

Q) How to find time and space complexity of
Recursive program?

✓ Iterative:

for ($i=1 \dots n$)
{ $a=b+c$ } \rightarrow directly we say TC.

i, a, b, c (4×2) Bytes.

$O(1)$ Complexity

$A(n)$

{ if $n > 0$

$(A(n-1) + A(n-2)) \Rightarrow$

}

else
return 1

γ

? TC of this prog
SC of this prog.

In order to find TC of recursive program, we use recurrence relation:

$$A(n) \Rightarrow T(n)$$

$$\{ \text{if } (n=0) \quad T(n-1) \quad T(n-2)$$

$$\{ A(n-1) + A(n-2)$$

}

else
return 1;

y

$$T(n) = \begin{cases} T(n-1) + T(n-2), & n > 0 \\ 1, & n \leq 0 \end{cases}$$

we have to solve this

methods to solve recurrence relations :-

- Substitution method
- recursive tree method
- master theorem.

$$T(n) = a \underline{T(n/b)} + O(n) \rightarrow \text{master} \rightarrow \underline{\underline{\text{fast.}}}$$

$$T(n) = \underbrace{T(n-1) + T(n-2) + T(n-3)}_{\text{more than one term}} \rightarrow \text{Recursive tree}$$

$$T(n) = \underbrace{2T(n-1) + n}_{\text{only one term}} \rightarrow \text{Substitution method}$$

In substitution method and recursion tree method, we can find space complexity also.

Substitution method: ✓

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ T(n-1) + n & \text{if } n > 1 \end{cases}$$

base condition
or
Termination condition

$$T(n) = T(n-1) + n \rightarrow ①$$

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

$$\underline{T(n-2)} = T(n-3) + n-2 \rightarrow ③$$

Substitute ② in ①

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

$$T(n) = \underline{\underline{T(n-2)}} + (n-1) + n \rightarrow \textcircled{4}$$

③ in ④

$$T(n) = \underline{\underline{T(n-3)}} + \underline{\underline{(n-2)}} + \underline{\underline{(n-1)}} + \underline{\underline{n}}$$

{ K times } \leqslant

$$T(n) = T(n-K) + (n-(K-1)) + (n-(K-2)) + \dots + \underline{\underline{(n-2)}} + \underline{\underline{(n-1)}} + \underline{\underline{n}}$$

stops @ $T(1)$

$$n-K=1 \Rightarrow K=n-1$$

$SC = O(n)$

$$T(n) = T(\underbrace{n-k}_1) + T(n-\cancel{(k-1)}) + T(n-\cancel{(k-2)}) + \dots + \cancel{(n-2)} + (n-1) + n.$$

$$\boxed{n-k=1} \checkmark$$

$$\Rightarrow k = n-1$$

$$= T(1) + T(n-\cancel{(n-1)-1}) + T(n-\cancel{(n-1)-2}) + \dots + n-2 + n-1 + n$$

$$= \cancel{1} + \cancel{2} + \cancel{3} + 4 + \dots + \cancel{n-2} + \cancel{n-1} + \cancel{n}.$$

$$= \cancel{\Theta(n^2)} \quad \boxed{\frac{n(n+1)}{2}} = O(n^2) \checkmark$$
$$= \Omega(n^2) \checkmark$$
$$= \Theta(n^2) \checkmark$$

