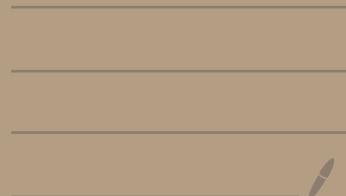


practice link

<https://www.codingninjas.com/codestudio/guided-paths/competitive-programming/content/126222/offering/1476042>



Time & Space Complexity



→ Linear Search :-



bool isPresent (int *arr, int n, int value)

{

for (int i=0; i<n; i++)

{

if (arr[i] == value)
return true;

}

down false;

}

T.C $\rightarrow O(n)$
 $n \rightarrow \underline{\text{arr size}}$

S.C $\rightarrow O(1)$

→ Recursion :-

Fac

Power / Counting

Factorial

T.C

time required
as function of input
 $f(n)$

int fact (int n)

i

// Basic Case

if ($n == 0$)

return 1;

K_1

T.C → ?

// Rec. Call

return $n * fact(n-1);$

$\downarrow K_2$
 $\downarrow T(n-1)$

① Try \rightarrow Eg \rightarrow Recurrence

$$f(n) = n \star f(n-1)$$

$$\underline{T(n)} = k_1 + k_2 + T(n-1)$$

$$\begin{aligned} T(n) &= k + T(n-1) \\ T(n-1) &= k + T(n-2) \\ T(n-2) &= k + T(n-3) \end{aligned}$$

