Last class: substitution method.

Recursion tree method

- A recursion tree captures the cost of a recursive execution of an algorithm.
- It gives a guess of the running time, and thus provides a guess for the substitution method.

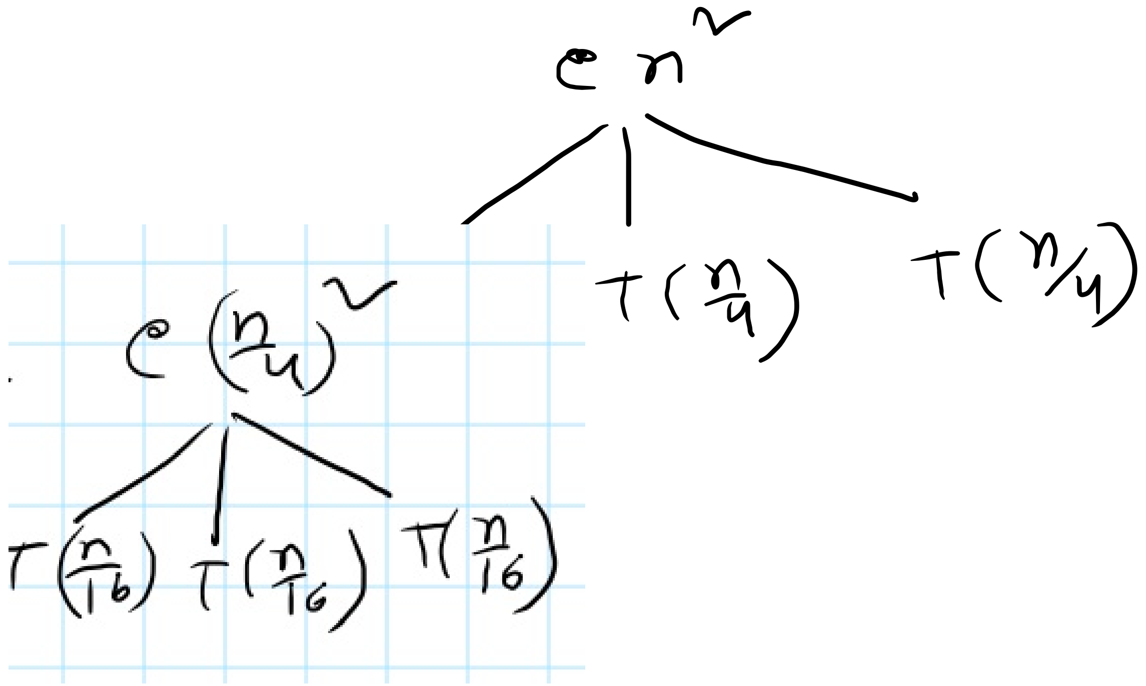
Ex^n Guess the solution of the recurrence