$$T(n) = T(n-1) * n \rightarrow ①$$

$$T(n-1) = T(n-2) * (n-1) \rightarrow ②$$

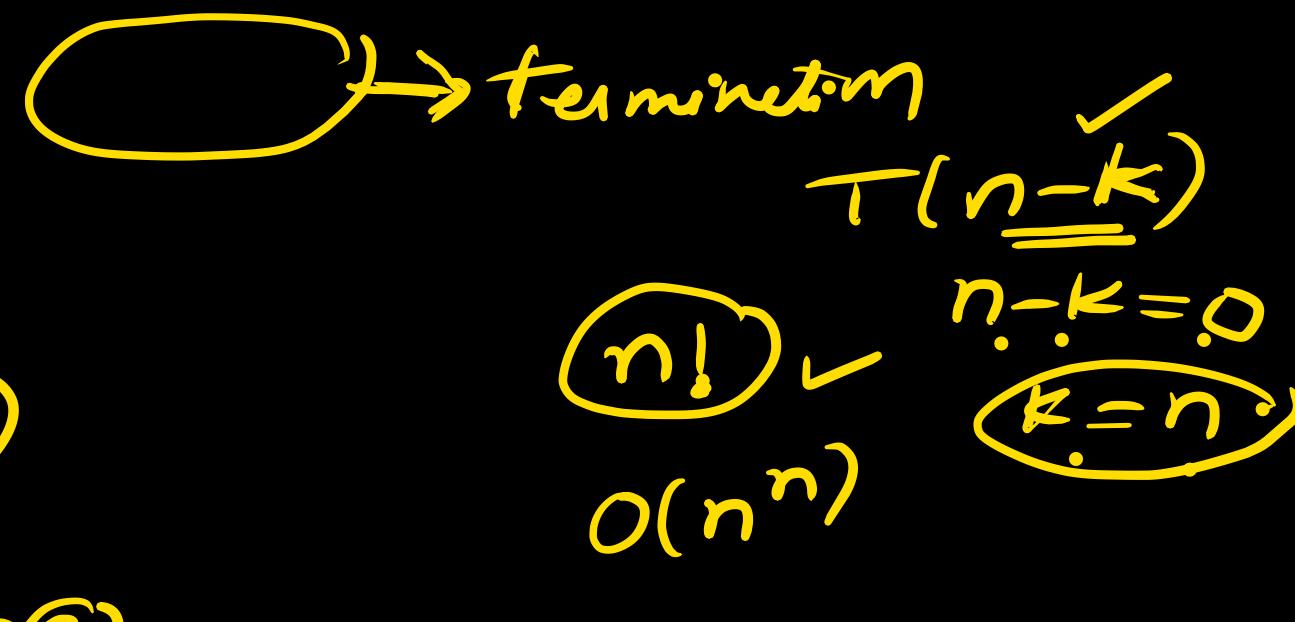
$$T(n-2) = T(n-3) * (n-2) \rightarrow ③$$

