

master theorem:

MT is applicable only to recursive relations of form

$$T(n) = \underbrace{a T(n/b)}_{\cdot} + f(n), \underbrace{a \geq 1}_{\cdot}, \underbrace{b > 1}_{\cdot}, \underbrace{a, b \rightarrow \text{constants}}_{\cdot}$$

$f(n) \rightarrow$ function of ' n '

Ex: $T(n) = 10T(n/2) + n^2$

$$T(n) = 8T(n/4) + n \log n.$$

Masters theorem :-

Let $T(n) = aT(n/b) + f(n)$, a, b are constants

$$a \geq 1, b > 1$$

Case(i) If $f(n) = O(n^{\log_b a - \epsilon})$ $\epsilon > 0$ then $T(n) = \Theta(n^{\log_b a})$

Case(ii): If $f(n) = \Omega(n^{\log_b a + \epsilon})$ $\epsilon > 0$, then $T(n) = \Theta(f(n))$

Case(iii): If $f(n) = \Theta(n^{\log_b a} \cdot (\log n)^k)$ where k is constant,

$$K \geq 0, \text{ then } T(n) = \Theta(n^{\log_b a} (\log n)^{k+1})$$

Don't follow these rules

$$T(n) = a T\left(\frac{n}{b}\right) + f(n)$$

- $f(n)$ $n^{\log_b a}$
- 1) which one is greater.
- 2) Is it greater by polynomial time

$\Theta(\text{greater term})$

=

$f(n)$

n^3 $\Theta(n^3)$ n^2 $n^{\log b}^a$

$2^{2.81}$ $\Theta(2^{2.81})$ n^2

n^{50} $\Theta(n^{100})$. n^{100} n^{50}

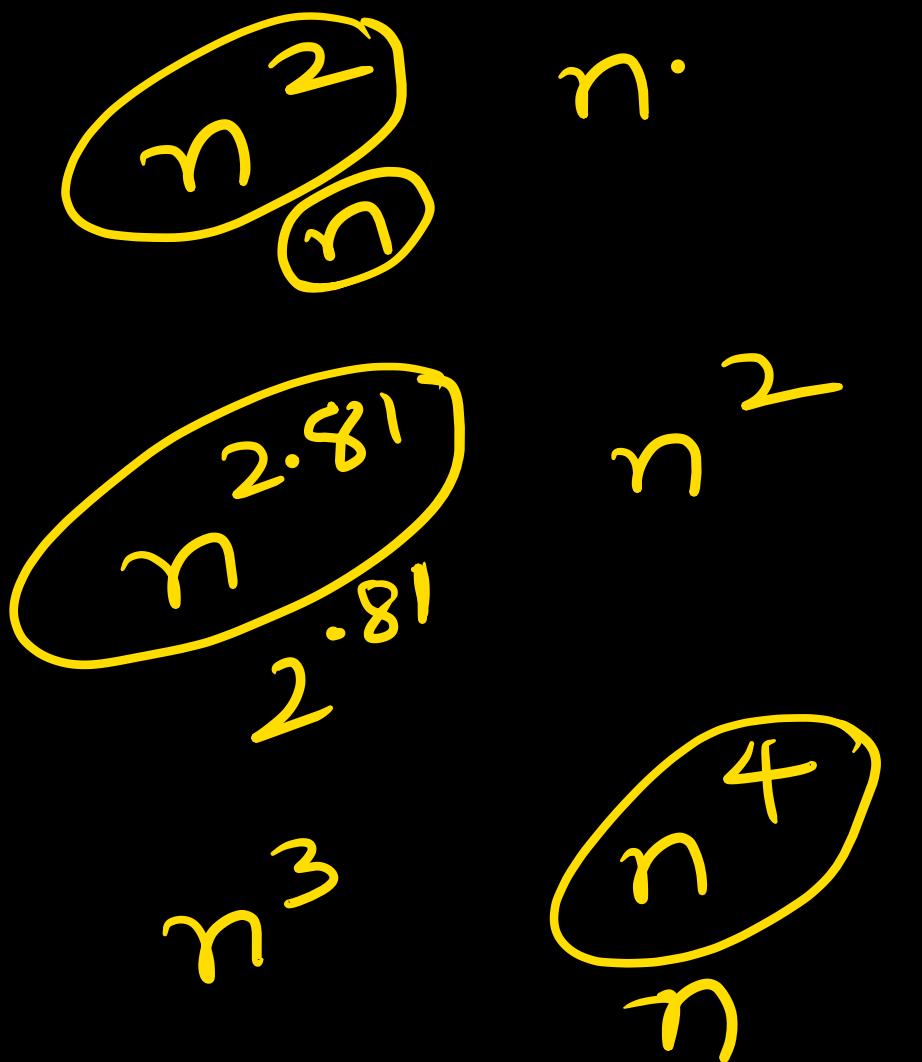
$$\begin{aligned}
 & \frac{f(n)}{n^{\log b}} \\
 & \frac{n^{\log n}}{n^{\log b}} > n \\
 & \text{But it is } \nearrow \text{ polynomial} \\
 & \text{greater } (\log n)^k \\
 & \text{make them equal} \\
 & \underline{\underline{n \times (\log n)^{1-k}}}
 \end{aligned}$$

$$\begin{aligned}
 & = \Theta(n(\log n)^{k+1}) \\
 & = \Theta(\overline{n(\log n)^2})
 \end{aligned}$$

$$f(n) = n(\log n)^{10}$$
$$g(n) = n^k$$

But not polynomial greater.