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

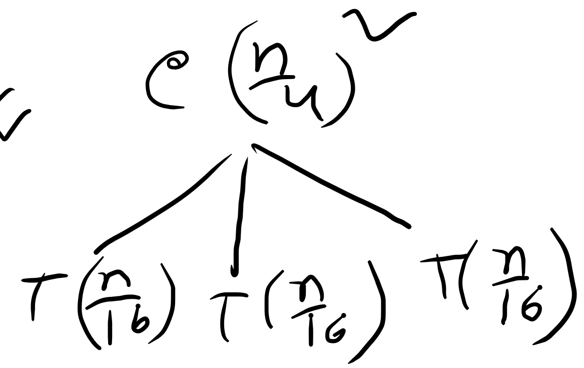
we can write is as

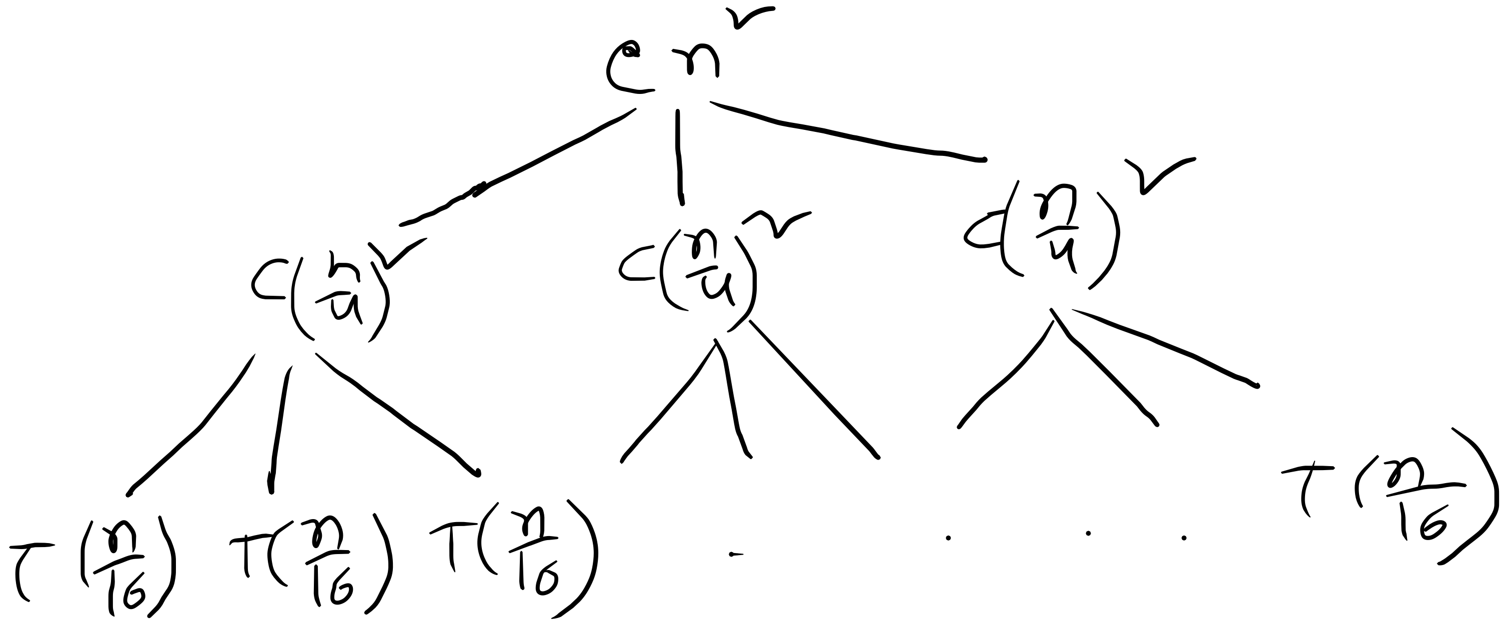
$$T(n) = 3T\left(\frac{n}{4}\right) + cn^2$$