② in ①

$$T(n) = T(n-2) * n-1 * n \rightarrow ④$$

③ in ④

$$T(n) = T(n-3) * (n-2) * n-1 * n$$



$$\tilde{T}(n) = \tilde{T}(n-3) * (n-2) * n-1 * n$$

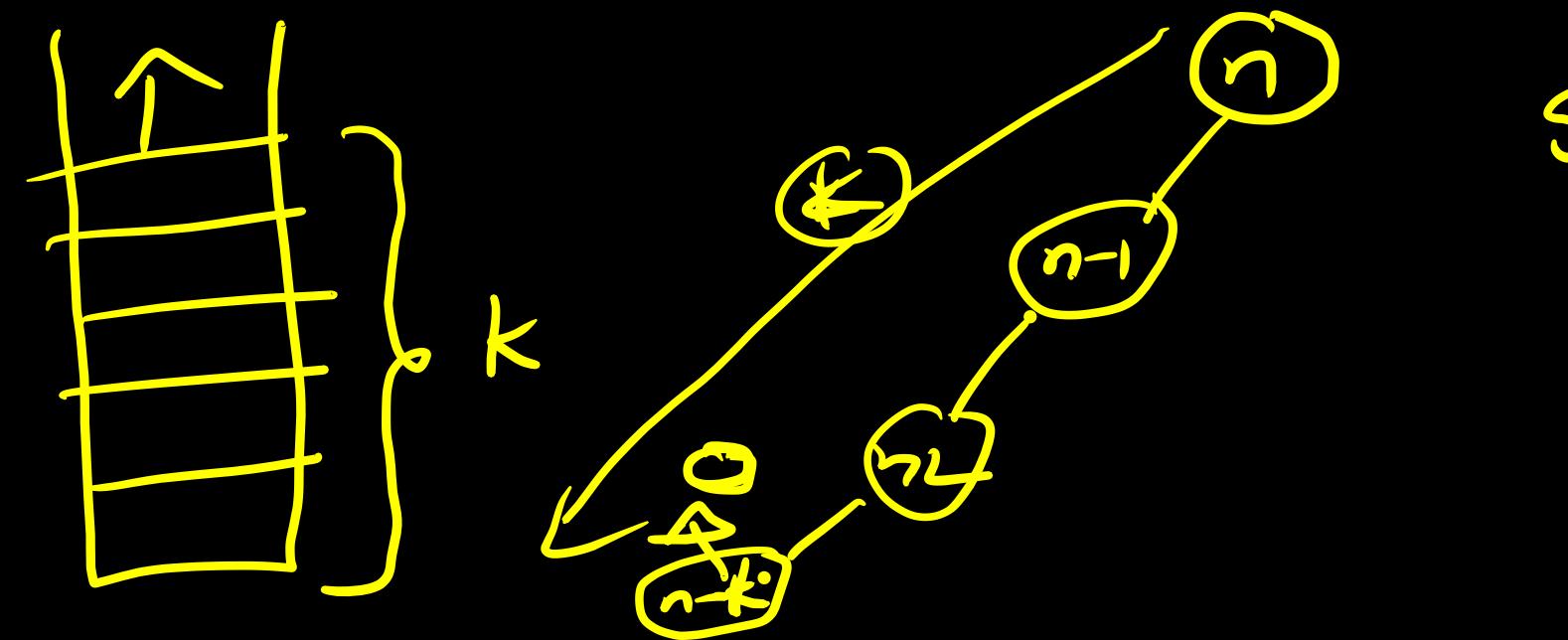
$\underbrace{\qquad\qquad}_{k \text{ time}}$

$$T(n) = T(n-k) * \cancel{T(n-(k-1))} * (n-(k-2)) * \dots * (n-1) * n.$$

$\underbrace{\qquad\qquad}_{k}$

$k \leq n-k \Rightarrow k = \underline{\underline{n}}$

whatever the value of k is, it is space complete.



$$SC = O(n)$$

$$\tau(n) = \tau\left(\frac{n}{n-k}\right) * \cancel{\tau(n-(k-1))} * (n-(k-2)) * \dots * (n-1) * n.$$

