$C.1.$ *Recurrence Tree Method* $-$ *Example* 3

Solve the following recurrence equation:

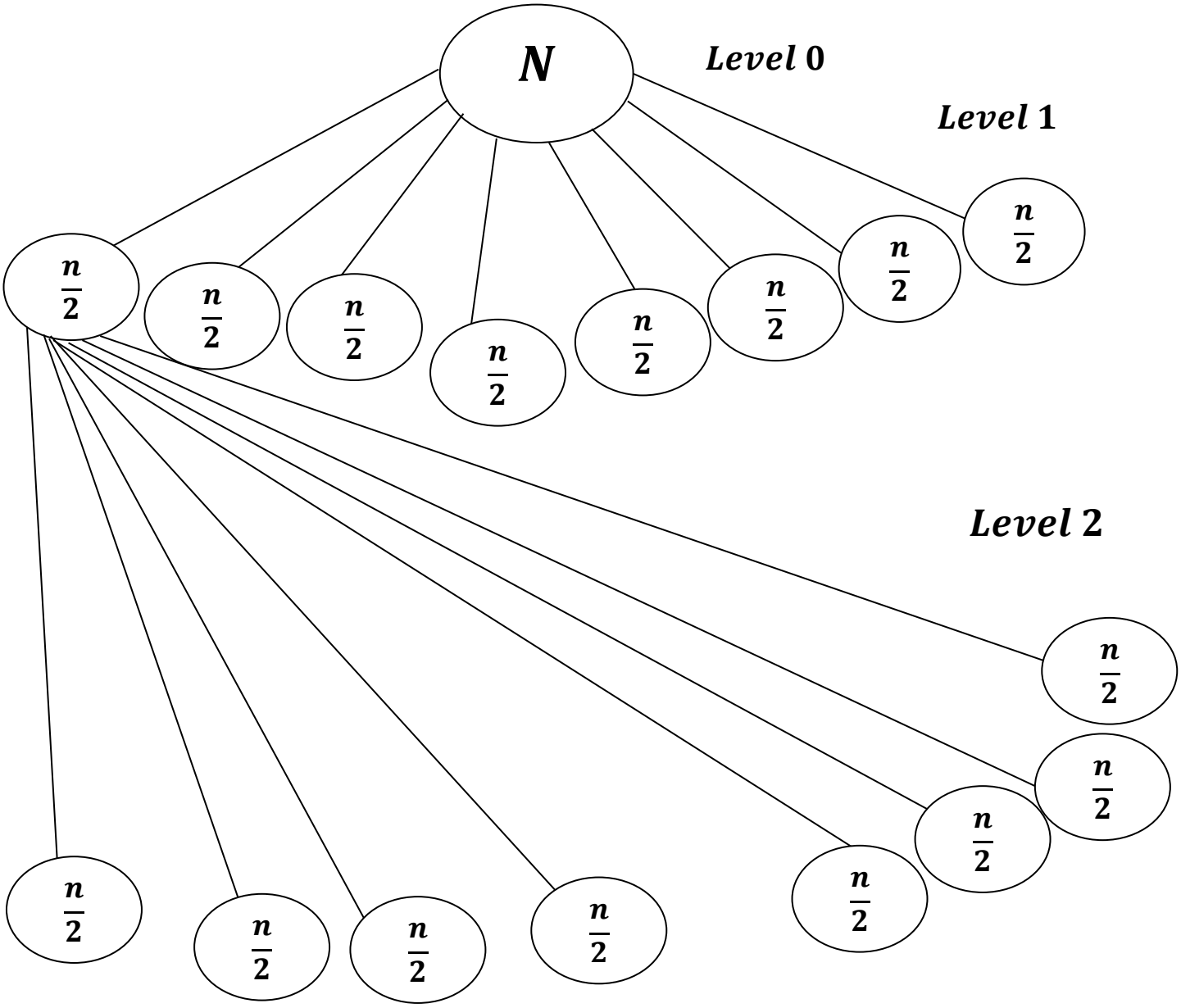
$$t_n = \begin{cases} 1 & for\ n = 1 \\ 8T\left(\frac{n}{2}\right) & for\ n > 1 \end{cases}$$

Solution:

Initially the input is n. *It is sub* $-$ *divided into eight sub* $-$ *problems*. *In the next level, each of these eight sub* $-$ *problems is again divided into eight sub problems*(*so a total of* $8 \times 8 = 64$ *sub problems*).

Here $8T$ *represents* 8 *subdivision and* $\frac{n}{2}$ *represents* $\frac{n}{2}$ *increase of problem size*.

This process is continued till a pattern is obtained.