n times

$$\cancel{T(1)} = \cancel{K} + \cancel{T(0)}$$

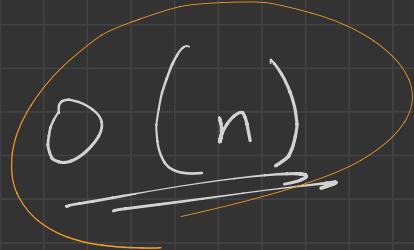
$$\cancel{T(0)} = \underline{K_1}$$

$$T(n) = n \star K + K_1$$

$$T(n) = \cancel{Kn} + \underline{\cancel{K_1}} \cancel{= K}$$

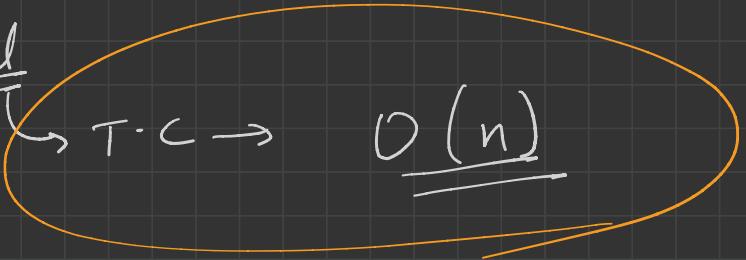
$$T(n) = \underline{Kn}$$

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

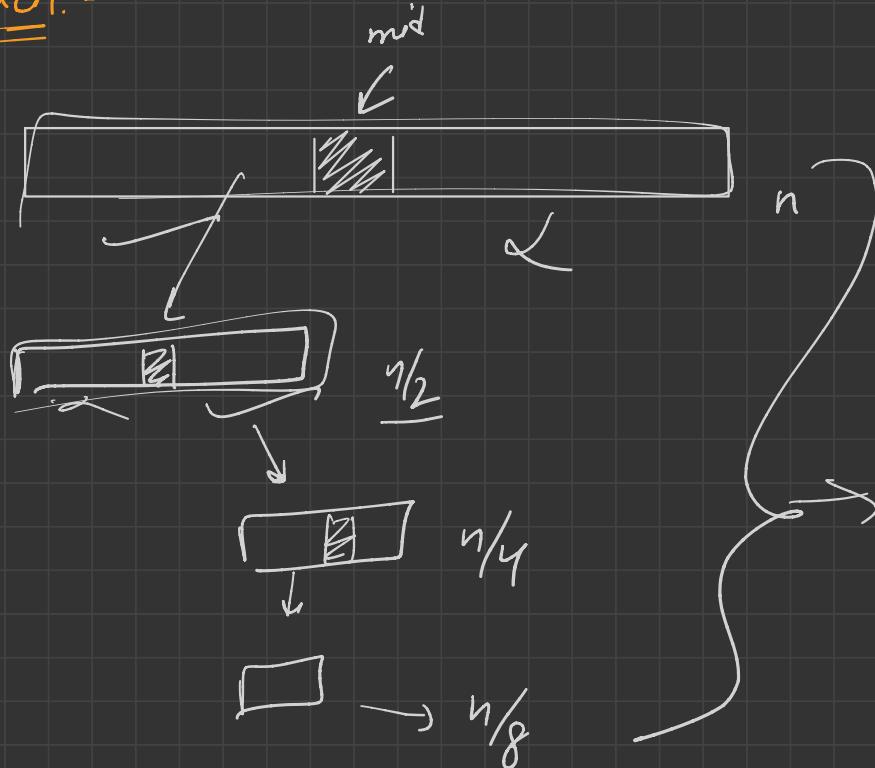

$$\underline{\underline{O(n)}}$$

Factorial

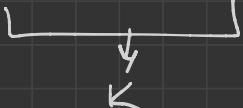
$$T.C \rightarrow$$


$$\underline{\underline{O(n)}}$$

Binary Search:-



$$f(n) = \cancel{xyz} + f(n/2)$$

$$T(n) = K_1 + K_2 + T(n/2)$$


$$\underline{T(n)} = \underline{K} + T(n/2)$$

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

$$T\left(\frac{n}{4}\right) = \underline{K} + T\left(\frac{n}{8}\right)$$

1
|
|

1
|
|

a times

$$T(2) = K + T(1)$$

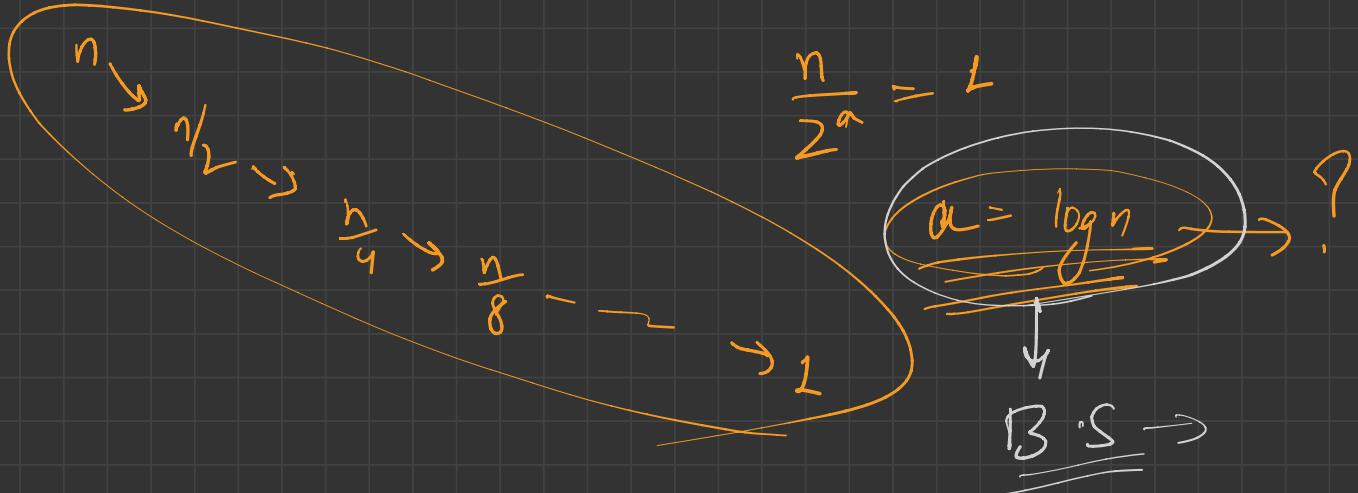
$$T(1) = K$$

$$T(n) = a \star K$$

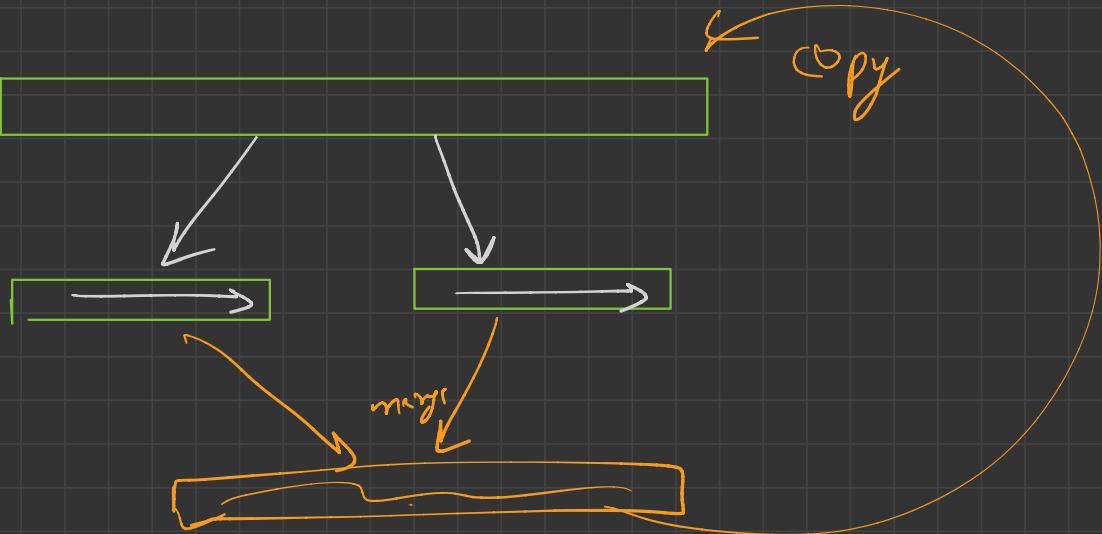
$$T(n) = K \star \log n$$

$$= \log n$$

$$\therefore T \in O(\log n)$$



→ Merge Sort



steps

break 2 array left right →

Rec-> sort (left / right)

new array → **merge** → $\frac{k_3 n}{2}$

copy new array content → original array → $k_4 n$

$$\overbrace{T(n)}^{\text{left}} = K_1 + K_2 + T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + K_3 n + K_4 n$$

left *right part*

$$T(n) = K + 2T\left(\frac{n}{2}\right) + n \underbrace{(K_3 + K_4)}_{K_5}$$

$$\overbrace{T(n)}^{\text{left}} = K + 2T\left(\frac{n}{2}\right) + n K_5$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n K_5$$