$$= \tau(0) * (n-(n-1)) * (n-(n-2)) * \dots * (n-1)(n)$$

$$= 1 * 1 * 2 * 3 * \dots * (n-1)(n)$$

$$= n!$$

$$= O(n!)$$

$$= \Omega(n!)$$

$$= \Theta(n!)$$

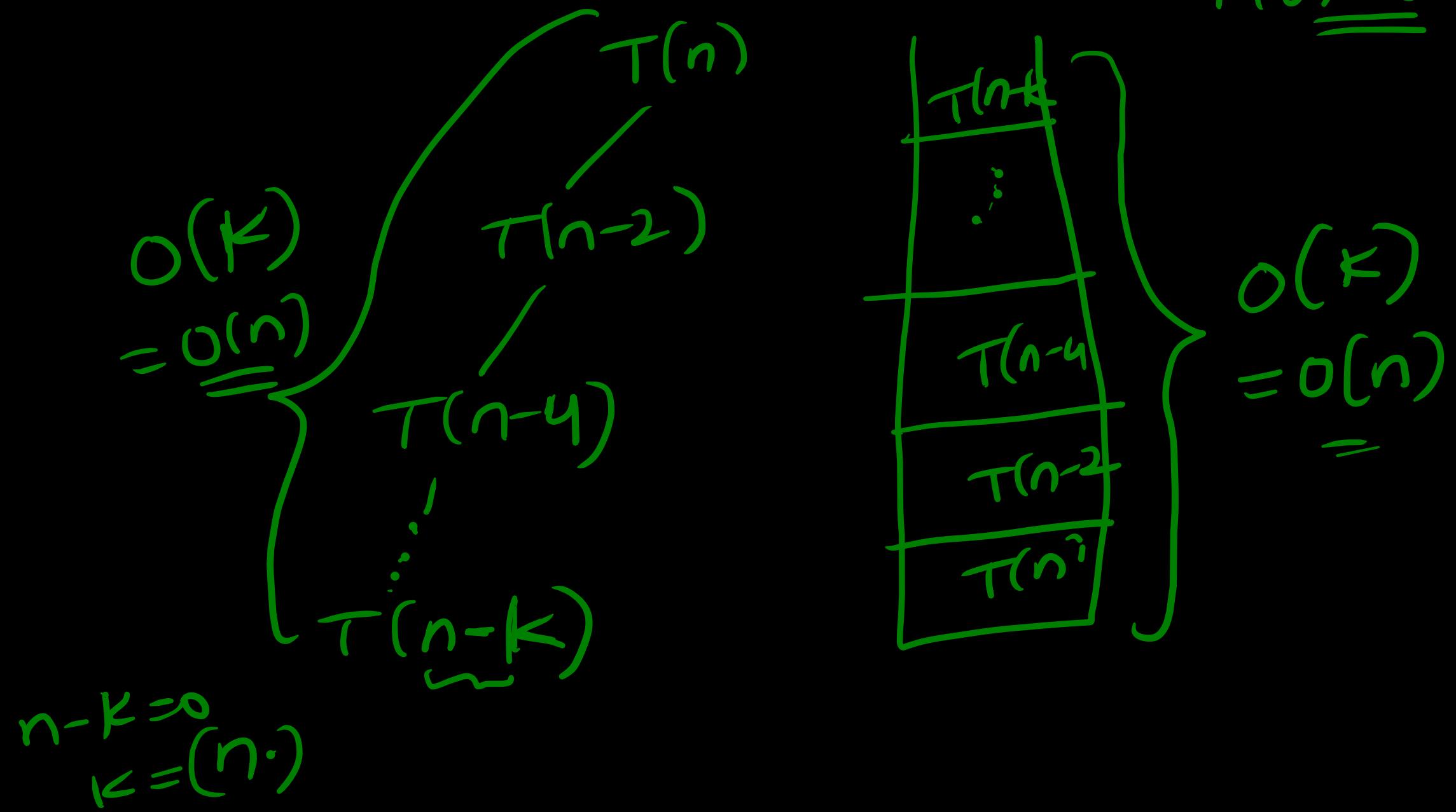
$$O(n^n)$$

$$O(n^n)$$

$$T(n) = \begin{cases} 0 & \text{if } n=0 \\ T(n-2) + n^2 & \text{if } n>0 \end{cases}$$

if $n=0$ $\rightarrow T(\underbrace{n-k})$ ✓
 but assume $n-k=0$

$$T(0) \equiv 0$$



$$T(n) = T(n-2) + n^2 \rightarrow ①$$

3 equations

$$T(n-2) = T(n-4) + (n-2)^2 \rightarrow ②$$

$$T(n-4) = T(n-6) + (n-4)^2 \rightarrow ③$$

② in ①

$$T(n) = T(n-4) + (n-2)^2 + n^2 \rightarrow ④$$

③ in ④

$$T(n) = T(n-6) + \underbrace{(n-4)^2}_{\{ K \text{ times} \}} + \underbrace{(n-2)^2}_{2K-2} + n^2$$

$$T(n) = \boxed{T(n-2K)} + (n-(2K-2))^2 + \dots + (n-2)^2 + n^2$$

Let $\boxed{n-2K}=0$
 $n=2K$; $K=n/2$; $Sc = O(n)$.