Recurrence tree at levels 1 and 2 (and only one problem division is shown)

$\boldsymbol{Level}$	$\boldsymbol{No.of\ problems}$	$\boldsymbol{Problem\ Size}$	$\boldsymbol{Work\ done = No.of\ problems \times problem\ size}$
$\mathbf{0}$	$\mathbf{1}$	$\boldsymbol{n}$	$\mathbf{1} \times \boldsymbol{n} = \boldsymbol{n}$
$\mathbf{1}$	$\mathbf{8}$	$\frac{\boldsymbol{n}}{\mathbf{8}}$	$\mathbf{8} \times \frac{\boldsymbol{n}}{\mathbf{8}} = \boldsymbol{n}$
$\mathbf{2}$	$\mathbf{8^2}$	$\frac{\boldsymbol{n}}{\mathbf{8^2}}$	$\mathbf{8^2} \times \frac{\boldsymbol{n}}{\mathbf{8^2}} = \boldsymbol{n}$
.	.	.	.
.	.	.	.
$\boldsymbol{k}$	$\mathbf{8}^{\boldsymbol{k}}$	$\frac{\boldsymbol{n}}{\mathbf{8}^{\boldsymbol{k}}}$	$\mathbf{8}^{\boldsymbol{k}} \times \frac{\boldsymbol{n}}{\mathbf{8}^{\boldsymbol{k}}} = \boldsymbol{n}$
$\mathbf{log_2}\ \boldsymbol{n}$	$\mathbf{8}^{\mathbf{log_2}\ \boldsymbol{n}}$	$\mathbf{1}$	$\mathbf{8}^{\mathbf{log_2}\ \boldsymbol{n}} \times \mathbf{1} = \boldsymbol{n}^{\mathbf{3}}$

$\boldsymbol{The\ problem\ size\ reduces\ to\ 1\ as\ T(n) = 1}$
$\boldsymbol{for\ n = 1\ i.e.\ T(1) = 1\ or\ t_1 = 1.}$

$\boldsymbol{It\ can\ be\ observed\ that\ at\ every\ level\ a\ problem\ is\ divided\ into\ eight\ subproblems\ or\ nodes\ is\ increasing\ in\ the\ following\ pattern}: \mathbf{1, 8, 64, \ldots} \left(\mathbf{8^0, 8^1, 8^2, \ldots., 8^i}\right).$

The problem size is decreasing in a geometric series as follows: $\left(n, \frac{n}{8}, \frac{n}{8^2}, \ldots, \frac{n}{8^k}, \ldots, 1\right)$. ***Based on the table, the amount of work done at the*** $\log_2 n$ ***can be calculated as follows:***