$T(n)$



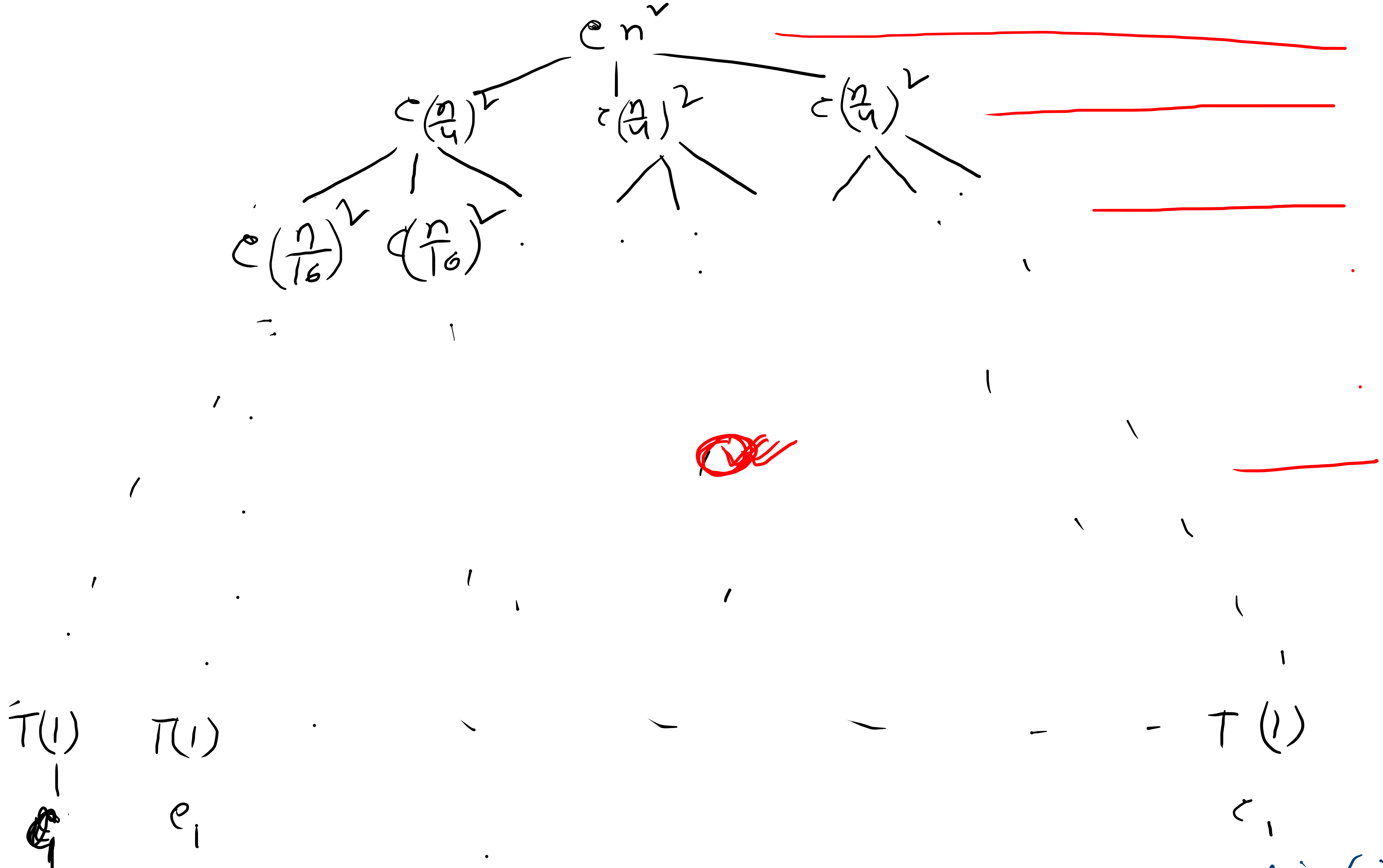
$T\left(\frac{n}{4}\right) \approx$





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

$$T\left(\frac{n}{16}\right) = 3T\left(\frac{n}{64}\right) + c\left(\frac{n}{16}\right)^2$$



cn^2

$\frac{3}{16}cn^2$

$\left(\frac{3}{16}\right)^2 cn^2$

i-th depth cost: $\left(\frac{3}{16}\right)^i cn^2$

Height of the tree: let h then h satisfies $\frac{n}{4^h} = 1 \Rightarrow \boxed{h = \log_4 n}$

How many $T(1)$'s are there

cost at height -0 is $3^h = 3^{\log_4 n} = n^{\log_4 3}$

The total cost of the tree: $T(n)$

$$T(n) = \sum_{i=0}^{\log_4 n - 1} \left(\frac{3}{16}\right)^i cn^2 + c_1 n^{\log_4 3}$$

