GATE

CRASH COURSE

DS & AI

Algorithms

Miscellaneous
(Lecture 13)

By - Aditya sir

Topics to be Covered

1.
2.
3. Recurrence Tree for TC.
4. Problem Solving

About Aditya Jain sir

1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters program
5. Joined IIT Bombay for my 2 year Masters program, specialization in Data Science
6. Published multiple research papers in well known conferences along with the team
7. Received the prestigious excellence in Research award from IIT Bombay for my Masters thesis
8. Completed my Masters with an overall GPA of 9.36/10
9. Joined Dream11 as a Data Scientist
10. Have mentored working professions in field of Data Science and Analytics
11. Have been mentoring GATE aspirants to secure a great rank in limited time
12. Have got around 27.5K followers on Linkedin where I share my insights and guide students and professionals.

@ADITYASIR_PW

Telegram Link for Aditya Jain sir:

https://t.me/AdityaSir_PW

Time Complexity Recurrence :

1) Back - substitution $\longrightarrow$ value of Recurrence + TC

2) Master's method $\longrightarrow$ TC

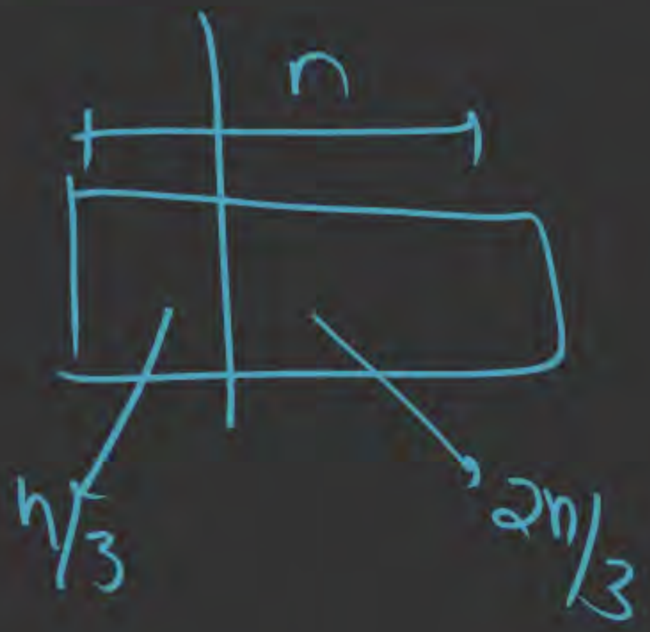
3) Recurrence Tree $\longrightarrow$ TC

Types of Recurrence :

1) Symmetric : $T(n) = a * T(n/b) + f(n)$, $a \geq 1, b > 1$, $f(n) = +ve$

→ Master's Mtd.

2) Asymmetric : $T(n) = T(\alpha n) + T((1-\alpha)n) + f(n)$



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

→ Recurrence Tree

eg1) $T(n) = 2T(n/2) + n$

Part1) Value of Recurrence?

$$T(n) = 2T(n/2) + n \quad — \quad 1$$

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

$$T(n) = 2\left(2T(n/2^2) + n/2\right) + n$$

$$T(n) = 2^2 T(n/2^2) + 2n \quad — \quad 2$$

$$T(n) = 2^k T(n/2^k) + k * n \quad — \quad 3$$

$$n/2^k = 1 \qquad \underline{2^k = n}$$

$$\underline{k = \log_2 n}$$

$$T(n) = 2^k T(n/2^k) + k * n$$

$$T(n) = n * T(1) + n * \log_2 n$$

$$\boxed{T(n) = c * n + n * \log_2 n}$$

value of Rec

$$\rightarrow O(n \log_2 n)$$

2) master's method:

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

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

$$\left.\begin{array}{l} a = 2 \\ b = 2 \\ f(n) = n \end{array}\right\} \checkmark$$

Case 1: Is $n = O\left(n^{\log_2 2 - \varepsilon}\right)$, some $\varepsilon > 0$?

$$n = O\left(n^{1-\varepsilon}\right), \quad \times$$

Case 2 :- If $n = \Theta\left(n^{\log_2^2} * (\log n)^k\right)$, some k.

a) $k \geq 0$.

