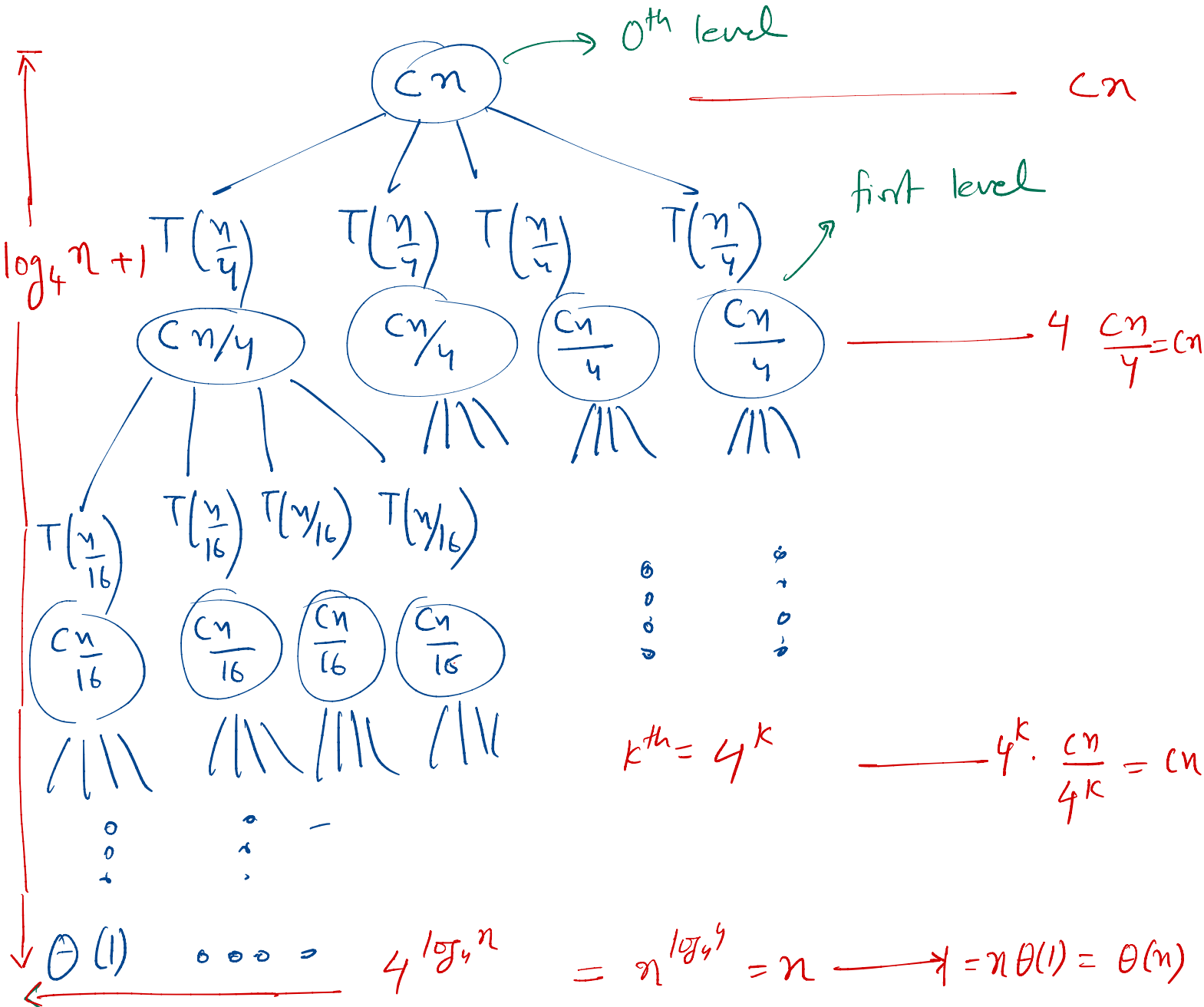
Recursion Tree Method

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

$$= \underbrace{T\left(\frac{n}{4}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{4}\right)}_{\text{4 times cost of solving problem of half the size}} + \underbrace{cn}_{\text{cost at a level}}$$



$$T(n) = \underbrace{cn + cn + \ldots + cn}_{\log_4 n} + \Theta(n)$$

↘ cost at leaf nodes

$$= (\log_4 n)\, cn + \Theta(n)$$

$$= c\, n \log_4 n + \Theta(n)$$

$$= \Theta(n \log_4 n)$$

Note:

$n\ \Theta(1)$

$= n\ c$

$= \Theta(n)$

Master Method

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

$f(n)$ → driving function

$n^{\log_b a}$ → Watershed function

?.

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

$a = 3$

$b = 2$

$$n^{\log_b a} = n^{\lg_2 3}$$

$$f(n) = n$$

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