$$\Theta(n^{(\log n)^{10}}) \xrightarrow{k+1} =$$



$$(\log n)^2 = \log n \log n$$

$$\log^2 n = \underline{\log} \underline{\log} n.$$

$$f(n) = n(\log n)^5$$
$$n^{\log_5 a}$$

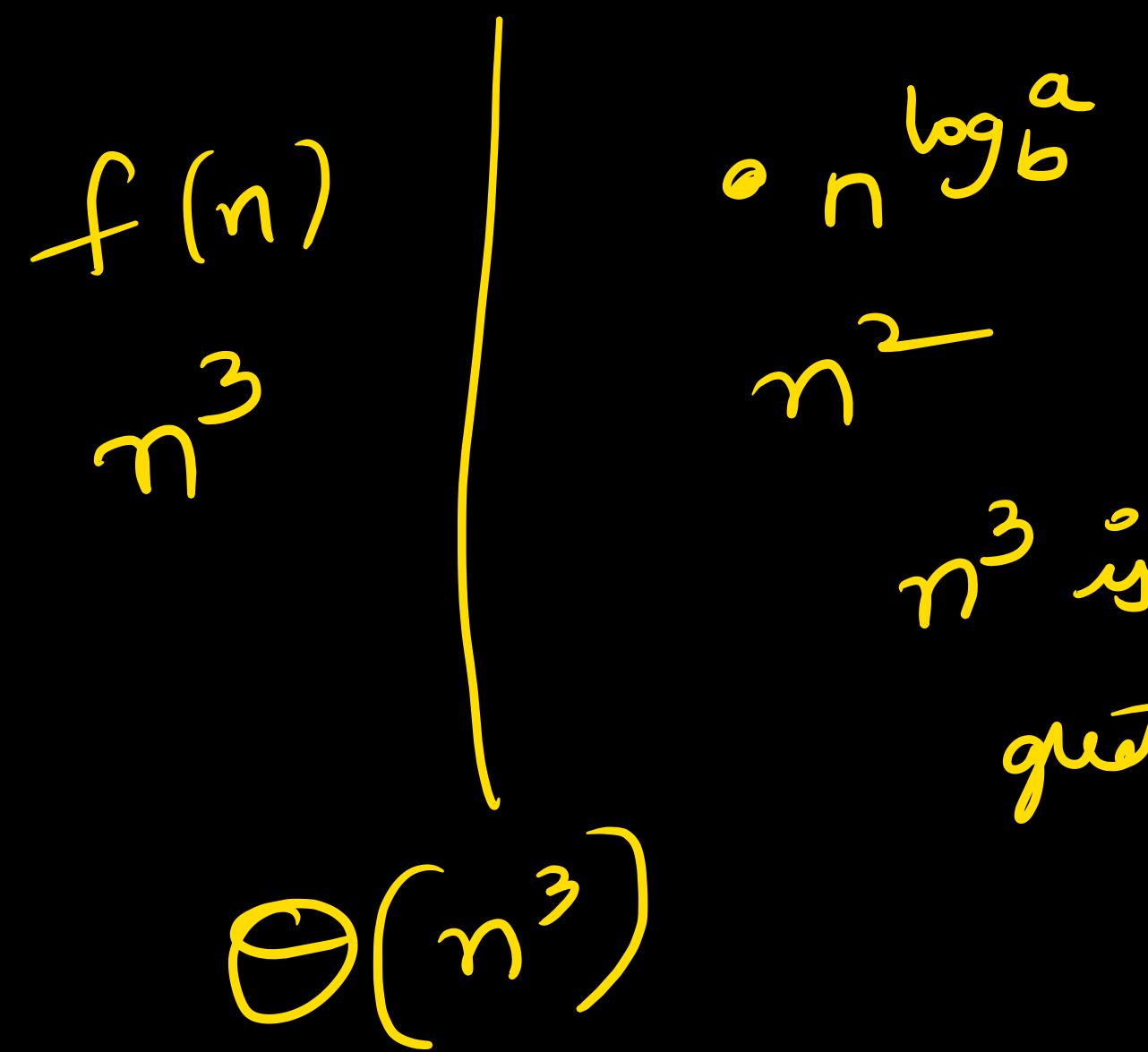
not polynomial greater.

$$n(\log n)^5 = n(\log n)^5 \cdot n^k = \Theta(n(\log n)^6)$$

$$f(n) = n^2 \log n > n^{\log b}$$

$n \log n$ is greater by polynomial time.

$$\Theta(n^2 \log n)$$

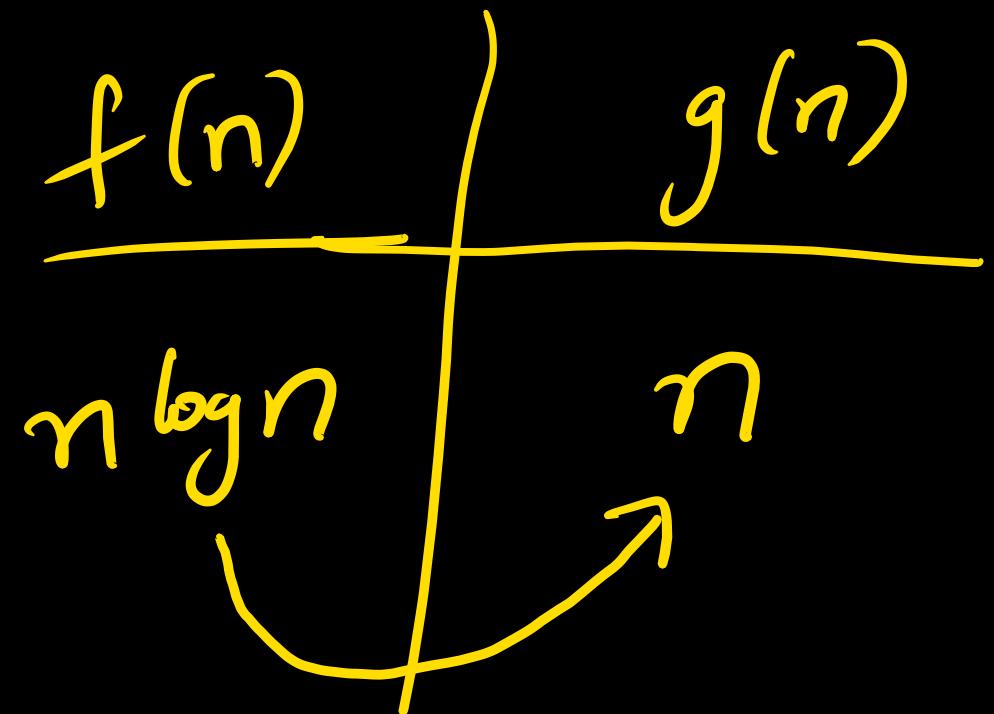


n^3 is greater polynomial
greater than n^2 .

$$\frac{f(n)}{n^4} \quad \frac{g(n)}{n^5}$$

$\Theta(n^5)$

n^5 is polynomial time
greater.



$$\Theta(n(\log n)(\log n)) = \Theta(n(\log n)^2)$$

↓
multiply with
 $\log n$

$$f(n) \geq n^{\log_b a}$$

$n(\log n)^{10} > n$
But it is not polynomial greater.

$$\Theta(n(\log n)^{10} * \log n) = \Theta(n(\log n)^{11})$$

↓
multipy
 $\log n$

$$\frac{f(n)}{g(n)} = \frac{n^2(\log n)^0}{n^2}$$

$$(n^2(\log n)^{0+1}) \\ = n^2 \log n$$

$$\frac{f(n)}{g(n)} = \frac{n^{\log b}}{n^3} \\ \Theta(n^{3-\log b})$$

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

$$a=8, b=2$$

$$\begin{aligned} f(n) &= n^{\log_b^a} \\ &= n^{\log_2 8} = n^3 \end{aligned}$$

$$\begin{aligned} f(n) &= n^2 \\ &= \end{aligned}$$

$$\begin{aligned} n^{\log_b^a} &= 2^n \\ \Theta(2^n) &= \end{aligned}$$

greater by
polynomial time

Gate: $T(n) = 2T(n/2) + n^2$

$$a = 2$$

$$b = 2$$

$$n^{\log_a b} = n$$

$$f(n)$$



$$n^2$$

$$\downarrow$$

$$n$$

$$\Theta(n^2)$$

polynomial
grows.