$$\begin{aligned} T(n) &= T(0) + 2^2 + 4^2 + 6^2 + 8^2 + \dots + n^2 \\ &= \cancel{T(0)} (2 \times 1)^2 + (2 \times 2)^2 + (2 \times 3)^2 + \dots + (2 \times \frac{n}{2})^2 \\ &= 2^2 (1^2 + 2^2 + 3^2 + \dots + \frac{n^2}{4}) \end{aligned}$$

$$= 2^2 \left(1^2 + 2^2 + 3^2 + \dots + \left(\frac{n}{2}\right)^2 \right)$$

$$= 2^2 \left(\frac{\frac{n}{2} (2\frac{n}{2} + 1) (\frac{2n}{2} + 1)}{6} \right)$$

$$1^2 + 2^2 + 3^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\left. \begin{array}{l} TC \\ = O(n^3) \\ = \Omega(n^3) \\ = \Theta(n^3) \end{array} \right\} \checkmark \quad SC = O(n)$$

$$T(n) = \begin{cases} 0 & \text{if } n=0 \\ T(n-2) + \log_2 n & \end{cases}$$

~~$T(n-2)$~~ $\xrightarrow{n-2k=0}$
 ~~$n-2$~~ $\xrightarrow{n=2k \wedge k=\frac{n}{2}}$ $\Theta SC = O(n)$

$$T(\overset{\uparrow}{n}) = \overset{\uparrow}{T(n-2)} + \log_2 \overset{\uparrow}{n} \xrightarrow{\textcircled{1}}$$

$$T(n-2) = \overset{\uparrow}{T(n-4)} + \log_2 \overset{\uparrow}{(n-2)} \xrightarrow{\textcircled{2}}$$

$$T(n-4) = T(n-6) + \log_2 \overset{\uparrow}{(n-4)} \xrightarrow{\textcircled{3}}$$

② in ①

$$T(n) = T(n-4) + \log_2 n - 2 + \log_2 n \rightarrow ④$$

$$\textcircled{3} \quad \dots \quad \textcircled{4}$$

$$= T(n-6) + \log_2(n-4) + \log_2(n-2) + \log_2 n$$

$\brace{k\text{-tak my}}$

$$= \cancel{T(n-2k)} + \log_2(n-(2k-2)) + \log_2(n-(2k-4))$$

$$+ \dots + \log_2 n - 2 + \log_2 n$$

n
 |
 $n-2$
 |
 $n-4$
 |
 $n-6$
 .
 $\textcircled{n-2k}$

$$n-2k=0$$

$$= \log_2 2 + \log_2 4 + \log_2 6 + \dots + \log_2 n$$

$$= \log_2 + \log 4 + \log 6 + \dots + \log_2 n \quad n! = O(n^n)$$

$$= \cancel{\log_2 * 1} + \cancel{\log_2 * 2} + \cancel{\log_2 * 3} + \dots + \log_2 * n/2$$

$$= \underbrace{\log_2 + \log_1}_{\log a * b} + \underbrace{\log_2 + \log_2}_{\log a + \log b} + \log_2 + \log 3 + \dots + \log_2 + \log n/2$$

$$= \cancel{\log_2}(n/2) + (\log_1 + \log_2 + \log_3 + \dots + \log_{n/2})$$

$$= n/2 + \log 1 \cdot 2 \cdot 3 \cdot \dots \cdot n/2 \quad \cancel{O(n \log n)} \quad O(n \log n)$$

$$= \cancel{n/2} + \cancel{\log(n/2)!} = O(\log(n/2)!) = O(\log n^n) = O(n \log n)$$

$$T(n) = \begin{cases} 1 & n=1 \\ T(n/2) + c & \text{otherwise} \end{cases}$$

Easy in masters theorem method.

But we want to practice in
substitution method just for practice.