$n = \Theta\left(n * (\log n)^0\right)$

$n = \Theta(n)$ ✓

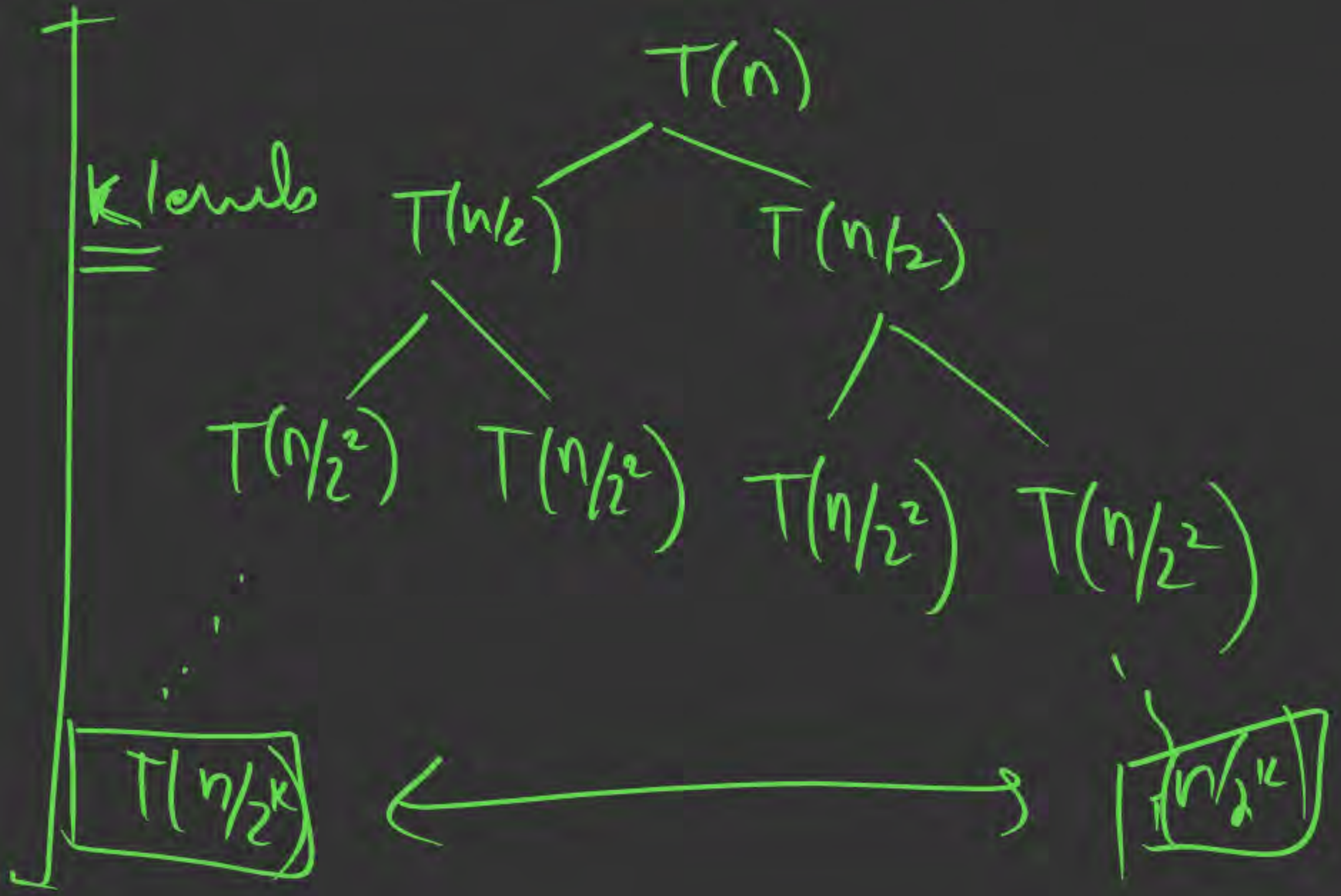
$k = 0$ ✓

Hence $T(n) = \Theta\left(n^1 * (\log n)^{k+1}\right)$

$= \Theta(n \log_2 n)$ ✓

3) Recurrence Tree Appr :