Recurrence: $T(n) = 2T(n/2) + n$

$$\begin{array}{ccc} f(n) & & n^{\log_b a} \\ \downarrow & & \downarrow \\ n & = & n \end{array}$$

$$\Theta(n^{\log n})$$

$$T(n) = 7(T(n/2)) + n^2$$

$$a = 7 \quad b = 2$$

$$f(n) \downarrow n^2$$

$$n^{\log_5 7}$$

$$n^{\log_2 7}$$

$$\Rightarrow n^{2.81}$$

greater by polynomial time.

$$\Theta(n^{2.81})$$

==

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

$$a=1 \quad b=2$$

$$f(n)$$



$$n^{\log_b a}$$

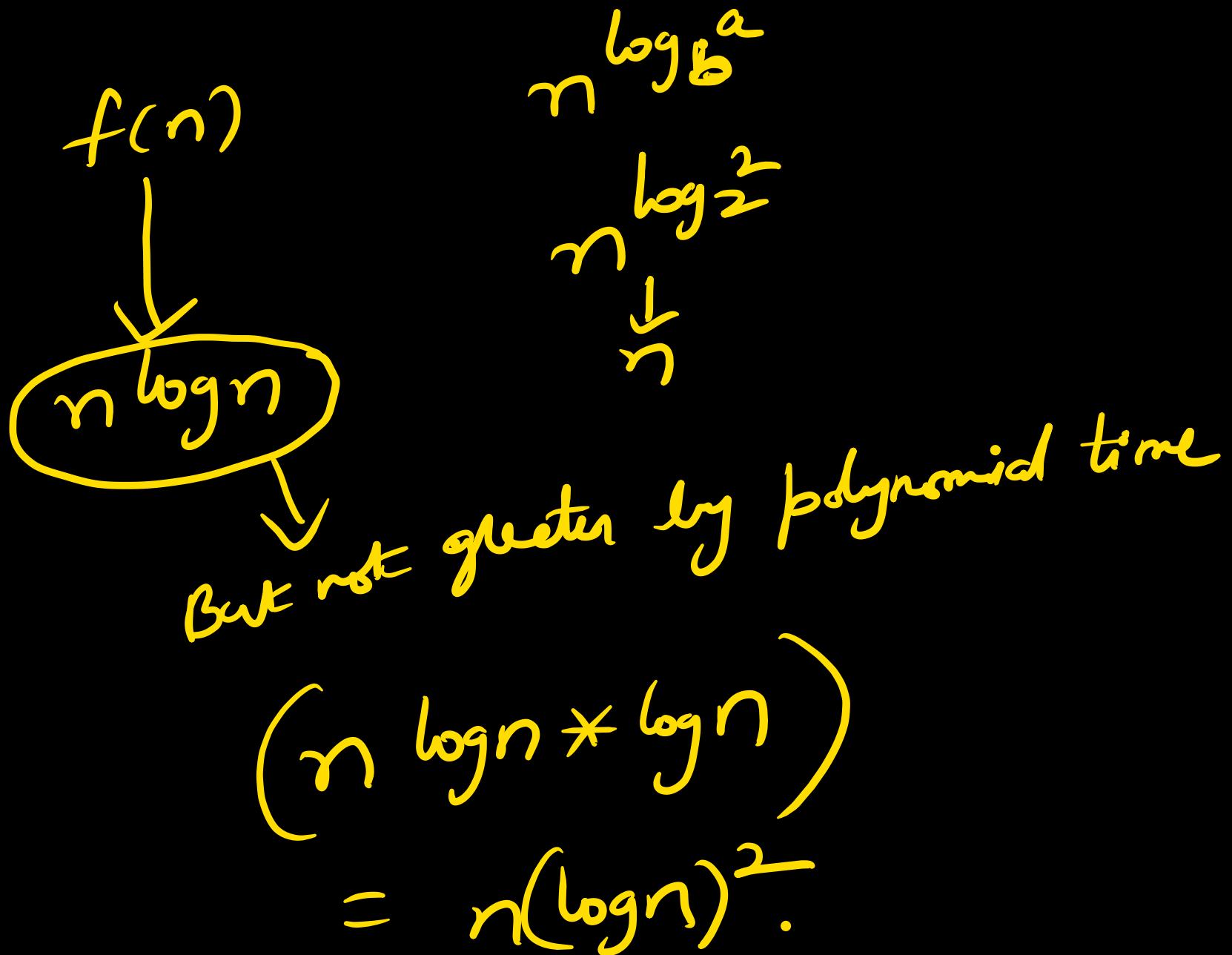
$$n^{\log_2 1}$$



$$\Theta(n)$$

=

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



$$T(n) = 8T(n/2) + n^3 \log n$$

$f(m)$

$n^3 \log n$

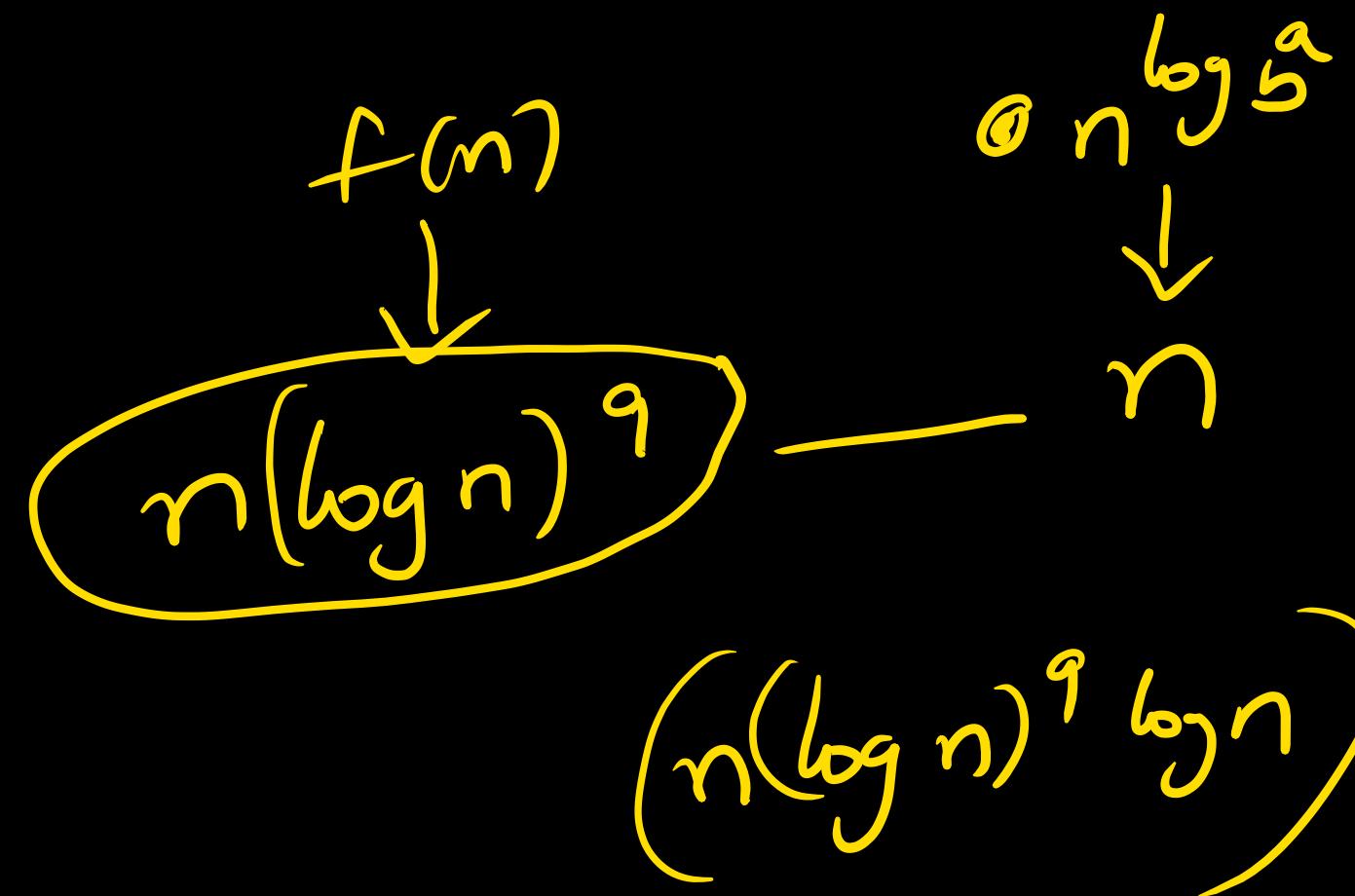
$n^{\log b} a$

n^3

~~$n^3 \log^{\frac{1}{2}} n$~~

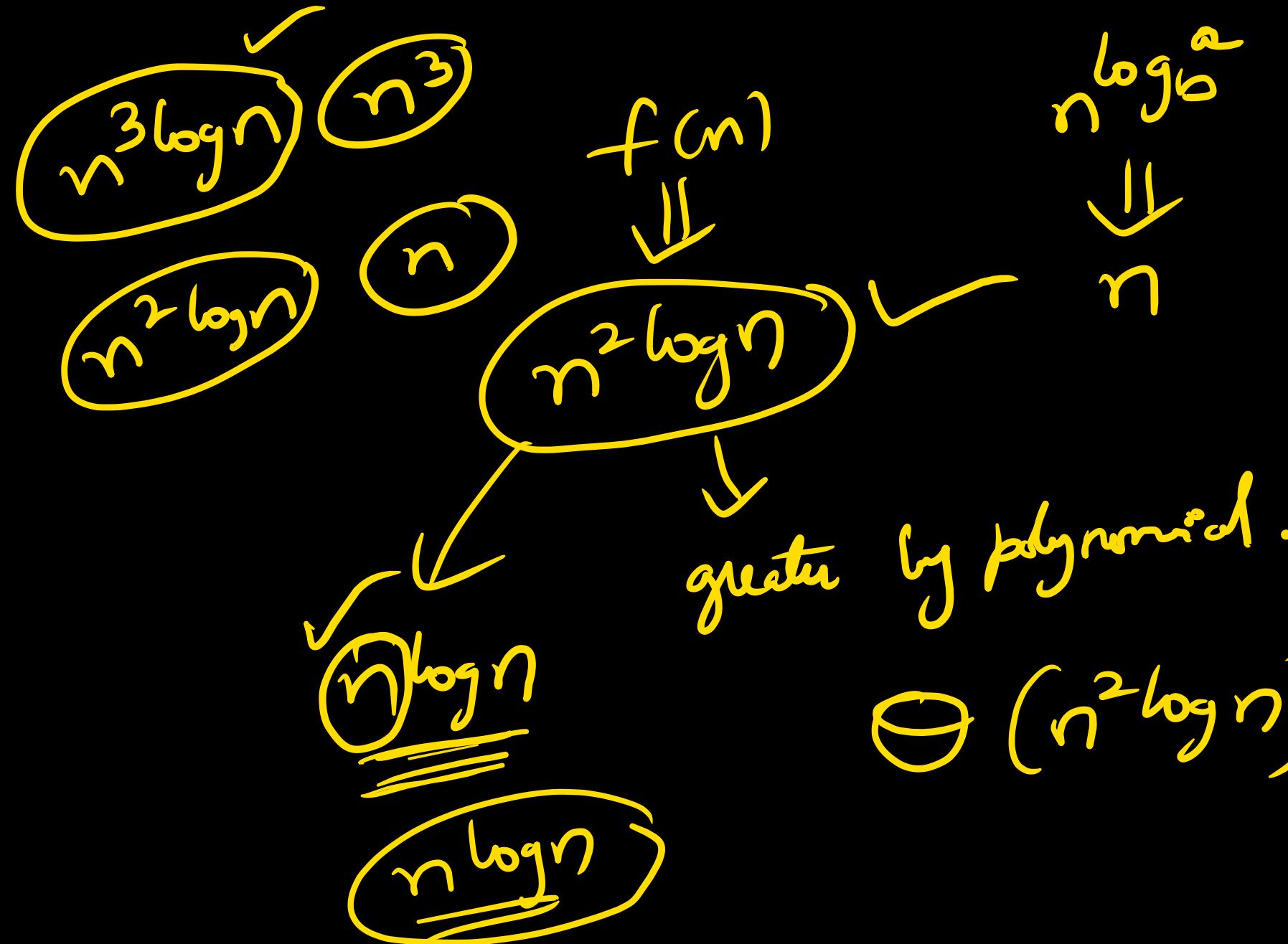
$n^3 (\log n)^2$

$$T(n) = 2T(n/2) + n \cdot (\log n)^9$$



Don't ask me why?
It is complicated to prove.