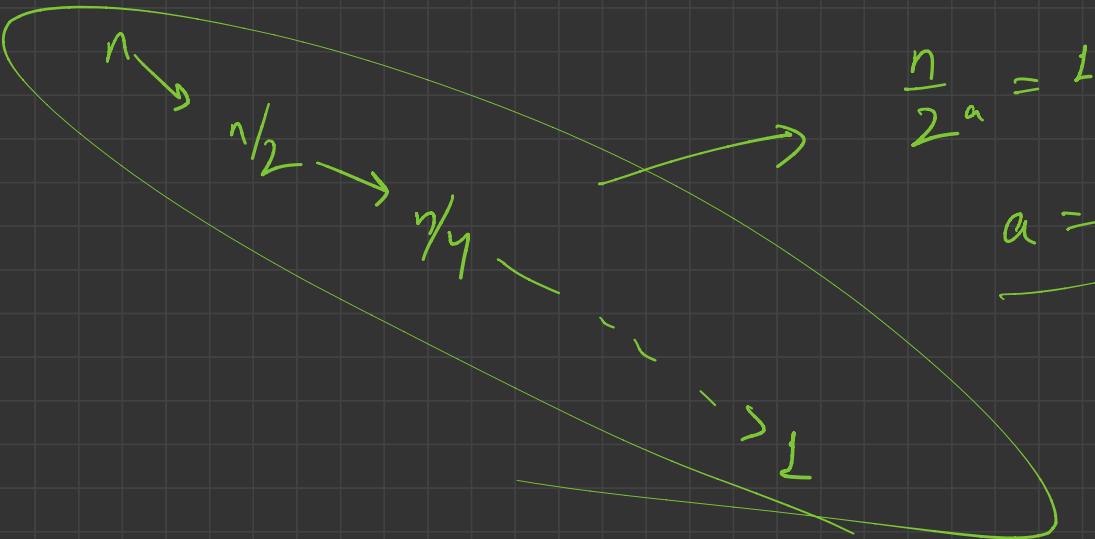
$$T(n) = \cancel{2T\left(\frac{n}{2}\right)} + nk$$

$$2 \times \cancel{T\left(\frac{n}{2}\right)} = 2T\left(\frac{n}{4}\right) + \frac{n}{2}k = \cancel{4T\left(\frac{n}{4}\right)} + nk$$

$$4 \times \cancel{T\left(\frac{n}{4}\right)} = 2T\left(\frac{n}{8}\right) + \frac{n}{4}k = \cancel{8T\left(\frac{n}{8}\right)} + nk$$

$$T(1) = k$$

$$T(n) = a * nk$$



→ Fibonacci Series

$\text{fib(int } n)$

if

$\left(n = = 0 \right) \mid \mid \left(n = = 1 \right)$

return $n;$

K_1

$T.C \rightarrow \underline{\underline{O(2^n)}}$

$\mid \mid R.C$

return $\text{fib}(n-1) + \text{fib}(n-2);$

\downarrow

K_2

3

$$\begin{aligned} T(n) &= K_1 + K_2 + T(n-1) + T(n-2) \\ &= K + T(n-1) + T(n-2) \end{aligned}$$

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

$$T(n-1) = K + T(n-2) + T(n-3)$$

$$T(n-2) = K + T(n-3) + T(n-4)$$

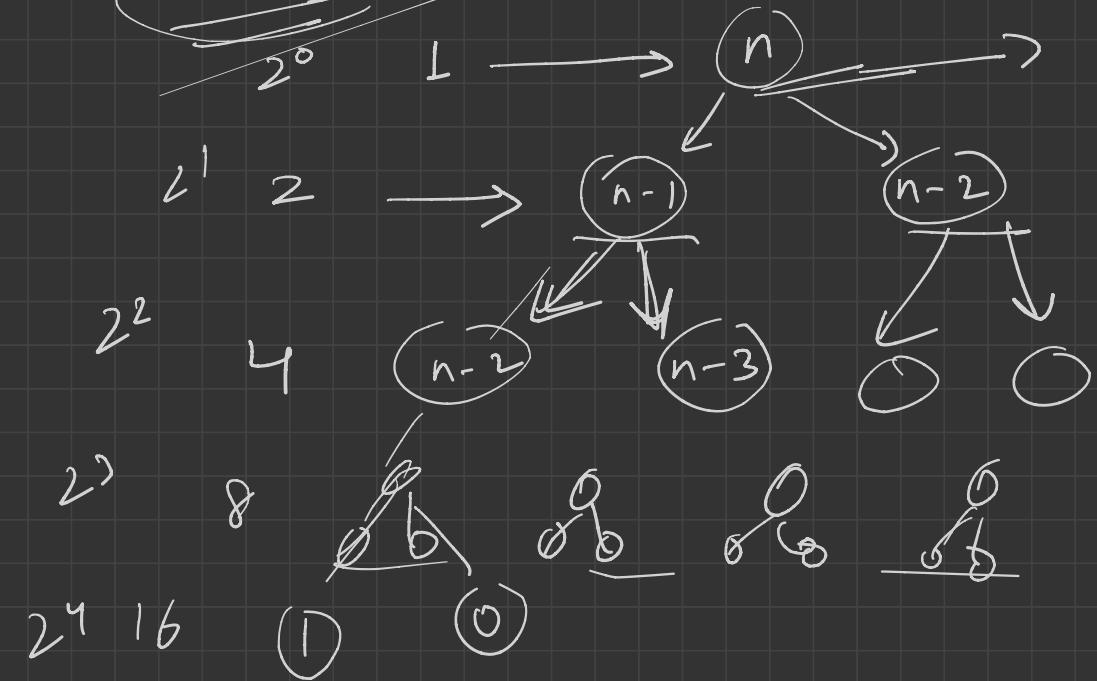
$$T(1) = K_1$$

$$T(0) = K_1$$

Solução?