$8^{\log_2 n} \times T(1)$

$= (2^3)^{\log_2 n} \times 1$

$= \left(2^{\log_2 n}\right)^3$

$= (n)^3 \ \left[a^{\log_a b} = b\right]$

$= n^3$

Alternatively,

$8^{\log_2 n} \times T(1)$

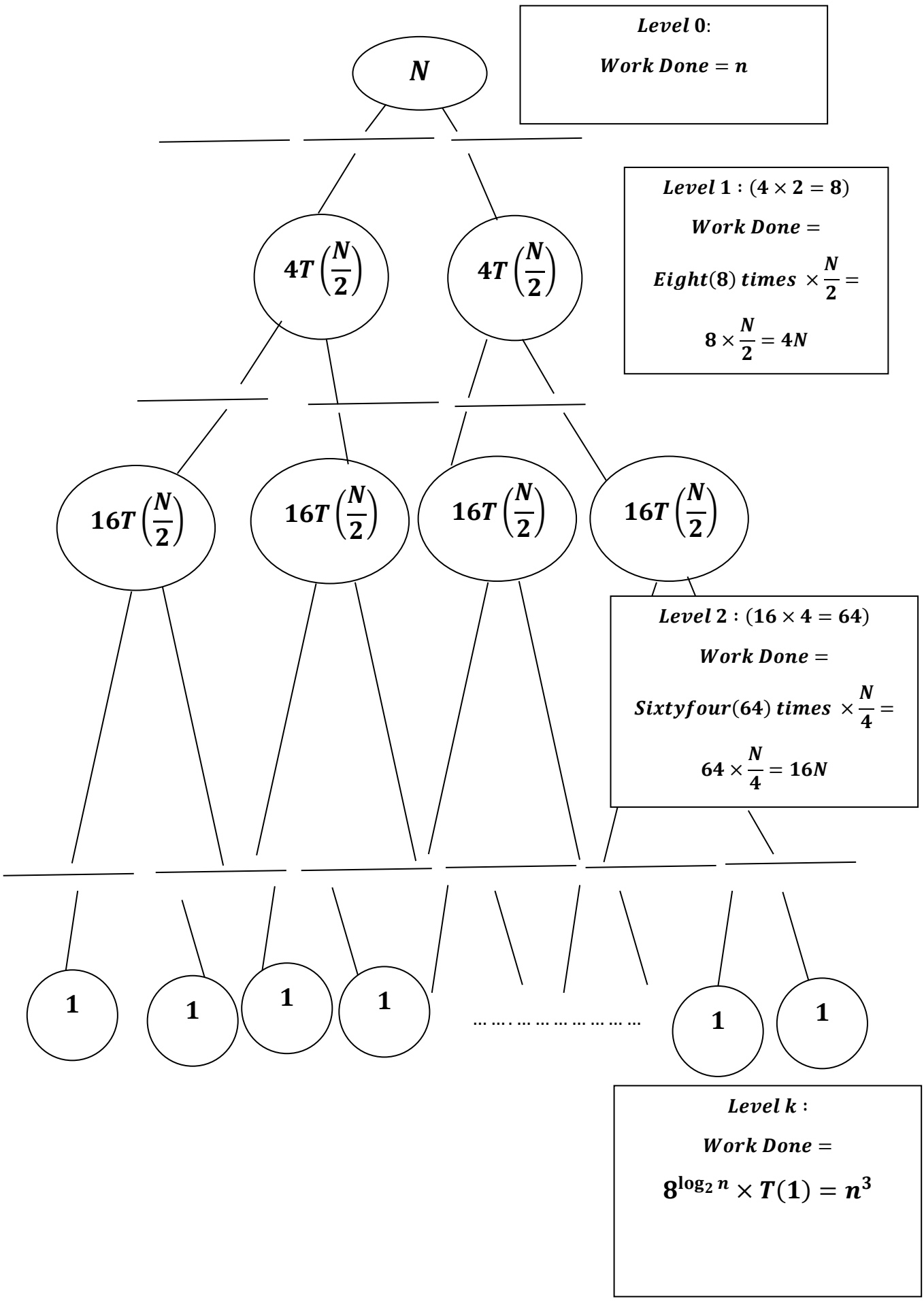
$= (2^3)^{\log_2 n} \times 1$

$= \left(2^{\log_2 n}\right)^3$

$= \left(n^{\log_2 2}\right)^3 \ \left[a^{\log_a b} = b^{\log_a a} = b\right]$

$= n^3$

Now to calculate amount of work done at other level lets divide the above again:


Level 0:
Work Done = n
N
Level 1 : (4 × 2 = 8)
Work Done =
Eight(8) times × N/2 =
8 × N/2 = 4N
4T(N/2)
4T(N/2)
Level 2 : (16 × 4 = 64)
Work Done =
Sixtyfour(64) times × N/4 =
64 × N/4 = 16N
16T(N/2)
16T(N/2)
16T(N/2)
16T(N/2)
1
1
1
1
1
1
Level k :
Work Done =
8^(log2 n) × T(1) = n^3

***Re − writing the above table as*:**

Level	***No. of problems***	***Problem Size***	***Work done = No. of problems × problem size***
0	**1**	n	$1 \times n = n$
1	**8**	$\frac{n}{2}$	$8 \times \frac{n}{2} = 4n$
2	8^2	$\frac{n}{4}$	$8^2 \times \frac{n}{4} = 16n$
.	.	.	.
.	.	.	.
k	8^k	$\frac{n}{2^k}$	$8^k \times \frac{n}{2^k} = n \times 2^{2k}$
$\log_2 n$	$8^{\log_2 n}$	1	$8^{\log_2 n} \times 1 = n^3$

Hence at level* 0 *, work done is N. At the next level , the cost is* 8 *times* $\left(\frac{N}{2}\right) = 4N$ *; at level* 2 *, the cost is* 64 *times* $\left(\frac{N}{4}\right)$ $= 16N$ *and so on.

***The work done is increasing in the following pattern*:**

$\mathbf{1, 4, 16, \ldots\ldots}$ ***Therefore the total cost is the work done at the last level and work done at all other level*** $\mathbf{(0, 1, 2, 3, \ldots,}$ $\mathbf{(\log_2 n - 1)}$.

