ASSIGNMENT 1
RECCURENCE RELATIONS

- $$T(n) = \begin{cases} \Theta(1) & \text{if } n=1 \\ 2T(n/2+17)+n & \text{otherwise} \end{cases}$$

Answer:

$$T(n) = 2T(n/2+17)+n$$

We ignore 17 as it is a constant.

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

From Master's Theorem,

$$T(n) = aT(n/b)+f(n)$$

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

$$\therefore n^{\log_b a} = n^{\log_2 2}$$

$$= n$$

Hence, $f(n) = \Theta(n^{\log_b a})$

$$= \Theta(n)$$

$\therefore$ We can write $T(n) = \Theta(n^{\log_b a} \lg n)$

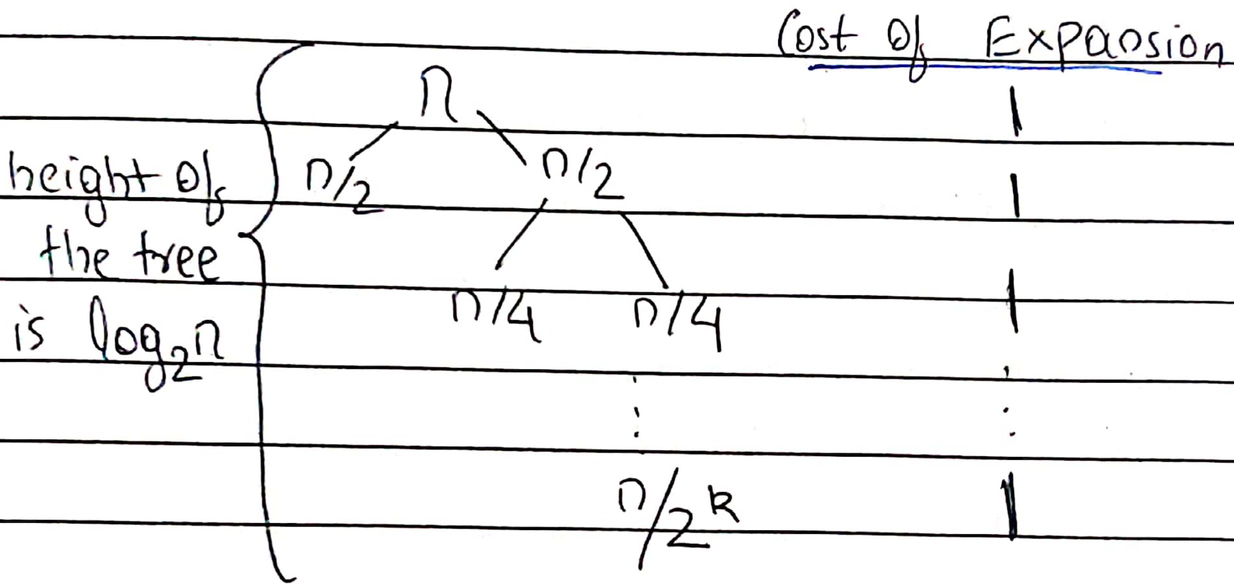
$$= \Theta(n \lg n)$$

Answer: $T(n) = \Theta(n \lg n)$

- $T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/2) + 1 & \text{otherwise} \end{cases}$

Answer:

We use recursion tree approach to solve this problem.



Now, $n/2^k = 1 \Rightarrow n = 2^k$

$\therefore k = \log_2 n$

The total time complexity for expanding the tree is the sum of the costs for expansion at each step.

$$\therefore T(n) = \sum_{i=1}^{k} 1$$

$$= \sum_{i=1}^{\log_2 n} 1 = \log_2 n$$

Answer $T(n) = O(\lg n)$

- $$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 3T(n/2) + \log n & \text{otherwise} \end{cases}$$

Answer:

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

From Master's Theorem,

$$T(n) = aT(n/b) + f(n)$$

$$\therefore a = 3,\ b = 2,\ f(n) = \log n$$

$$\text{Now, } n^{\log_b a} = n^{\log_2 3}$$

$$= n^{1.6}$$

$$\therefore f(n) = O(n^{1.6-\epsilon}) = O(\log n)$$

$$\therefore \text{We can write } T(n) = \Theta(n^{\log_b a})$$

$$= \Theta(n^{1.6})$$

Answer: $T(n) = \Theta(n^{1.6})$

- $$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 4T(n/2) + n^2 & \text{otherwise} \end{cases}$$

Answer:

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

From Master's Theorem,

$$T(n) = aT(n/b) + f(n)$$

$$\therefore a = 4,\ b = 2,\ f(n) = n^2$$

$$\text{Now, } n^{\log_b a} = n^{\log_2 4}$$

$$= n^2$$

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

Hence, we can write $T(n) = \Theta(n^{\log_b a} \lg n)$
$= \Theta(n^2 \lg n)$

Answer $T(n) = \Theta(n^2 \lg n)$

- $T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 8T(n/2) + n^2 & \text{otherwise} \end{cases}$

Answer:

$T(n) = 8T(n/2) + n^2$

From Master's Theorem,

$T(n) = aT(n/b) + f(n)$

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

Now, $n^{\log_b a} = n^{\log_2 8}$
$= n^3$

$\therefore f(n) = n^2 = O(n^{3-\epsilon})$
$= O(n^{\log_2 8 - \epsilon})$

Hence, we can write $T(n) = \Theta(n^{\log_2 8})$
$= \Theta(n^3)$

Answer $T(n) = \Theta(n^3)$

- $T(n) = \begin{cases} \Theta(1) & \text{if } n=1 \\ 4T(n/2) + n^2 \log n & \text{otherwise} \end{cases}$

Answer:

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

From Master's Theorem,

$$T(n) = aT(n/b) + f(n)$$

$$\therefore a = 4,\ b = 2,\ f(n) = n^2 \log n$$

Now, $n^{\log_b a} = n^{\log_2 4}$
$= n^2$

$f(n) = n^2 \log n$ is asymptotically larger than n^2, but it is not polynomially larger, i.e. the ratio $f(n)/n^{\log_b a} = \log n$ is asymptotically less than n^{ϵ}, for any $\epsilon > 0$.

Now, we know

if $f(n) = n^{\log_b a} \log^k n,\ k > 0$

$$T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$$

→ Couldn't find proper proof of the statement.

Hence, we can write $T(n) = \Theta(n^2 \log^2 n)$

Answer: $T(n) = \Theta(n^2 \log^2 n)$

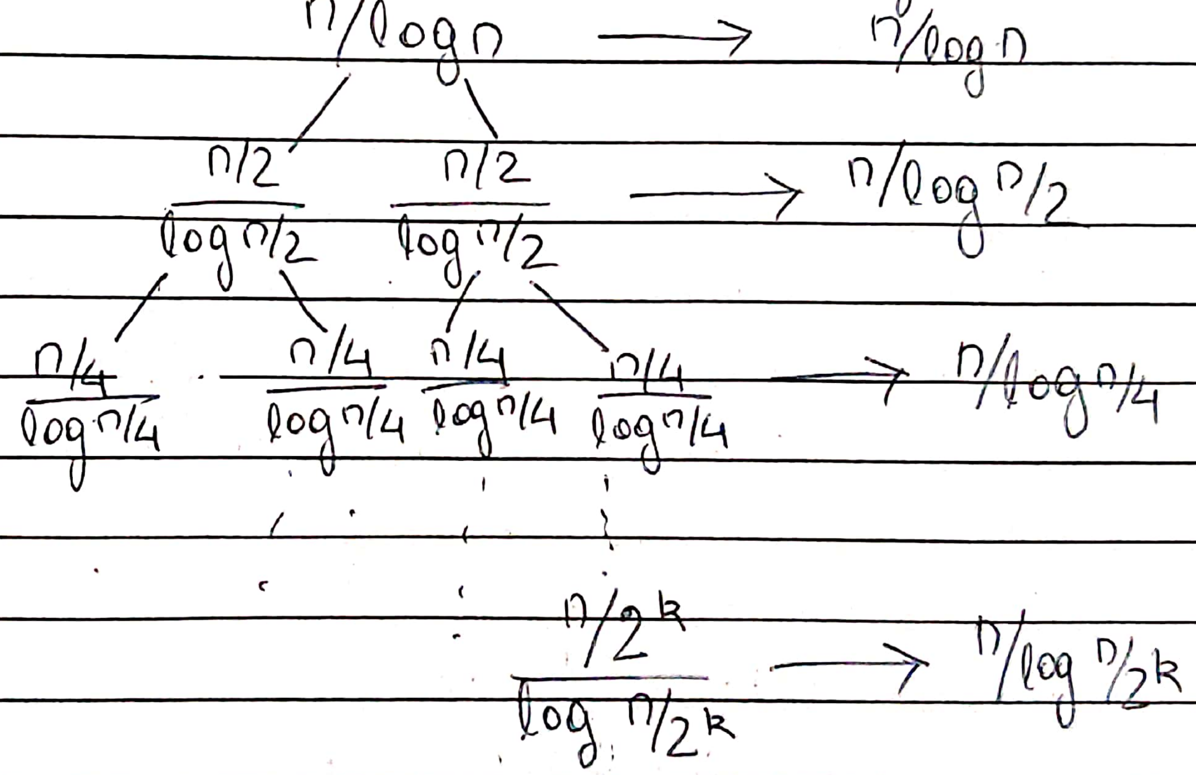
- $T(n) = \begin{cases} \Theta(1) & \text{if } n=1 \\ 2T(n/2) + n/\log n & \text{Otherwise} \end{cases}$