H/n

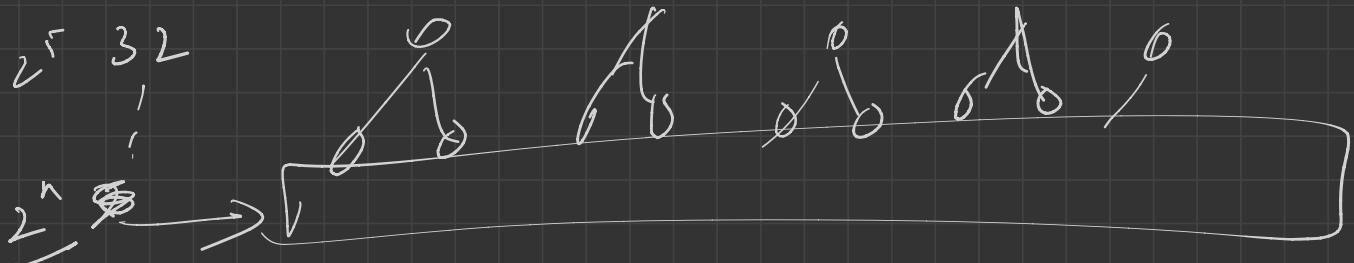
Rec Trace:-

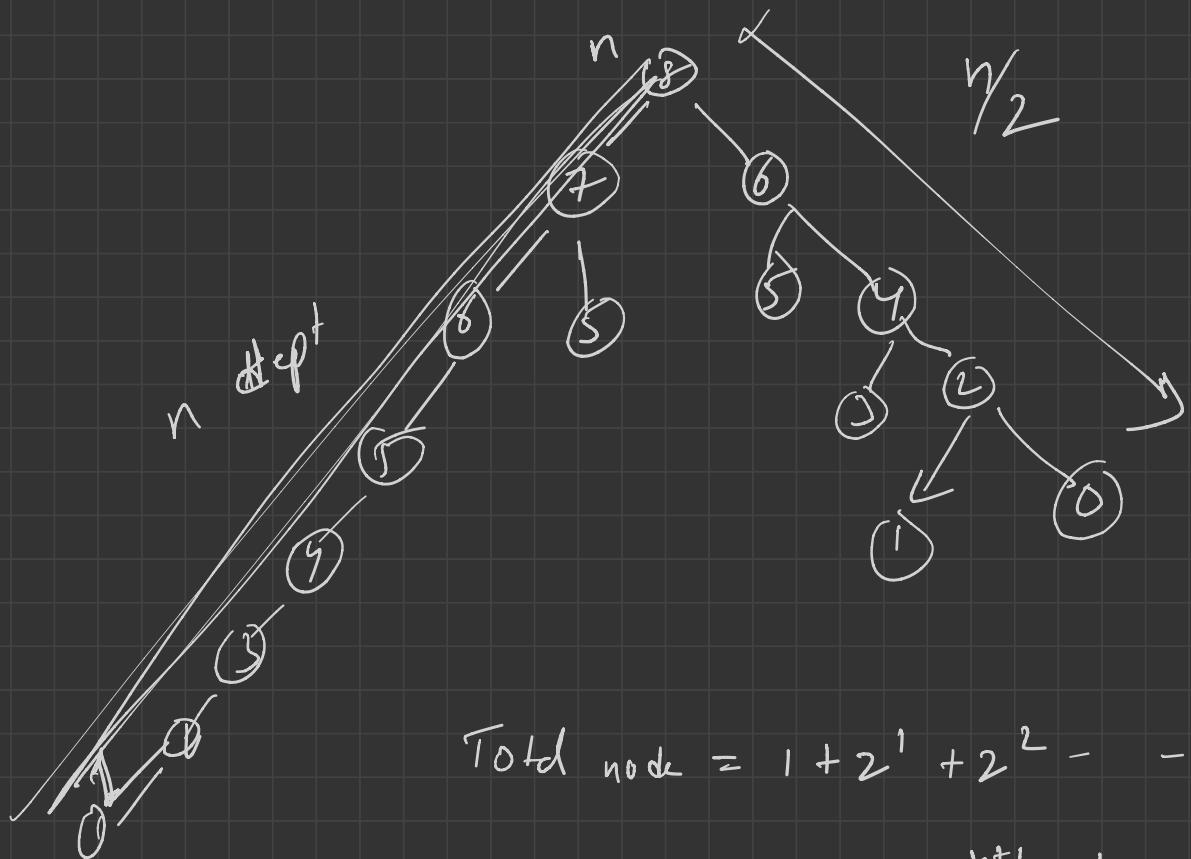


each node
 $\hookrightarrow K$ time

total time

$\hookrightarrow K * \underline{\text{total node}}$





$$\text{Total node} = 1 + 2^1 + 2^2 - \dots - 2^n$$

$$= \underline{\underline{2^{n+1} - 1}}$$

$$T \cdot C = K * \begin{pmatrix} 2^{n+1} & -1 \\ 0 & 2 \end{pmatrix}$$