So in exam/inter we masters them
for this example.

$$T(n) = T(\frac{n}{2}) + C \rightarrow ①$$

$$T(\frac{n}{2}) = T(\frac{n}{4}) + C \rightarrow ②$$

$$T(\frac{n}{4}) = T(\frac{n}{8}) + C \rightarrow ③$$

② in ①

$$T(n) = T(\frac{n}{4}) + C + C = T(\frac{n}{4}) + 2C \rightarrow ④$$

③ in ④

$$T(n) = T(\frac{n}{8}) + 3C$$

$$T(n) = T(n/8) + 3C$$

$\brace{ \text{ktwoy} }$

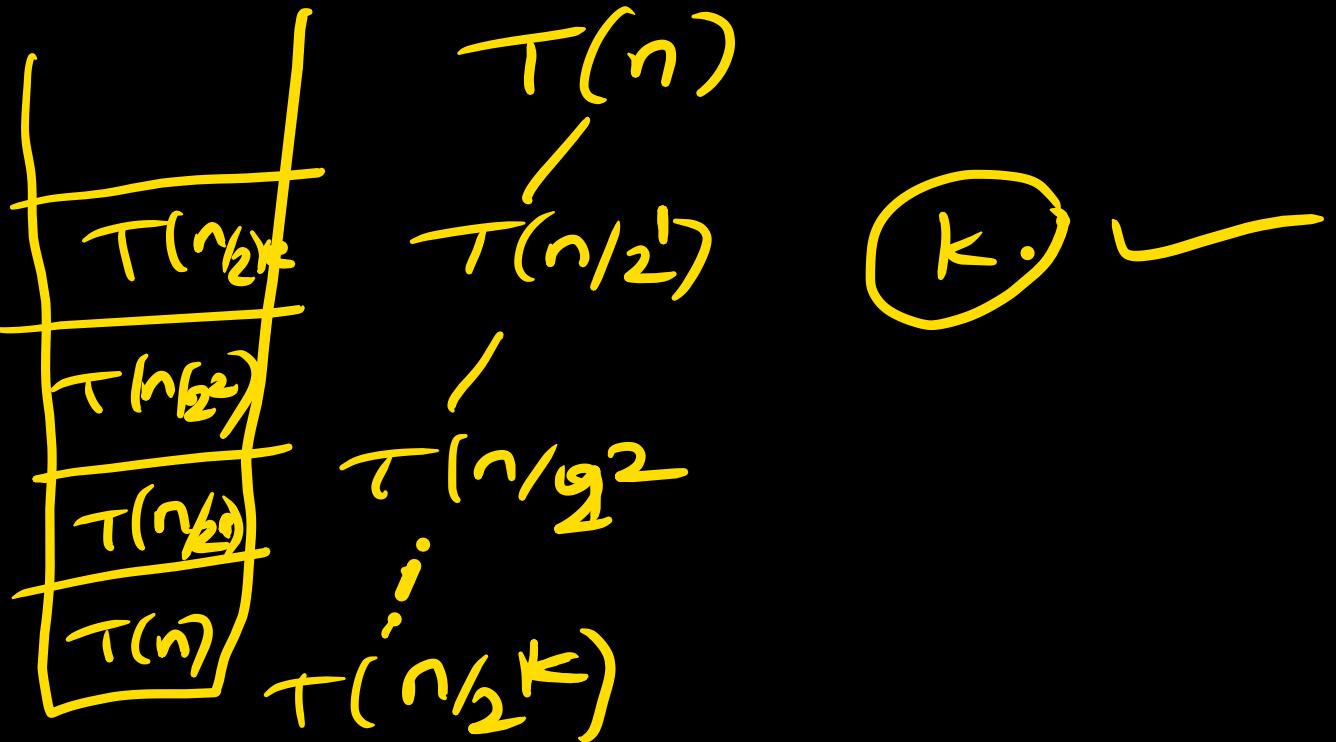
$$\pi(n) = \tau\left(\frac{n}{2^k}\right) + kc$$

