$\frac{n/2}{2}$

$\frac{n}{4}$

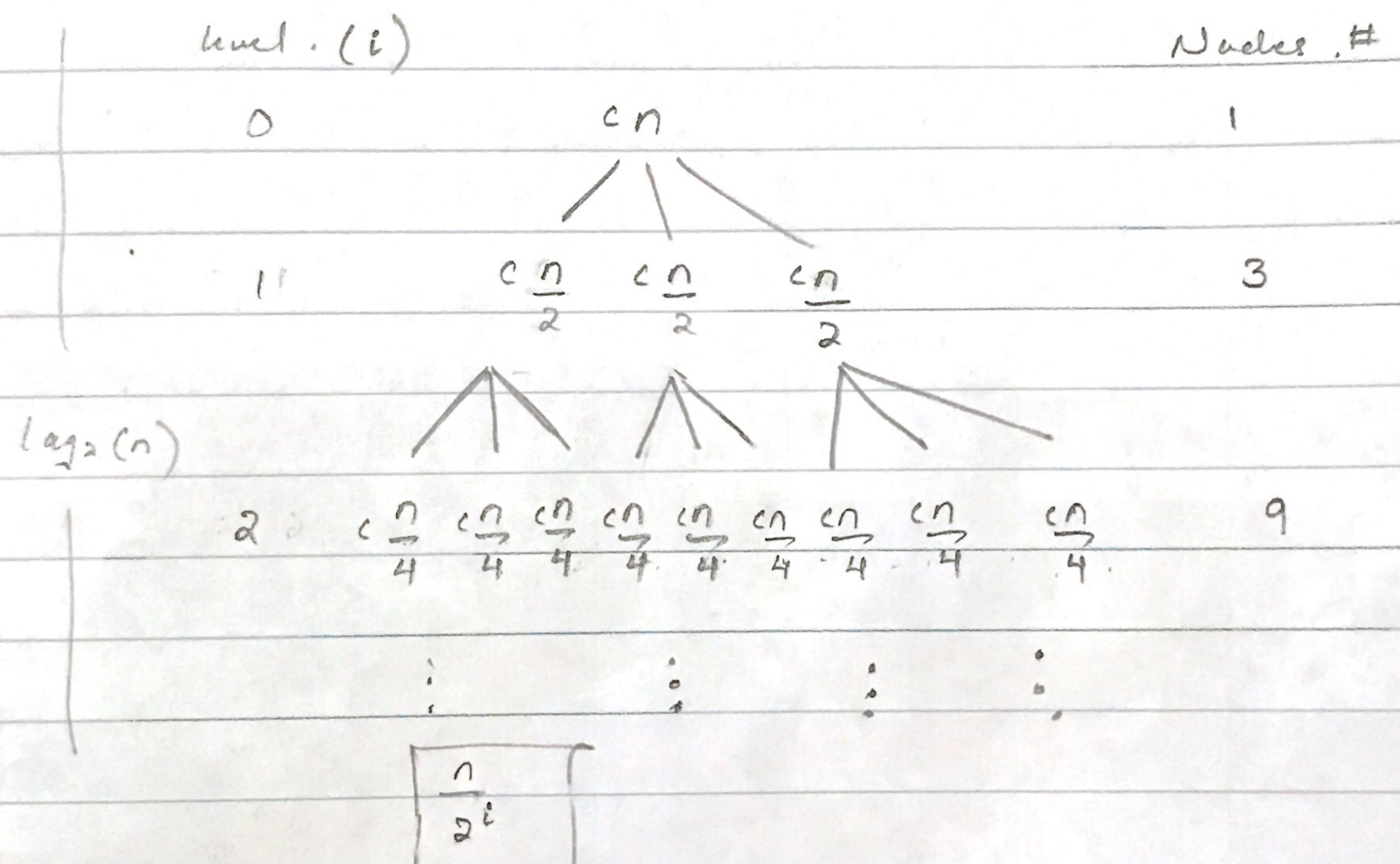
Q2. Draw a recursion tree for $T(n) = 3T(n/2) + n$ and provide an asymptotic upper bound on its solution:

$a = 3, \quad b = n/2$

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

$T(n/2) = 3T(n/4) + \frac{n}{2}$

$T(n/2) = 3T(\frac{n}{8}) + \frac{n}{4}$



Now height of tree:

The sub-problem size for a node at any level (i) (depth) i is $\left[\frac{n}{2^i}\right]$

The sub-problem size becomes 1 when $\left\lceil \frac{n}{2^i} \right\rceil = 1$

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

or, $n = 2^i$

or, $\log_2(n) = \log_2 2^i$ $\qquad$ $\log_2 8 = 3$

or, $\boxed{i = \log_2(n)}$

Now cost of each level : (Except bottom)

Number of Nodes at each level (i) = 3^i

Cost of each Node for any level i : $c \times \left(\frac{n}{2^i}\right)$

$\therefore$ Total cost for each level : $3^i \times c \times \left(\frac{n}{2^i}\right)$

$$= \left(\frac{3}{2}\right)^i cn$$

Now for cost of last level

Cost for last level.

(1) Level : $\log_2(n)$ ← Last level

Number of Nodes $= 3^i = 3^{\log_2(n)}$

$= n^{\log_2 3}$ (From properties)

here, each node cost $T(1) = k$ (constant)

Total cost on bottom level :

$$k n^{\log_2 3} = \Theta(n^{\log_2 3})$$

Now we have to determine upper bound.

$$T(n) = cn + \frac{3}{2} cn + \left(\frac{3}{2}\right)^2 .cn + \dots \left(\frac{3}{2}\right)^{\log_2(n) - 1}$$

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

$$= \sum_{i=0}^{\log n - 1} \left(\frac{3}{2}\right)^i cn + \Theta(n^{\log_2 3})$$

$$= \frac{(3/2)^{\log_2 n} - 1}{(3/2) - 1} cn. + \Theta(n^{\log_2 3}).$$

$$= O(n^{1.58})$$