$$= K \times 2^{n+1}$$

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

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

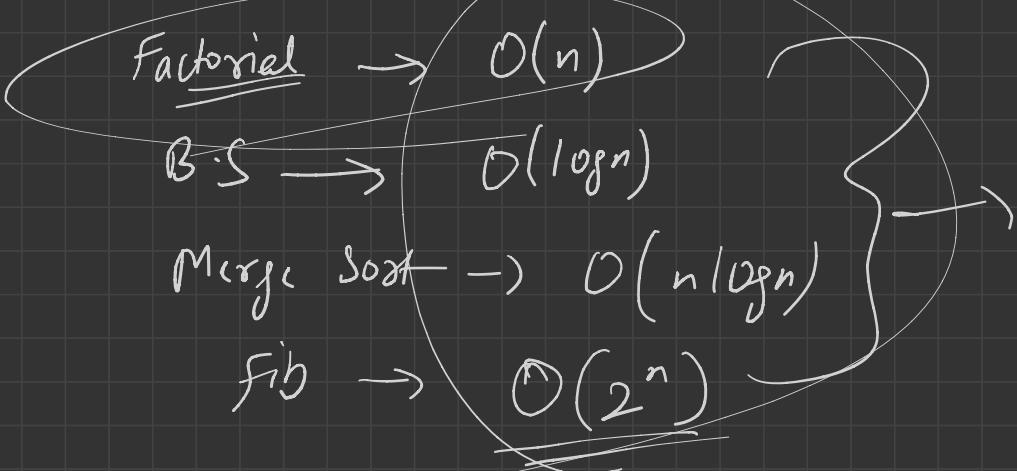
$$= 2^n$$

$$\rightarrow O(2^n)$$

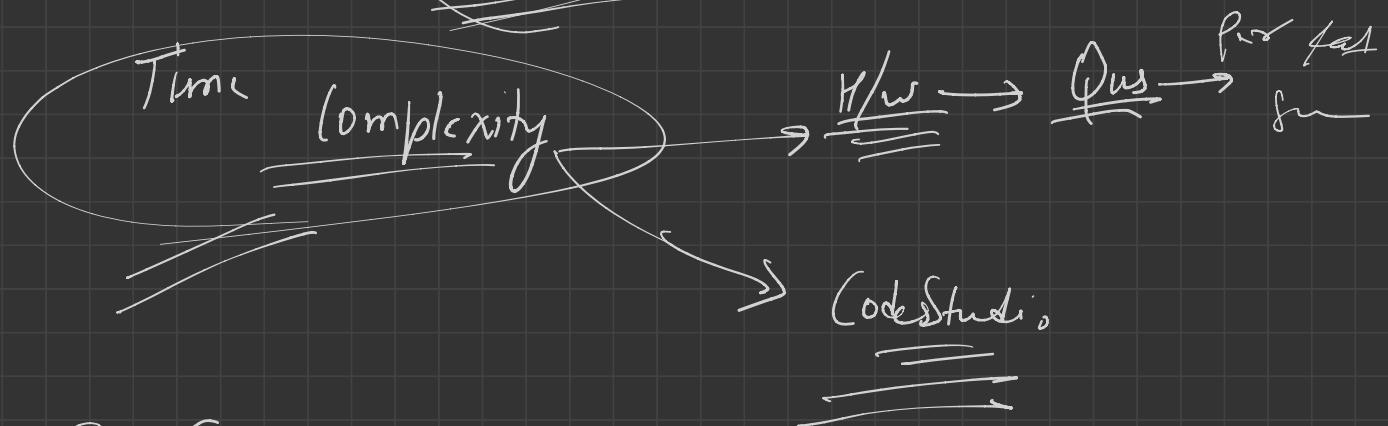
Fib

T-C

exponential



Rec Algo



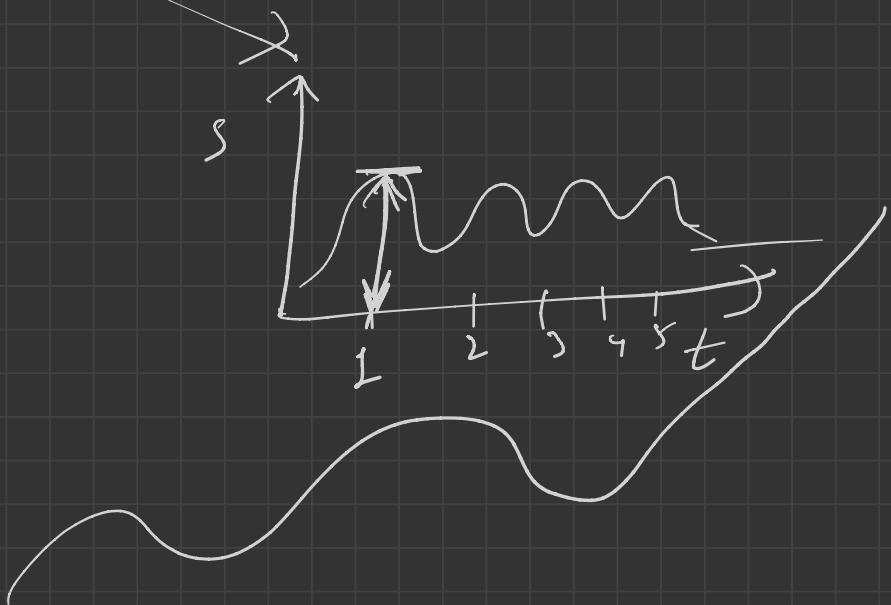
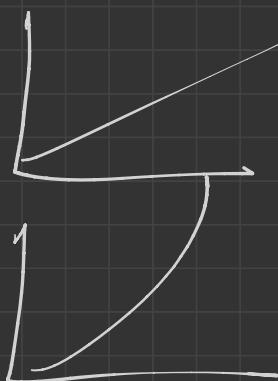
S.C.