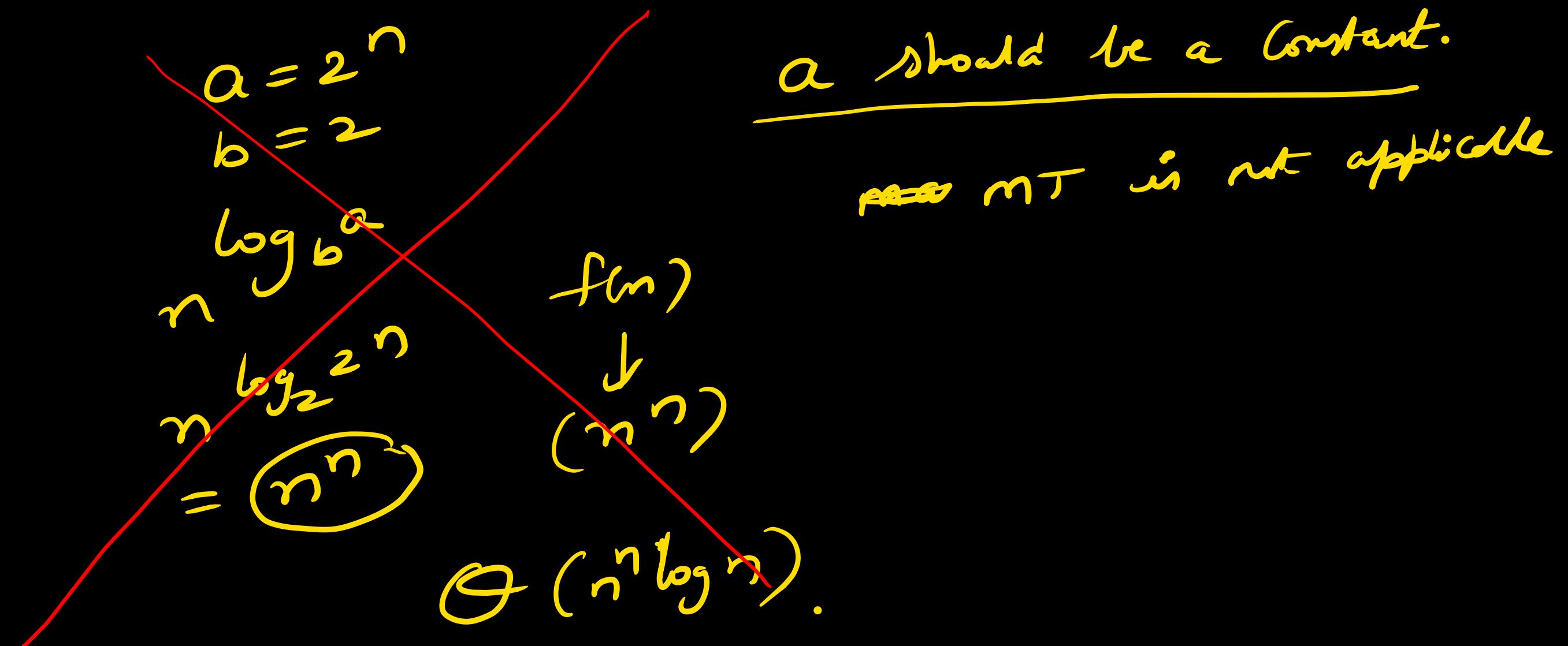
$$T(n) = 2T(n/2) + n^2 \log n$$



$n^2 \log n$ is polynomial
greater than n .

$n^3 \log n$ n^3
 $n^2 \log n$ n

$$T(n) = 2T\left(\frac{n}{2}\right) + n^n \Rightarrow \text{Substitution} \Rightarrow \text{big answe} \underline{\underline{r}}$$



$$T(n) = 0.5 T\left(\frac{n}{2}\right) + n$$

MT \times applicable.

$$a \geq 1$$

$$\begin{array}{c} \times \times \times \times \\ T(n) = T(\sqrt{n}) + C \end{array}$$

assume that $n = 2^K$

$$T(2^K) = T(2^{K/2}) + C$$

assume that $T(2^K) = S(K)$

MT is applicable.

$$S(K) = S(K/2) + C$$

$$\begin{array}{l} f(K) = K^{\log_b a} = K^{\log_2 1} \\ \downarrow \\ C = C = \Theta(C \log K) \end{array}$$

$$S(k) = \Theta(\log k)$$

$$T(2^k) = S(k)$$

$$S(k) = \Theta(\log k)$$

$$T(2^k) = \Theta(\log k)$$

$$n = 2^k \Rightarrow k = \log n$$

$$T(2^{\log_2 n}) = \Theta(\log_2 \log_2 n)$$

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

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

assume

$$n = 2^k$$

$$\underline{T(2^k)} = \underline{T(2^{k/2})} + C$$

assume $T(2^k) = S(k)$

$$T(2^k) = S(k) \quad \therefore S(k) = \Theta(c \log k)$$
$$T(2^k) = S(k) + C$$

$$f(k) \quad S^{\log_b^a}$$
$$\Downarrow \quad \Downarrow$$
$$C \quad C$$

$$S(k) = \Theta(\log k)$$

$$\Downarrow$$
$$T(2^k) = \Theta(\log k)$$

$$\textcircled{n} = 2^k \Rightarrow k = \log n$$

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

$$T(n) = \underline{\underline{\Theta(\log \log n)}}$$

$$T(n) = 2T(\sqrt{n}) + \log_2 n$$

assume $n = 2^k \Rightarrow \log n = k$

$$T(2^k) = 2T(2^{k/2}) + k$$

assume $T(2^k) = S(k)$

$$S(k) = 2S(k/2) + k$$

$$\begin{array}{ccc} f(k) & K^{\log_2 k} & S(k) = k(\log k) \\ \Downarrow & \Downarrow & \end{array}$$

\Rightarrow I have not given
base conditions

But in exam
base conditions
will be q given.

$$\underline{\underline{S(K) = \Theta(K \log K)}}$$

$$T(2^K) = S(K)$$

$$T(2^K) = \Theta(K \log K)$$

$$2^K = n \Rightarrow \log n = K$$

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

$$(K \log K)$$
$$K = \underline{\underline{\log n}}$$

$$(\log \log \log n).$$

$$\begin{aligned}
 T(n) &= T(\sqrt{n}) + C \\
 &= T(n^{1/2}) + C \\
 &= T(n^{1/4}) + C + C \\
 &= T(n^{1/8}) + C + C + C \\
 &= T(n^{1/2^3}) + 3C \\
 &\quad \left. \begin{array}{l} \text{\{} \\ \text{K time} \\ \text{\}} \end{array} \right\} \\
 &= T(n^{1/2^K}) + KC
 \end{aligned}$$

Substitution

let $T(2) = D$
given in Ques.