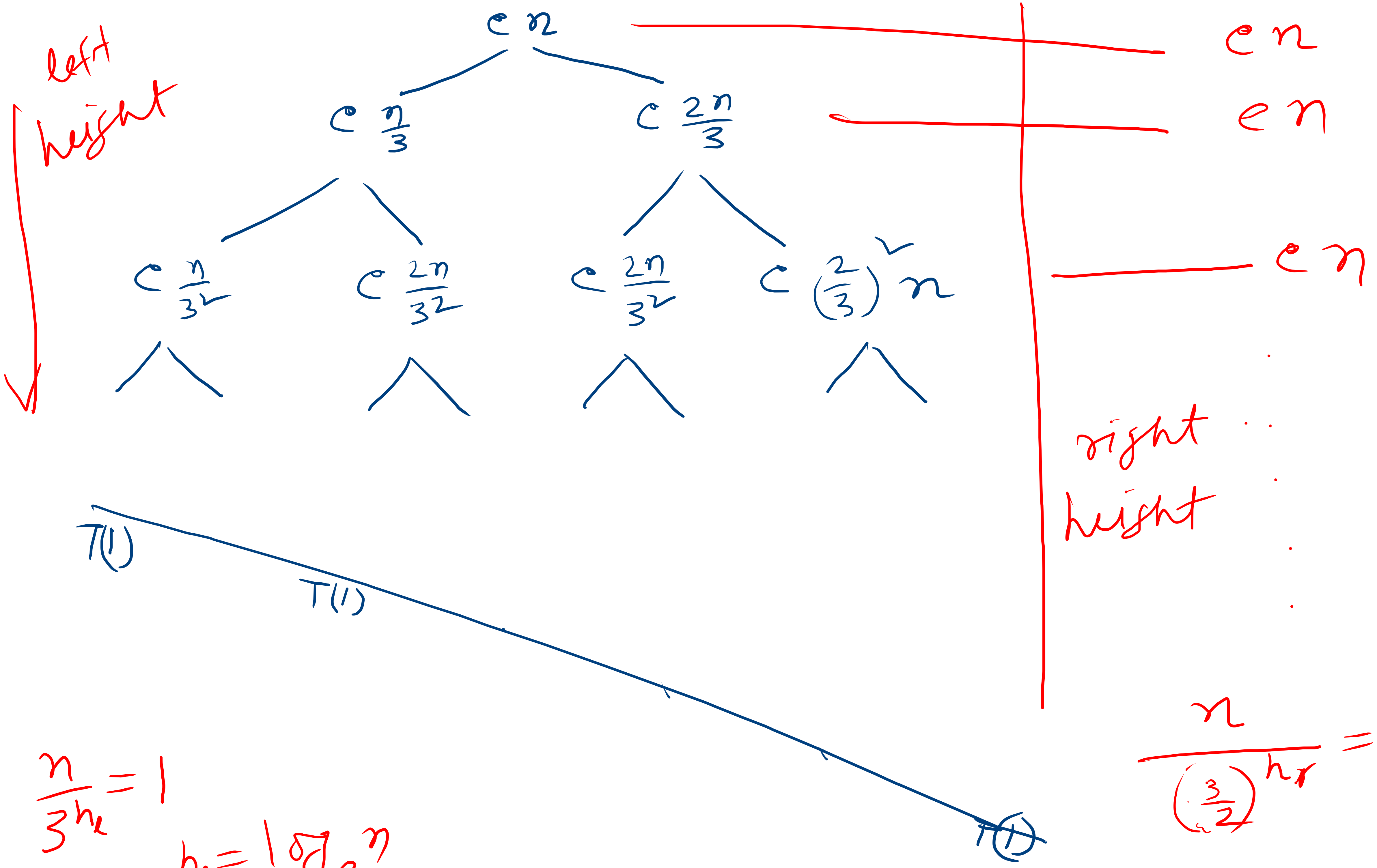
$$\leq \sum_{i=0}^{\infty} \left(\frac{3}{16}\right)^i cn^2 + c_1 n^{\log_4 3}$$

$$= \frac{1}{1 - \frac{3}{16}} cn^2 + c_1 n^{\log_4 3}$$

$$= \frac{16}{13} cn^2 + c_1 n^{\log_4 3} \quad \Rightarrow T(n) = O(n^2)$$

Guess the solution by recursion tree method.