Space complexity :- space required

as $\underline{f(n)}$

T.C

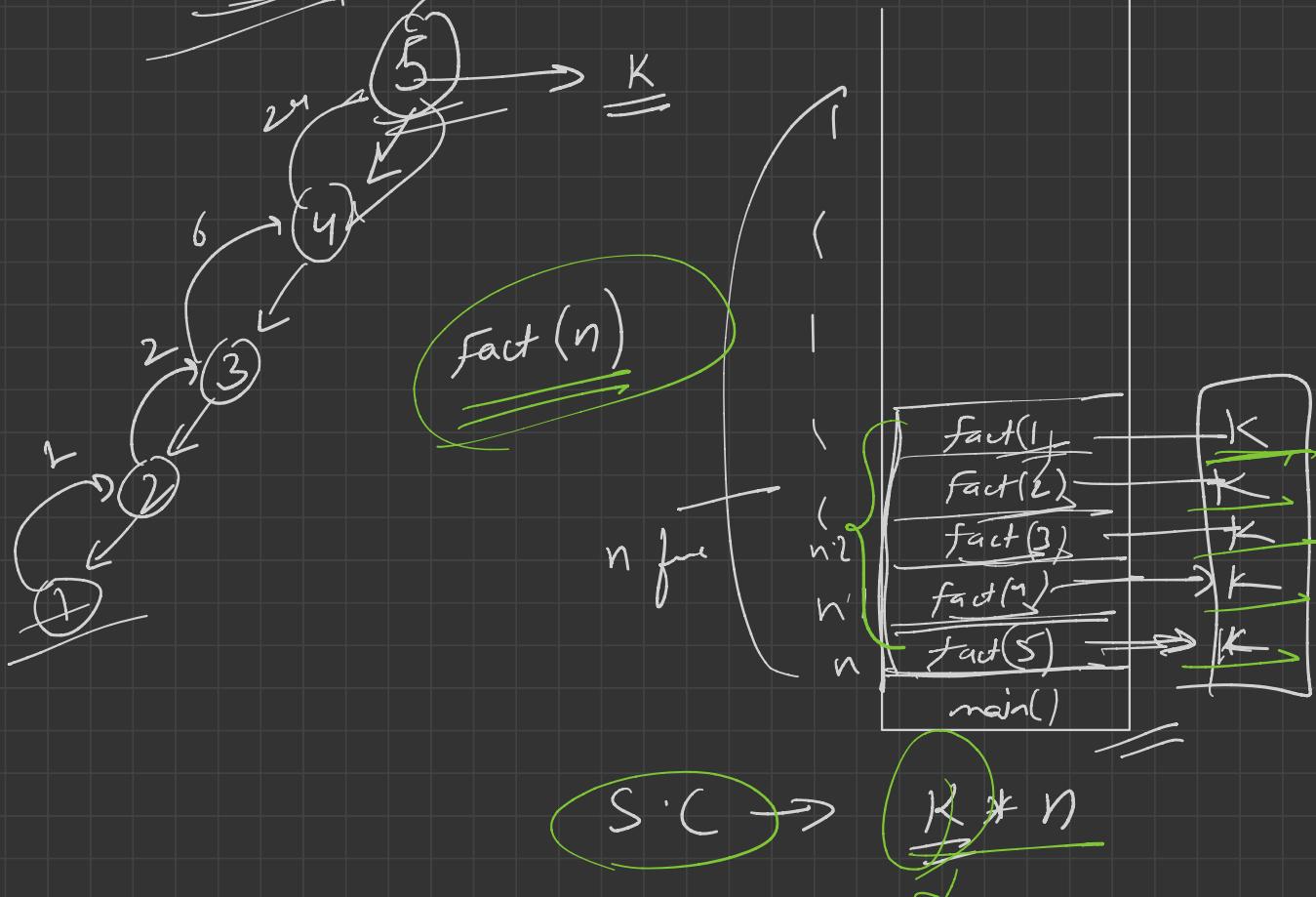
max. space
reqd. at any
instant



1

$$\cancel{\text{factorial}} : - \rightarrow 120$$

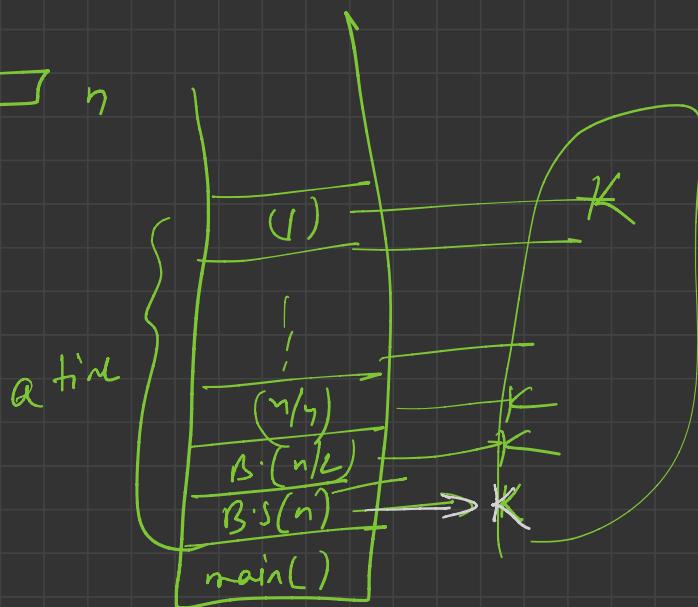
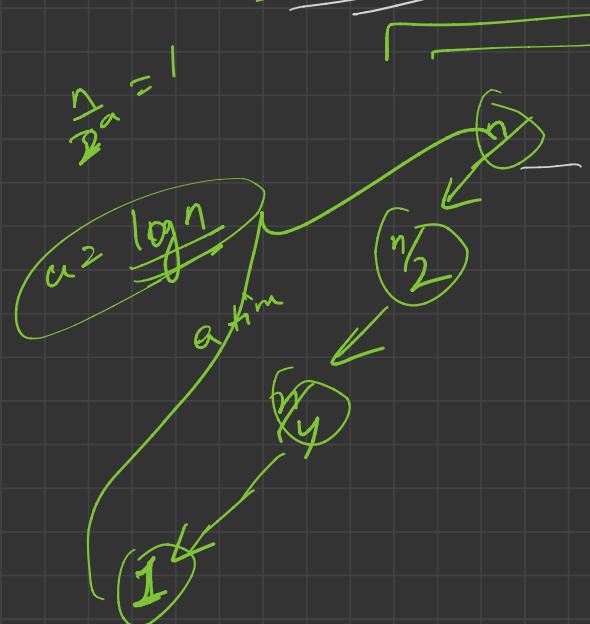
$$5 \rightarrow fad$$