Let $n/2^k = 1$

$$n = 2^k ; k = \log_2 n \quad SC = O(\log n)$$

$$T(n) = T(1) + \log_2 n \cdot C$$

$$\therefore T(n) = \underline{\Theta}(\overline{\log_2 n}) \quad \underline{\Omega}(\overline{\log_2 n}) \quad \underline{\Theta}(\overline{\log_2 n})$$



$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ T(n/2) + n & \text{if } n>1 \end{cases}$$

~~go for~~ for matters them in exam

Here we do it only for plastic.

$$T(n) = T(\frac{n}{2}) + n \rightarrow ①$$

$n/2$ $n/2$ $n/2$

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

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

Subs ② in ①

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

③ in ④

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

$$\tilde{\tau}(n) = \tau(n/8) + n/4 + n/2 + n$$

$$= \underbrace{\tau\left(\frac{n}{2^3}\right) + \frac{n}{2^2} + \frac{n}{2^1} + \frac{n}{2^0}}$$

$\underbrace{\qquad}_{k \text{ terms}}$

$$= \tau\left(\frac{n}{2^k}\right) + \frac{n}{2^{k-1}} + \frac{n}{2^{k-2}} + \dots + \frac{n}{2^2} + \frac{n}{2^1} + \frac{n}{2^0}$$

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

$$n = 2^k$$

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

$$\frac{n}{2^{k-1}} = \frac{n}{2^k \times 2^{-1}} = \left(\frac{n}{2^k}\right) \times 2 = 1 \times 2$$

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

$$n = 2^k$$

$$\frac{n}{2^{k-2}} = \frac{n}{2^k \times 2^{-2}} = \left(\frac{n}{2^k}\right) \times 2^2 = 2^2$$

$$= 1 + 2^1 + 2^2 + \dots + 2^K$$

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

$$n = 2^K$$

$$K = \log n$$

$$= 2^0 + 2^1 + 2^2 + \dots + 2^{\log_2 n}$$

$$= \frac{2^0(2^{\log_2 n + 1} - 1)}{2 - 1} = 2^{\log_2 n}$$

$$= n = O(n) \Theta(n) \Omega(n)$$

$$\frac{a(r^{n-1})}{r-1}$$

$$r > 1$$

$$T(n) = \begin{cases} 1 \\ 2T(n/2) + n \end{cases}$$

In interview / exam go with mastery

Here we just practice.

$$T(n) = 2T\left(\frac{n}{2}\right) + n \rightarrow ①$$

$$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + \frac{n}{2} \rightarrow ②$$

$$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + \frac{n}{4} \rightarrow ③$$

Subs ② in ①

$$\begin{aligned} T(n) &= 2 \left(2T\left(\frac{n}{4}\right) + \frac{n}{2} \right) + n \\ &= 2^2 T\left(\frac{n}{2^2}\right) + 2n \rightarrow ④ \end{aligned}$$

$$\begin{aligned} T(n) &= 2 \left(2T\left(\frac{n}{4}\right) + \frac{n}{2} \right) + n \\ &= 2^2 T\left(\frac{n}{2^2}\right) + 2n \rightarrow \textcircled{4} \end{aligned}$$

Subs $\textcircled{3}$ in $\textcircled{4}$

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

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

$$\underbrace{2^k T\left(\frac{n}{2^k}\right) + kn}_{\Downarrow n/2^k = 1} = \cancel{2^k} 2^k + kn$$

$$\begin{aligned} &= 2^{\log_2 n} + kn \\ &= n + (n \log n) \end{aligned}$$

$$a^{\log_c b} = b^{\log_c a}$$

$$2^{\log_2 n} \Rightarrow n^{\log_2 2}$$

$$\begin{aligned} &= O(n \log n) \\ &= \Omega(n \log n) \quad \Theta(n \log n) \end{aligned}$$