Answer:

We use the recursion tree approach to solve this problem

$$T(n) = 2T(n/2) + \frac{n}{\log n}$$

Cost of Expansion



Now, as we know that the recursion stops at 1, $n/2^k = 1$

$$\Rightarrow n = 2^k$$

$$\therefore k = \log_2 n$$

$$\therefore T(n) = \frac{n}{\log n} + \frac{n}{\log n/2} + \cdots + \frac{n}{\log n/2^k}$$

$$= n\left[\frac{1}{\log n} + \frac{1}{\log n/2} + \cdots + \frac{1}{\log 1}\right]$$

$$\left(\because \underline{n/2^k = 1}\right)$$

$$= n\left(\frac{1}{\log 2^k} + \frac{1}{\log 2^{k-1}} + \cdots + 1\right) + n\Theta(1)$$

$$= n\left(\frac{1}{k} + \frac{1}{k-1} + \cdots + 1\right) + n\Theta(1)$$

$$\leq n\log k + n$$

$$= n\log(\log n) + n$$

$$= O(n\log\log n)$$

Answer: $T(n) = O(n\log\log n)$

- $T(n) = \begin{cases} \Theta(1) & \text{if } n=1 \\ T(n/3) + T(2n/3) + n & \text{otherwise} \end{cases}$

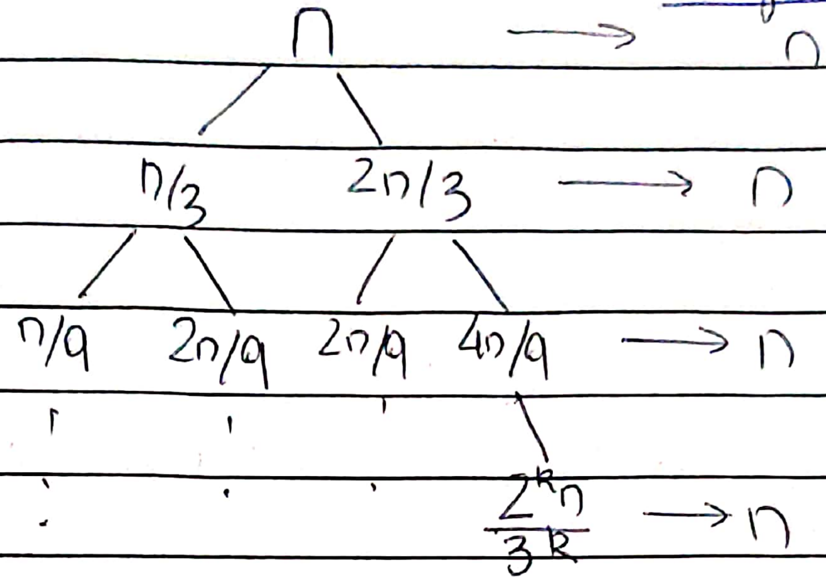
<u>Answer</u>:

We use the recursion tree approach to solve this problem

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

P.T.O.

