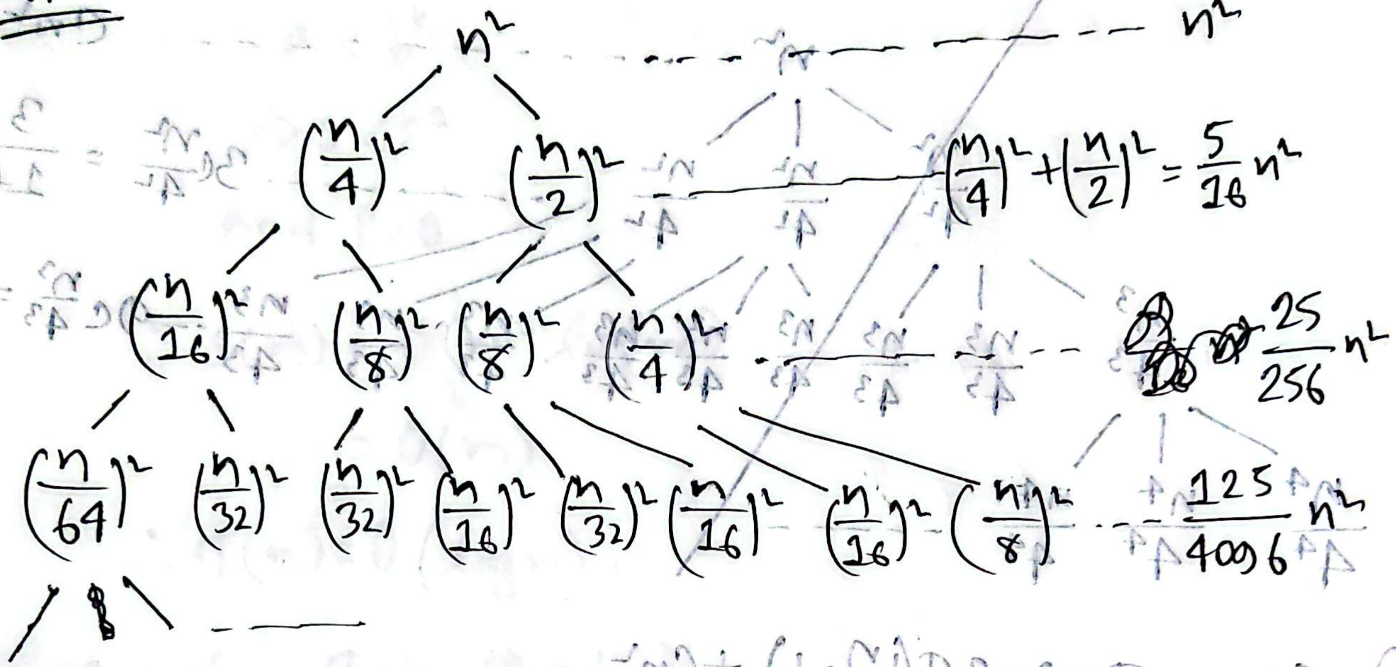
$$= cn^2\left[1 + \frac{3}{16} + \left(\frac{3}{16}\right)^2 + \left(\frac{3}{16}\right)^3 + \cdots\right]$$

$$= cn^2 \cdot \frac{1}{1 - \frac{3}{16}} = cn^2\left(\frac{16}{13}\right)$$

$$\therefore T(n) = O(n^2).$$

Q. Solve $T(n) = T(n/4) + T(n/2) + n^2$

Sol^n:



$$\therefore T(n) = n^2 + \frac{5}{16}n^2 + \frac{25}{256}n^2 + \frac{125}{4096}n^2 + \cdots$$

$$= n^2\left[1 + \frac{5}{16} + \left(\frac{5}{16}\right)^2 + \left(\frac{5}{16}\right)^3 + \cdots\right]$$