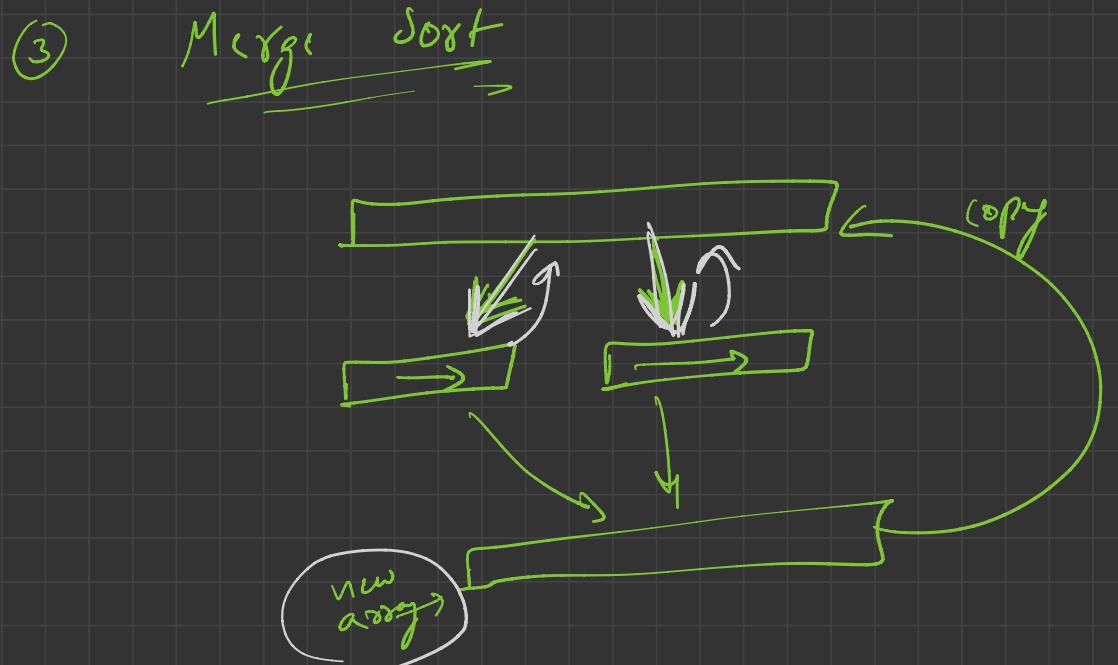
factorial

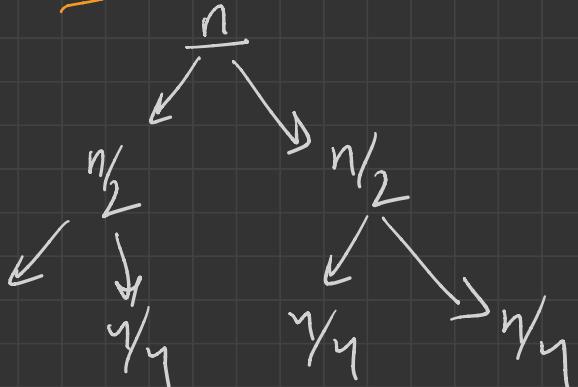
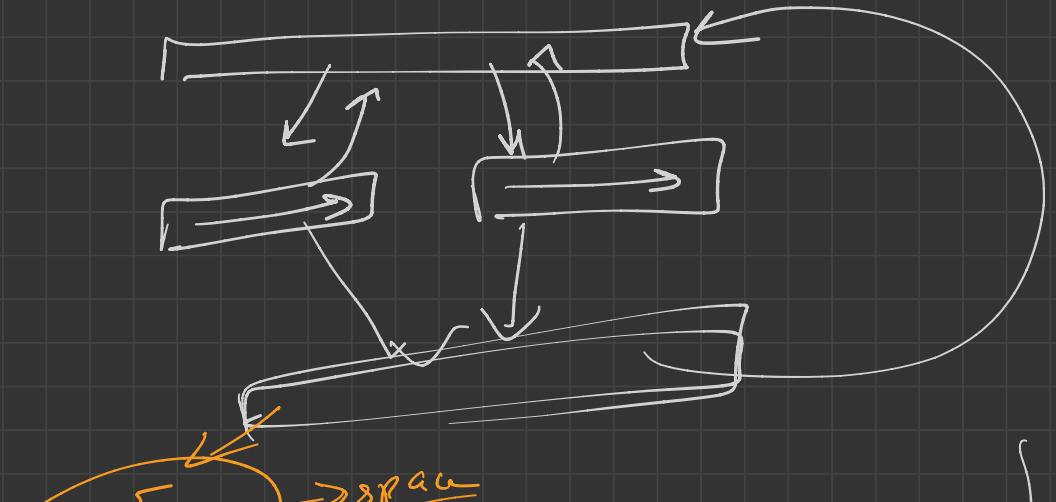
$\hookrightarrow O(n)$