Cost of Expansion



Now, as we know the recursion stops at 1, $\frac{2^k n}{3^k} = 1 \Rightarrow k = \log_{3/2} n$

$\therefore T(n) = n + n + n + \ldots + k$ times
$= kn$
$= n \log_{3/2} n$
$= O(n \log n)$

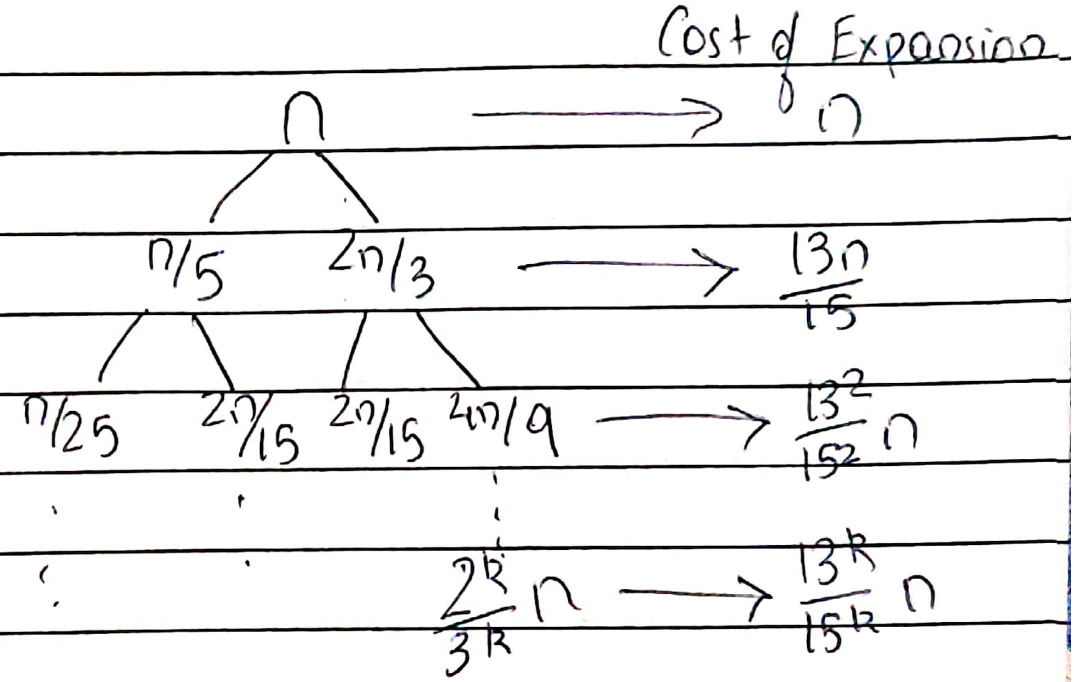
Answer: $T(n) = O(n \log n)$

- $T(n) = \begin{cases} 1 & \text{if } n = 1 \\ T(n/5) + T(2n/3) + n & \text{otherwise} \end{cases}$

Answer:

We use the recursion tree approach to solve the problem.

$$T(n) = T(n/5) + T(2n/3) + n$$



As we know that the recursion stops at 1, $\frac{2^k}{3^k} n = 1 \Rightarrow k = \log_{3/2} n$

$$\therefore T(n) = n + \frac{13n}{15} + \frac{13^2}{15^2} n + \dots + \frac{13^k}{15^k} n$$

$$= n \left(1 + \frac{13}{15} + \frac{13^2}{15^2} + \dots + \frac{13^k}{15^k}\right)$$

$$= n \left(1 \cdot \frac{1 - \left(\frac{13}{15}\right)^{k+1}}{1 - \frac{13}{15}}\right)$$