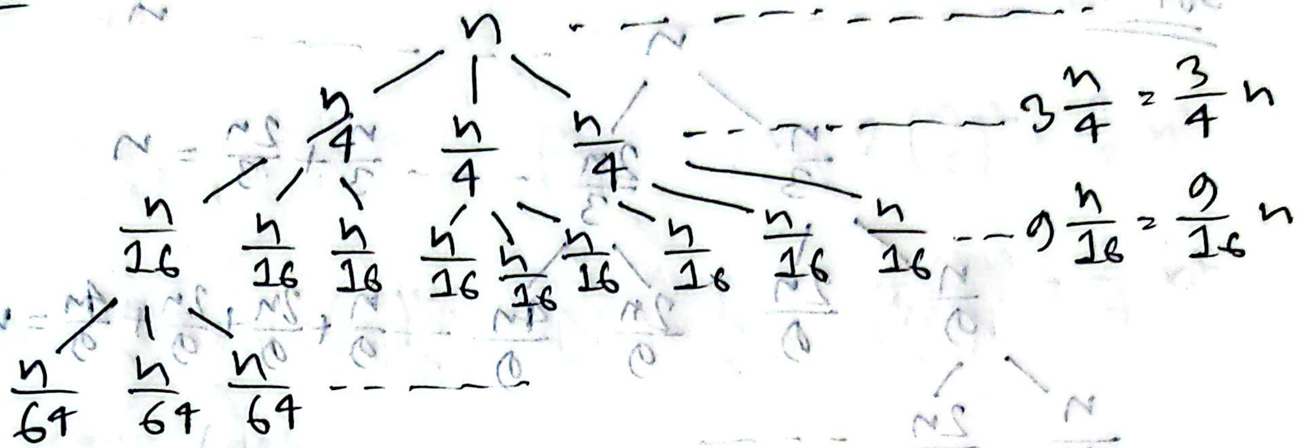
$$= n^2 \cdot \frac{1}{1 - \frac{5}{16}}$$

$$= n^2 \cdot \left(\frac{16}{11}\right)$$

$$\therefore T(n) = O(n^2).$$

Q. Solve $T(n) = 3T\left(\frac{n}{4}\right) + n$

Sol^n:



$$\therefore T(n) = n + \frac{3}{4}n + \frac{9}{16}n + \cdots$$

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

$$= n \cdot \frac{1}{1 - \frac{3}{4}}$$

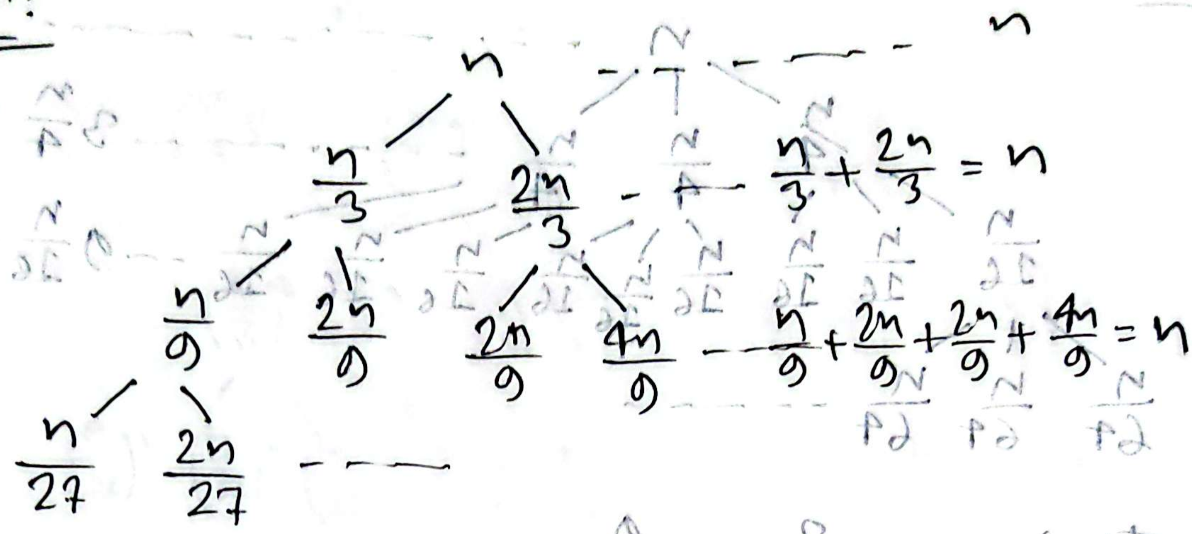
$$= n \cdot \frac{4}{1}$$

$$= 4n$$

$$\therefore T(n) = O(n).$$

Q. Solve $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$

Soln:



Total cost depends on Height

$\text{Height} = \log_{3/2} n \approx \log n$

$\therefore T(n) = n * \text{Height}$

$= n \log n$

$\therefore T(n) = O(n \log n).$

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

<u>Sol</u>^{n}:

$$n^2 \quad \text{------------} \quad n^2$$

$$\left(\frac{n}{2}\right)^2 \quad \left(\frac{n}{2}\right)^2 \quad \text{---} \quad 2\left(\frac{n}{2}\right)^2 = \frac{1}{2}n^2$$

$$\left(\frac{n}{4}\right)^2 \quad \left(\frac{n}{4}\right)^2 \quad \left(\frac{n}{4}\right)^2 \quad \left(\frac{n}{4}\right)^2 \quad \text{---} \quad 4\left(\frac{n}{4}\right)^2 = \frac{1}{4}n^2$$

$$\left(\frac{n}{8}\right)^2 \quad \left(\frac{n}{8}\right)^2 \quad \text{----}$$

$$\therefore T(n) = n^2 + \frac{1}{2}n^2 + \frac{1}{4}n^2 + \text{---}$$

$$= n^2\left[1 + \frac{1}{2} + \frac{1}{4} + \text{---}\right]$$

$$= n^2 \cdot \frac{1}{1-\frac{1}{2}}$$

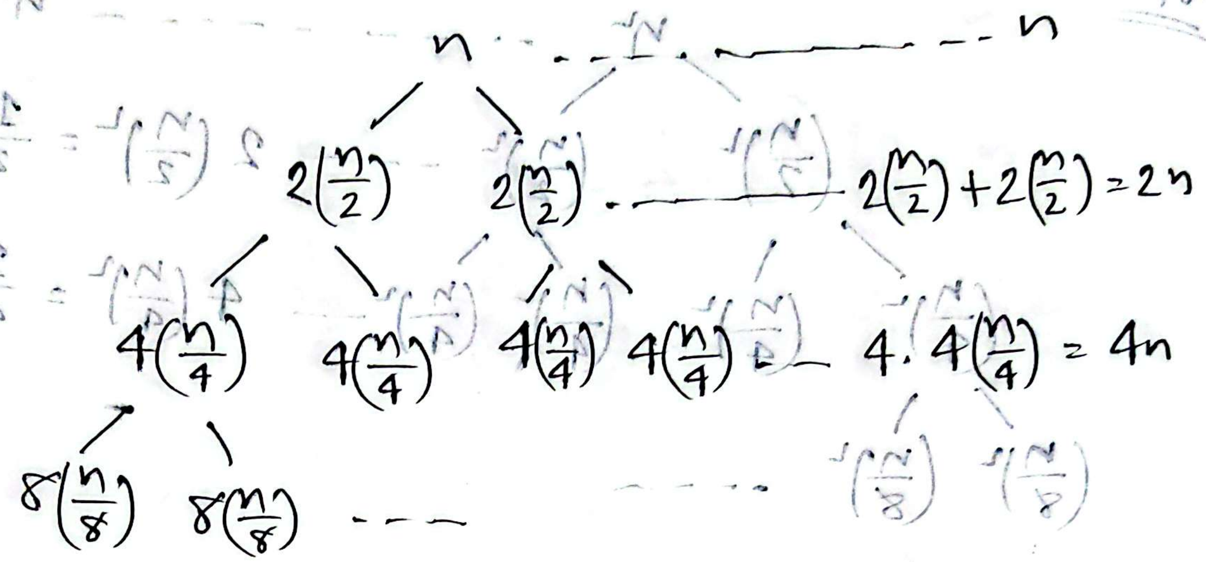
$$= n^2 \cdot \frac{2}{1}$$

$$= 2n^2$$

$$\therefore T(n) = O(n^2)$$

Q. Solve $T(n) = 4T(\lfloor n/2 \rfloor) + n$

Soln:



$$\therefore T(n) = n + 2n + 4n + \cdots + n\log n$$

$$= n[1 + 2 + 4 + \cdots + \log n]$$

$$= n \cdot \frac{1 - 2^{\log n}}{1-2}$$

$$= n \cdot \frac{1-n}{1-2}$$

$$= n \frac{n-1}{2-1}$$

$$= n^2 - n$$

$$\therefore T(n) = O(n^2).$$