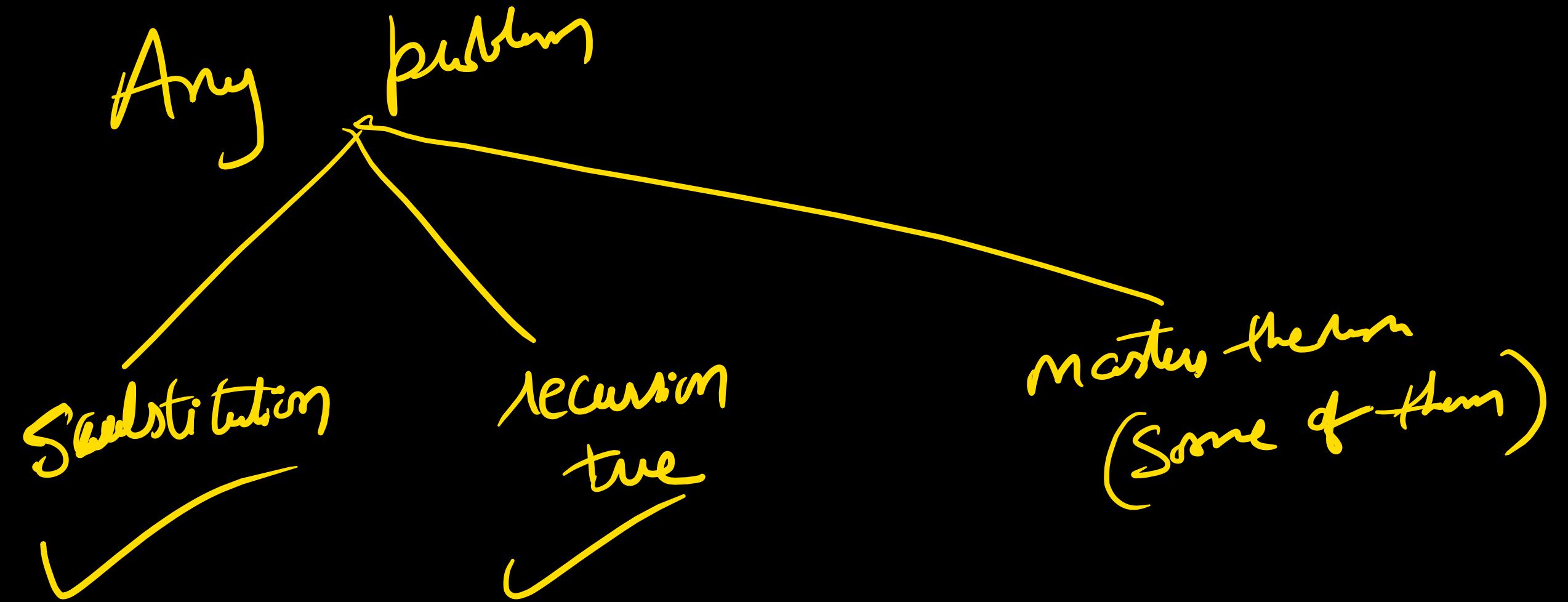
$$n^{1/2^K} = 2$$

$$\frac{1}{2^K} = \log n = 1$$

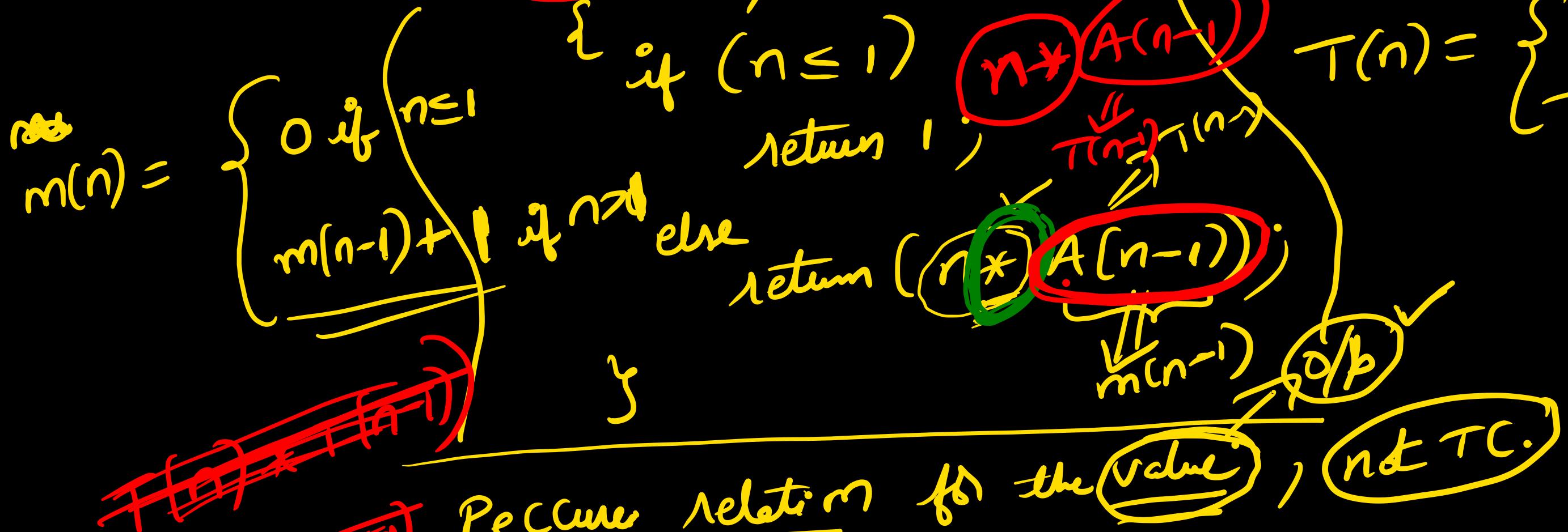
$$2^K = \log n$$

$$K = \underline{\log \log n}$$

$$\begin{aligned}
 T(n) &= 2 + \log \log n \\
 &= \Theta(\log \log n)
 \end{aligned}$$



function:



~~Algorithm~~ ~~Implementation~~ Recursive relation for the value, not TC.

$$A(n) = \begin{cases} 1, & n \leq 1 \\ n * A(n-1) & n > 1 \end{cases}$$

A(n)

if ($n \leq 1$) return (n)

else

return ($A(\underline{n/3}) + A(\underline{2n/3}) + n$)

RR fn value = α/ρ .

$v(n) = \begin{cases} n, & n \leq 1 \end{cases}$

$v(n) = \begin{cases} v(n/3) + v(2n/3) + n \end{cases}$

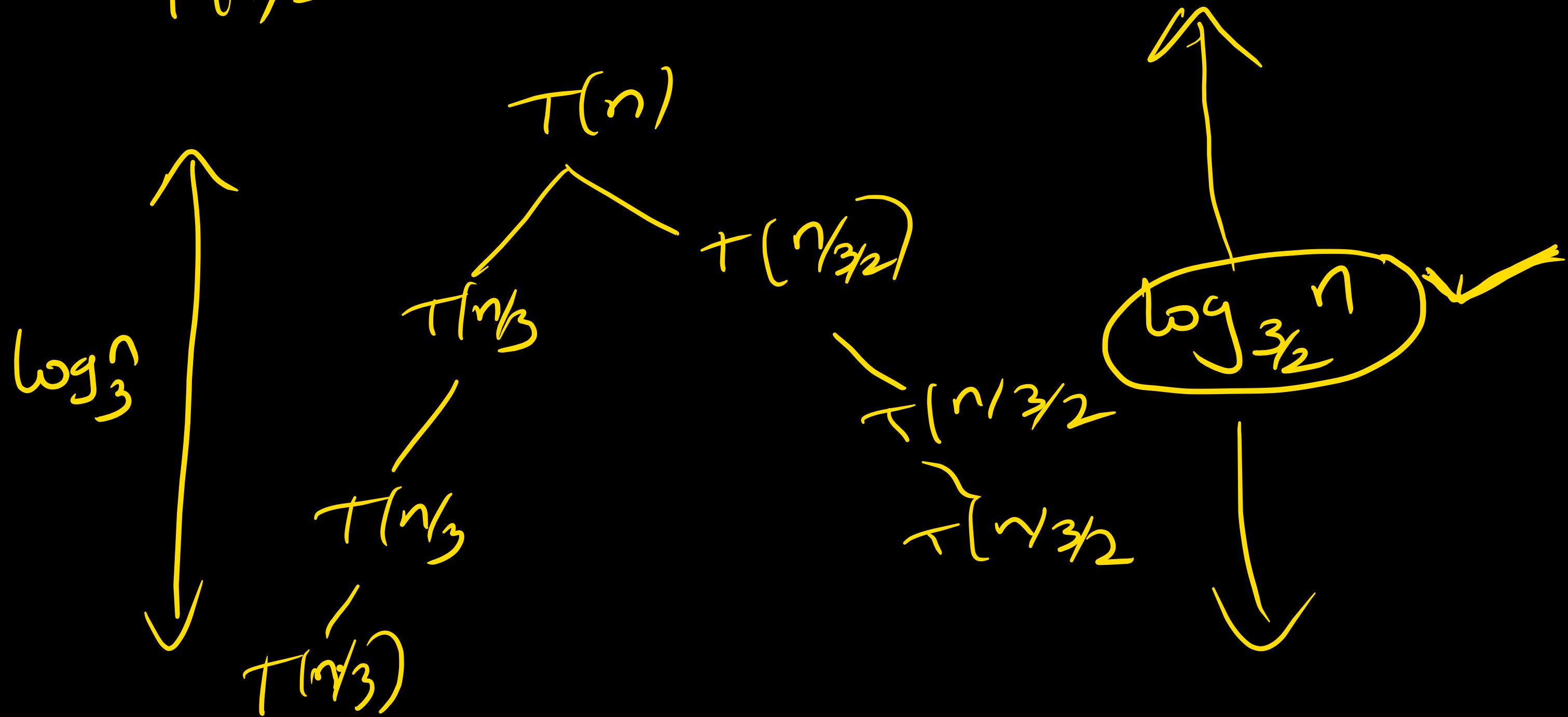
RR TC.