$\underline{\text{Ex}^m}$ $$T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + cn$$



$$\frac{n}{3^{h_l}} = 1$$

$$h_l = \log_3 n$$

$$\frac{n}{\left(\frac{3}{2}\right)^{h_r}} = 1 \Rightarrow \log_{2/3} n$$

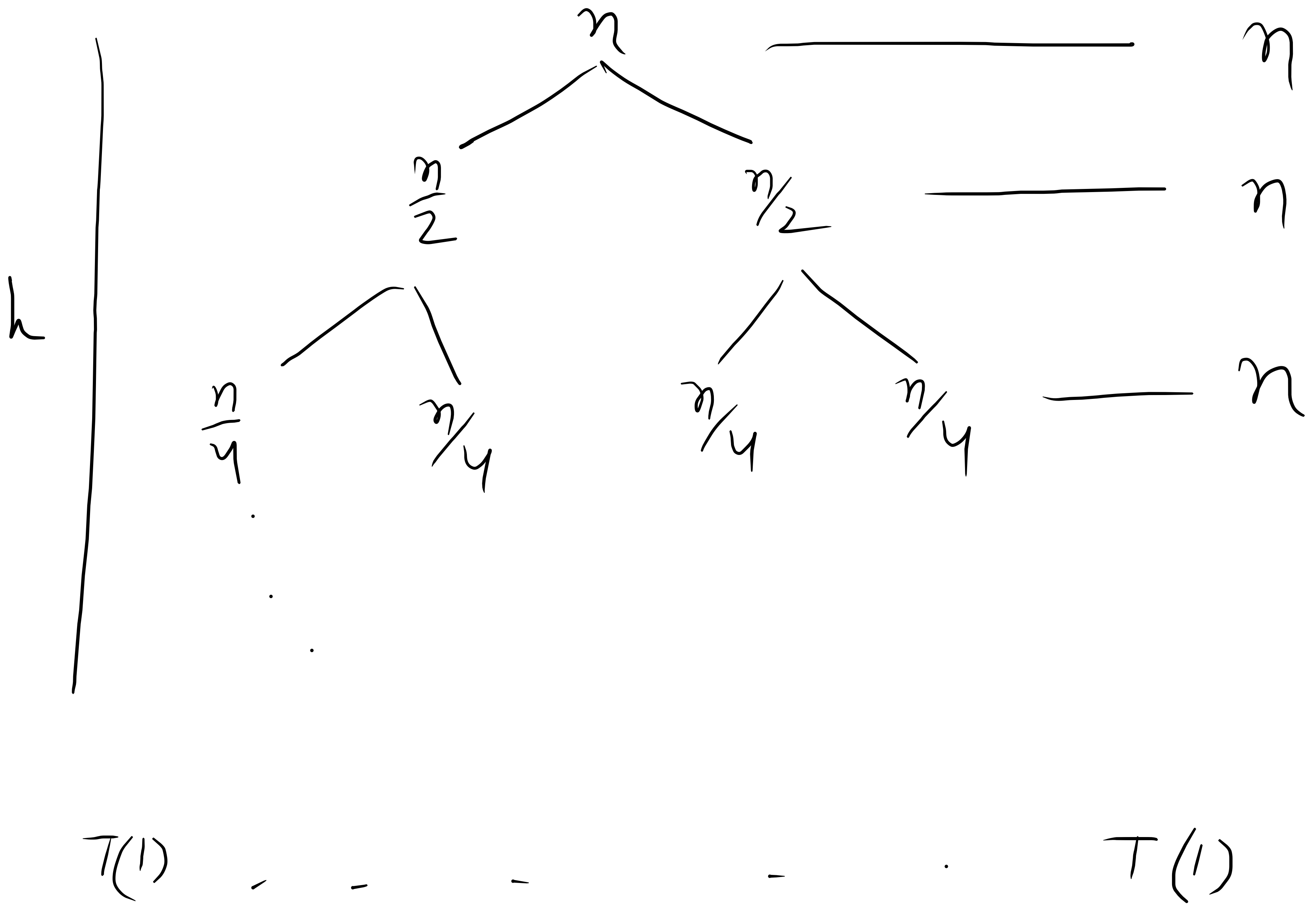
$$cn \log_3 n \le \text{cost of the tree} \le cn \log_{2/3} n$$

Ω $\qquad\qquad$ O

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

Ex^{m}

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



Total cost of the tree

$$T(n) = n \cdot \log_2 n$$

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

$$\frac{n}{2^h} = 1 \Rightarrow h = \log_2 n$$