$$= \frac{15n}{2}\left(1-0.9^{\log_{3/2} n}\right)$$

$$= O(n)$$

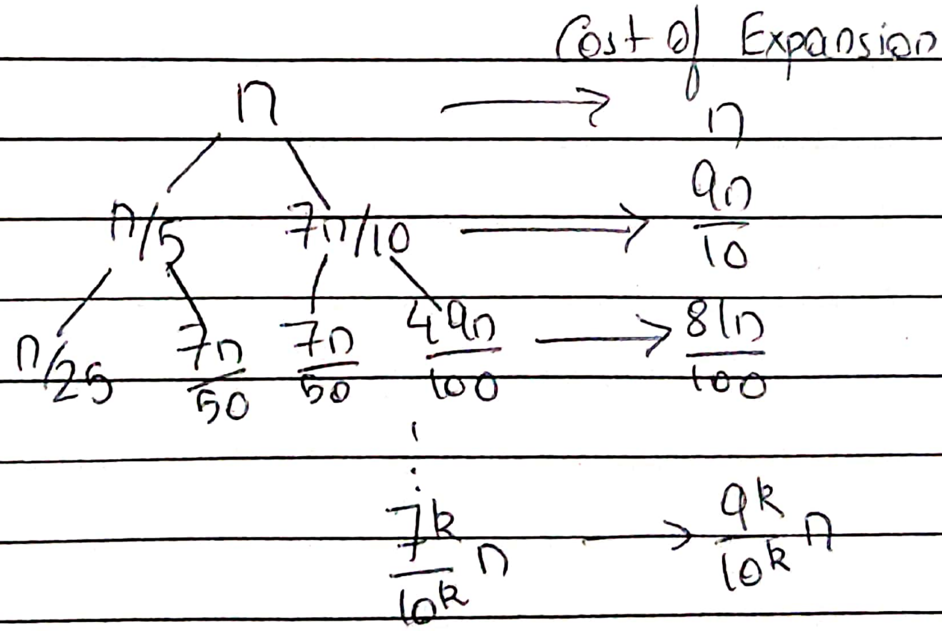
Answer: $T(n) = O(n)$

- $T(n) = \begin{cases} 1, & \text{if } n=1 \\ T(n/5) + T(7n/10) + n, & \text{otherwise} \end{cases}$

Answer:

We use the recursion tree approach to solve the problem.

$$T(n) = T(n/5) + T(7n/10) + n$$



As we know, the recursion stops at 1, $\frac{7^k}{10^k} n = 1 \Rightarrow n = \log_{10/7} n$

$$\therefore T(n) = n + \frac{9n}{10} + \frac{81n}{100} + \dots + \frac{9^k}{10^k}n$$

$$= n\left(1 + \frac{9}{10} + \frac{81}{100} + \dots + \frac{9^k}{10^k}\right)$$

$$= n\left(1 \cdot \frac{1-(9/10)^{k+1}}{1-9/10}\right)$$

$$= 10n\left(1 - 0.9^{\log_{10/7} n}\right)$$

$$\left(\because k = \log_{10/7} n\right)$$

$$= O(n)$$

Answer: $T(n) = O(n)$

- $T(n) = \begin{cases} \sqrt{n}\,T(\sqrt{n}) + n & \text{if } n \geq 1 \\ 1 & \text{if } n = 0 \end{cases}$

Answer:

Let $n = 2^k$

$$T(n) = \sqrt{n}\,T(\sqrt{n}) + n$$

or, $\dfrac{T(n)}{n} = \dfrac{T(\sqrt{n})}{\sqrt{n}} + 1$

or, $\dfrac{T(2^k)}{2^k} = \dfrac{T(2^{k/2})}{2^{k/2}} + 1$

Now, let $A(k) = \dfrac{T(2^k)}{2^k}$

∴ We can write

$$A(k) = A(k/2) + 1$$

We know from Page 2, that the solution of this recurrence relation is $A(k) = \Theta(\log k)$

Now, $n = 2^k$ and $A(k) = \frac{T(2^k)}{2^k}$,

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

or, $$\frac{T(n)}{n} = \Theta(\log\log n)$$

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

Answer: $T(n) = \Theta(n\log\log n)$