$$T(n) = \begin{cases} C & \text{if } n \leq 1 \\ T(\underline{n/3}) + T(\underline{2n/3}) + C \end{cases}$$

RR for additions

$$\text{add}(n) = \begin{cases} 0 & \text{if } n \leq 1 \\ 2\text{add}(n/3) + \text{add}(2n/3) + 2 \end{cases}$$

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



$f(n) \Rightarrow T(n)$

{

if ($n == 0$ || $n == 1$)

return (n)

else

return ($\underline{\underline{f(n-1)}} + \underline{\underline{f(n-2)}}$)

\Downarrow
 $t(n-1)$

\Downarrow
 $T(n-2)$

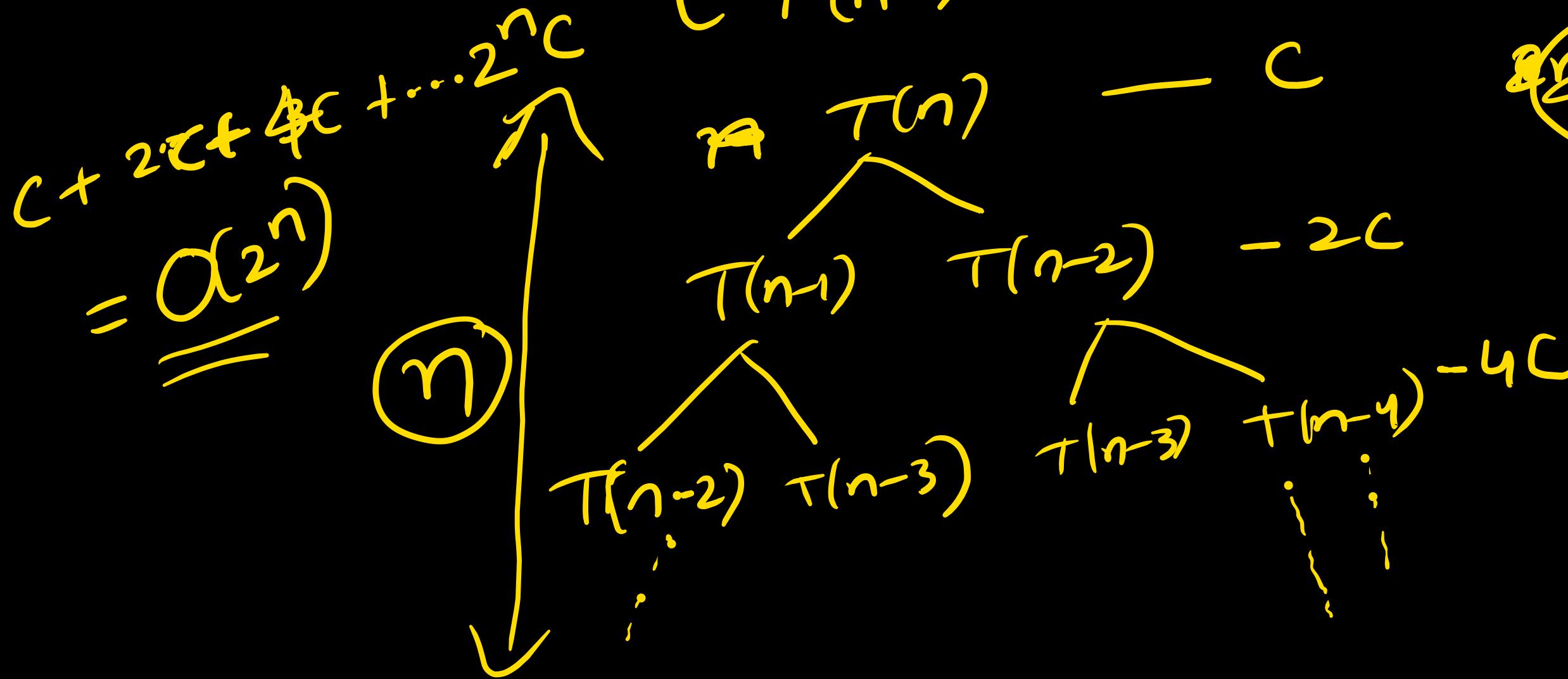