Binary Search:-

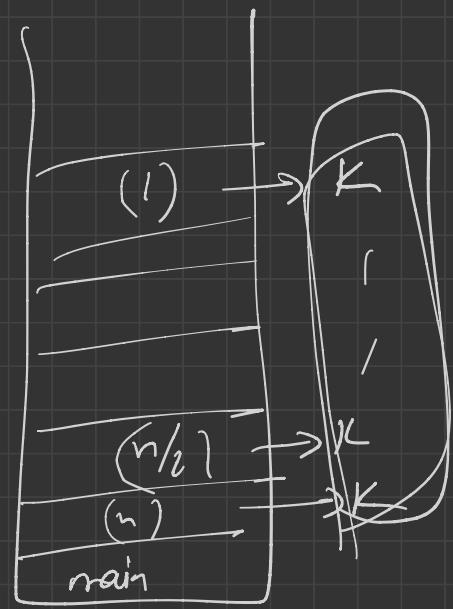


$$S \cdot C \rightarrow \frac{K}{2} * \log n$$
$$= \underline{\underline{O(\log n)}}$$





γ_γ





$$S.C \rightarrow K \cancel{\log n} + \underline{n}$$

$$\underline{n \gg \log n}$$

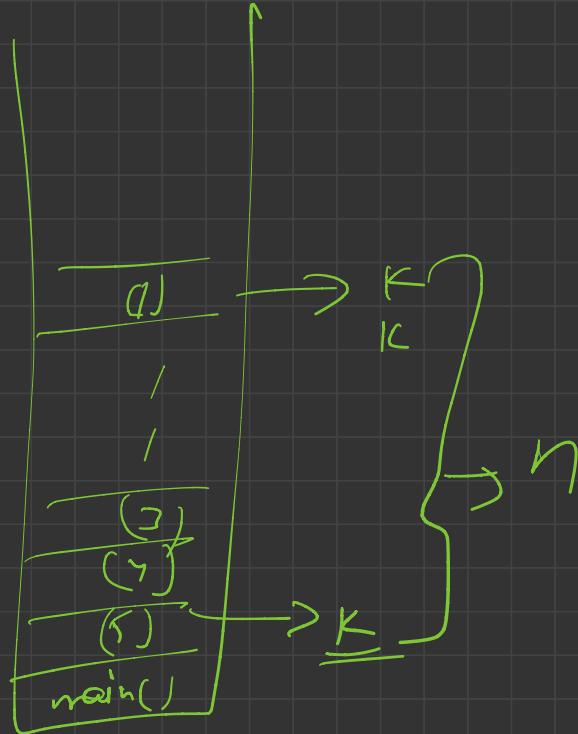
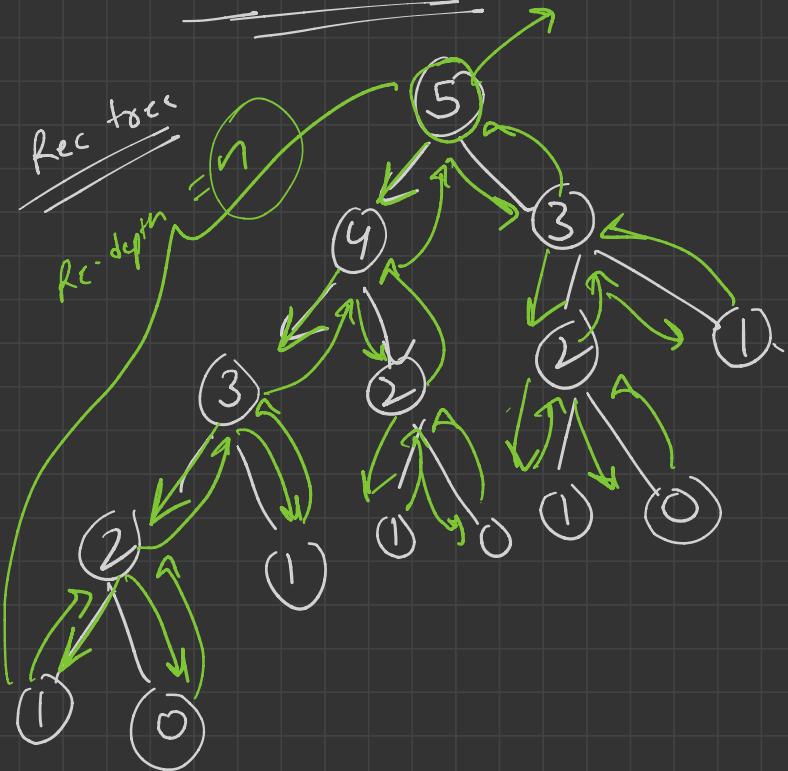
$$S.C \rightarrow O(n)$$

RCC: Merge sort

$$n = \underline{1024}$$

$$\log_2 \frac{1024}{10}$$

Fib Series:-



$S \cdot C \rightarrow n \times K$