***Thus , the total cost of the tree can be esitmated as follows*:**

$$\Longrightarrow \sum_{i=0}^{\log_2 n-1} 4^i n + 8^{\log_2 n} \times T(1)$$

Where , $8^{log_2 n} \times T(1)$ ***is the last level for*** $\log_2 n$.

and for $\log_2 n - 1$ ***i.e. till*** $\log_2 n - 1$ ***we have*** : $\sum_{i=0}^{\log_2 n-1} 4^i n$

Hence we got : $\sum_{i=0}^{\log_2 n-1} 4^i n + 8^{\log_2 n} \times T(1)$

And we know : $8^{\log_2 n} \times T(1) = n^3$, ***hence***:

$$\Longrightarrow \sum_{i=0}^{\log_2 n-1} 4^i n + n^3$$

$$\Rightarrow n \times \sum_{i=0}^{\log_2 n-1} 4^i + n^3$$

And we know the geometric series :

$$\sum_{k=1}^{n} k = 1 + x + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1}$$

And hence for $\sum_{i=0}^{\log_2 n-1} 4^i$ ***, we get***

$$\Rightarrow n \times \left(\frac{4^{(\log_2 n-1)+1} - 1}{4 - 1}\right) + n^3$$

$$\Rightarrow n \times \left(\frac{4^{\log_2 n} - 1}{4 - 1}\right) + n^3$$

$$\Rightarrow n \times \left(\frac{(n)^2 - 1}{4 - 1}\right) + n^3 \left[\because 4^{\log_2 n} = \left(2^{\log_2 n}\right)^2 = n^2\right]$$

$$\Rightarrow n \times \left(\frac{n^2 - 1}{3}\right) + n^3$$

$$\Rightarrow \frac{n^3 - n}{3} + n^3$$

$$\Rightarrow \frac{n^3 - n + 3n^3}{3}$$

$$\Rightarrow \frac{4n^3 - n}{3}$$

$$\textit{Hence total cost of the tree as} : \Theta\left(\frac{4n^3 - n}{3}\right)$$

$$\Rightarrow \Theta\left(\frac{4n^3}{3} - \frac{n}{3}\right)$$

$$\Rightarrow \Theta\left(\frac{4n^3}{3}\right)$$

$$\Rightarrow \frac{4}{3} \times \Theta(n^3)$$

$$\Rightarrow \Theta(n^3)$$

Alternative way,

Something is divisibe by 8 also are divisible by 4 or are multiples of 4.

The multiple 4 is the approach earlier, but if we keep the original table .

Level	***No. of problems***	***Problem Size***	***Work done = No. of problems × problem size***
0	**1**	n	$1 \times n = n$
1	**8**	$\frac{n}{8}$	$8 \times \frac{n}{8} = n$
2	8^2	$\frac{n}{8^2}$	$8^2 \times \frac{n}{8^2} = n$
.	.	.	.
.	.	.	.
k	8^k	$\frac{n}{8^k}$	$8^k \times \frac{n}{8^k} = n$
$\log_2 n$	$8^{\log_2 n}$	**1**	$8^{\log_2 n} \times 1 = n^3$

$$\textbf{Hence,}$$

$$= (n + n + n + \cdots + \log_2 n - 1 \text{ times}) + n^3$$

$$= \text{n} \times (\log_2 n - 1) + n^3$$

$$= \text{n} \log_2 n - n + n^3$$

$$= \Theta(\text{n} \log_2 n - n + n^3)$$

$$\textbf{Now let us view the exponential rates of growth:}$$

$$2^{2n} < n! < 4^{2n} < 2^n < n^3 < n^2 < nlogn < log(n!) < n$$

$$< 2^{logn} < log^2 n < \sqrt{logn} < loglogn < 1$$

$$\textbf{And complexities from fastest to slowest:}$$

$$\Theta(1) < \Theta(log\ n) < \Theta\left(\sqrt{logn}\right) < \Theta\left(\sqrt{n}\ \right) < \Theta(n) < \Theta(nlogn)$$

$$< \Theta(n^2) < \Theta(n^3) < \Theta(2^n) < \Theta(n!) < \Theta(2^{2n}) < \Theta\left(2^{logn}\right)$$

$$< \Theta(loglogn) < \Theta(3^n) < \Theta(n^n)$$

$$\textbf{We can write it oppositely:}$$

$$\Theta(n^n) > \Theta(3^n) > \Theta(loglogn) > \Theta\left(2^{logn}\right) > \Theta(2^{2n}) > \Theta(n!)$$

$$> \Theta(2^n) > \Theta(n^3) > \Theta(n^2) > \Theta(nlogn) > \Theta(n) > \Theta\left(\sqrt{n}\ \right) >$$

$$\Theta\left(\sqrt{logn}\right) > \Theta(log\ n) > \Theta(1)$$

$\boldsymbol{We\ can\ examine:}$

$\boldsymbol{\Theta(\ n\log_2 n - n + n^3)}$

$\boldsymbol{\Rightarrow \Theta(n^3) > \Theta(n\log_2 n) > \Theta(n)}$

$\boldsymbol{\Rightarrow \Theta(n^3)\ is\ the\ anwer.}$

******