}

RR fn value -

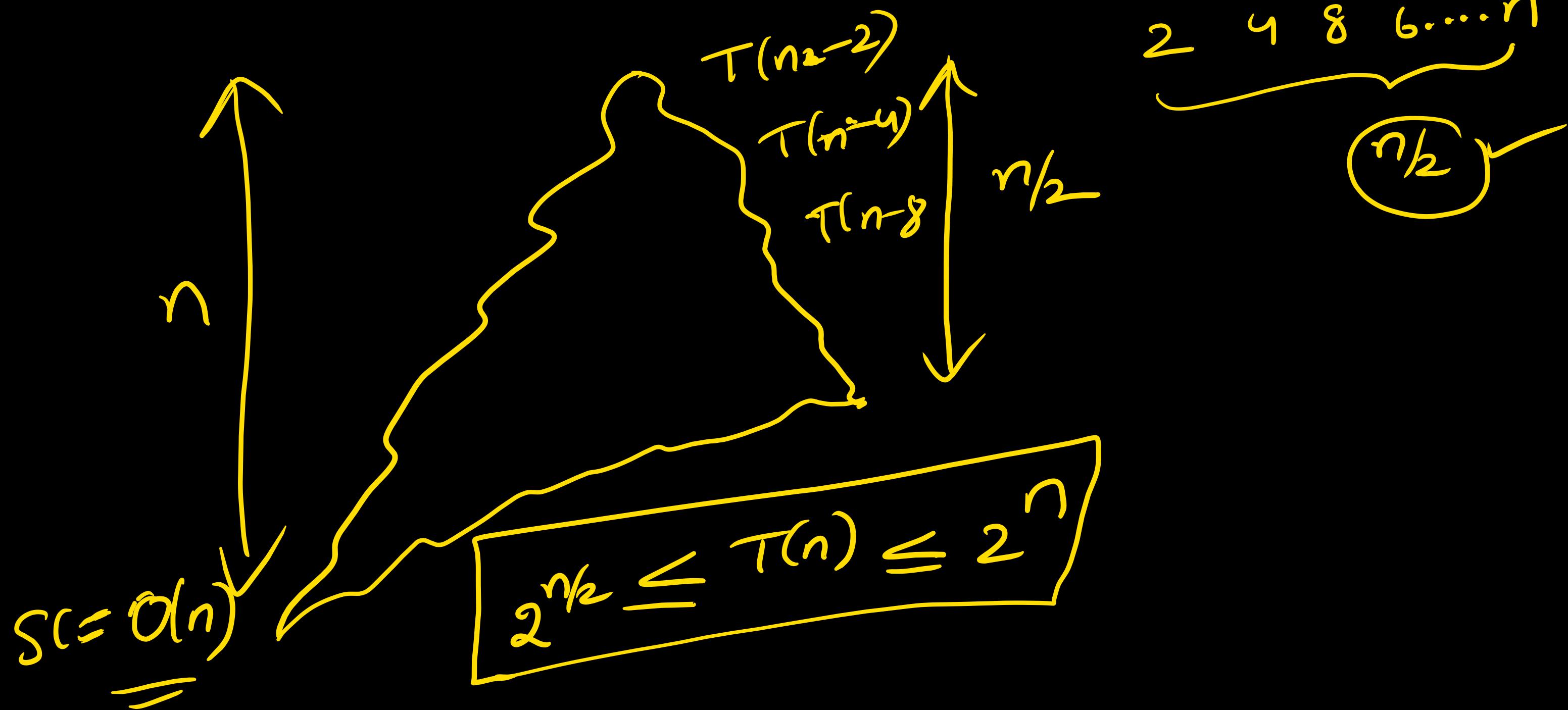
$$v(n) = \begin{cases} n & ; \text{if } n=0 \wedge n=1 \\ v(n-1) + v(n-2) & \text{otherwise.} \end{cases}$$

TC:

$$T(n) = \begin{cases} O(1) & \text{if } n=0 \text{ or } n=1 \\ T(n-1) + T(n-2) + c; \end{cases}$$

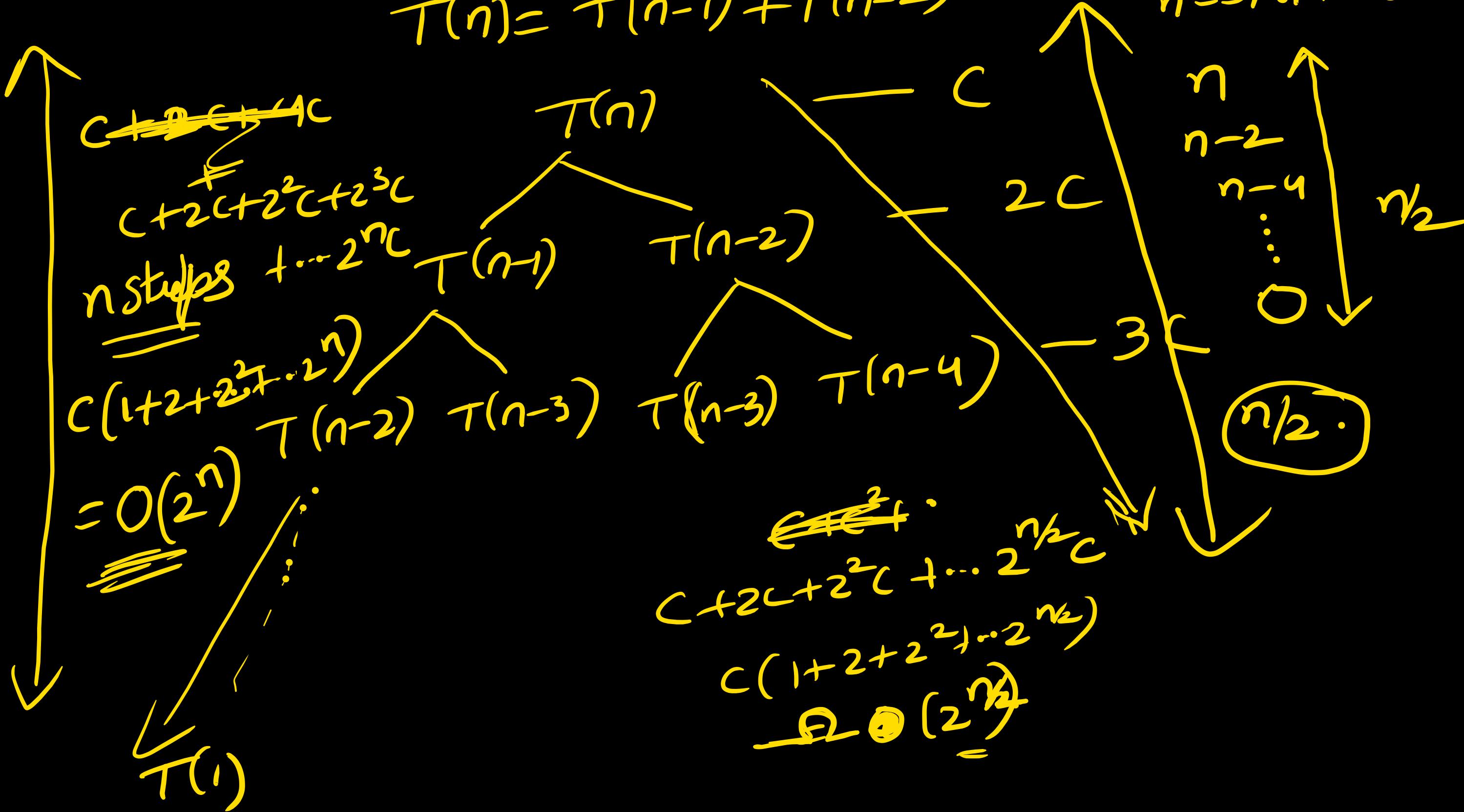


~~$\Omega(2^{n/2})$~~



$$T(n) = T(n-1) + T(n-2)$$

$$\eta = -1 \quad \Delta\eta = > 0.$$



$$T(c) = \Omega(2^{n/2})$$
$$= O(2^n)$$

~~not different by constant.~~

$n/2$

$$\Omega(2^{n/2})$$

"n"

$$O(\alpha 2^n).$$