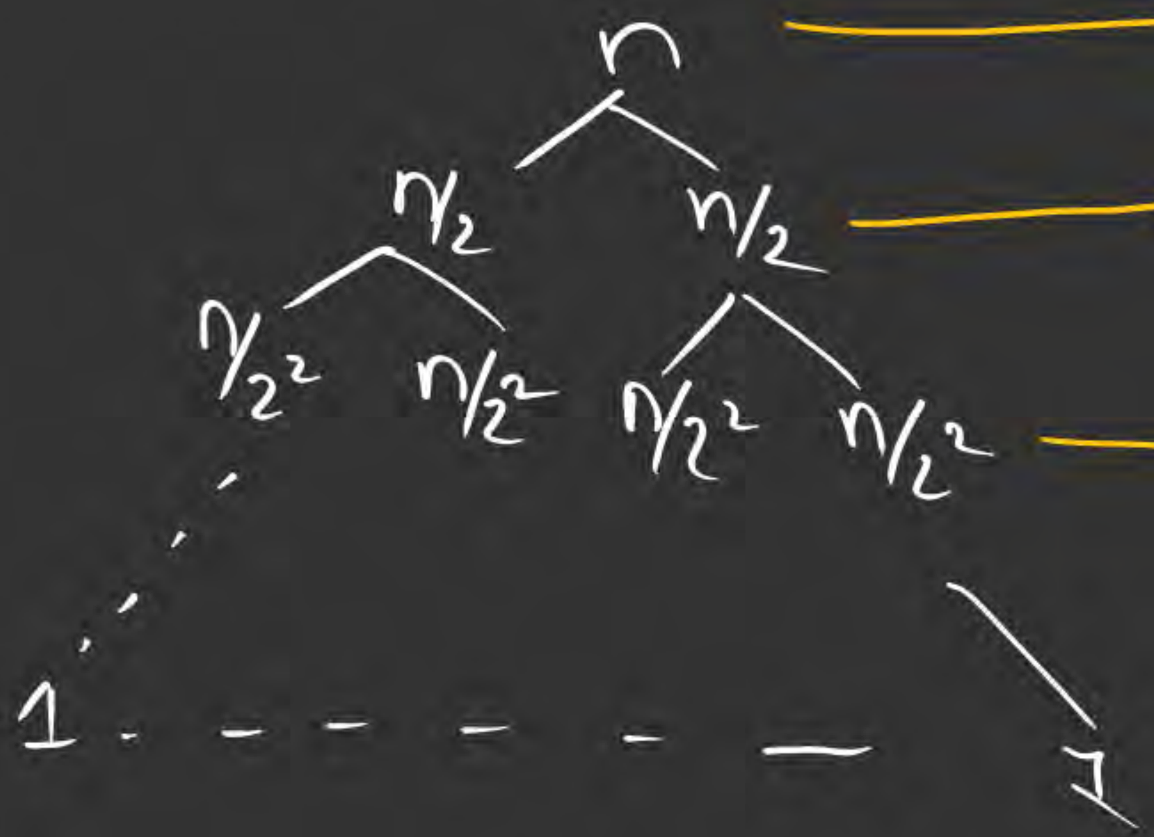
$T(n) = 2T(n/2) + n$ → cost/penalty/time



$n/2^k = 1$

$2^k = n$

$(k = \log_2 n)$ → no. of levels



$\longrightarrow n$

$\longrightarrow \left(n/2 + n/2\right) = n$

$\longrightarrow 4 \times \left(n/2^2\right) = n$

for each level,
<u>cost = n</u>.

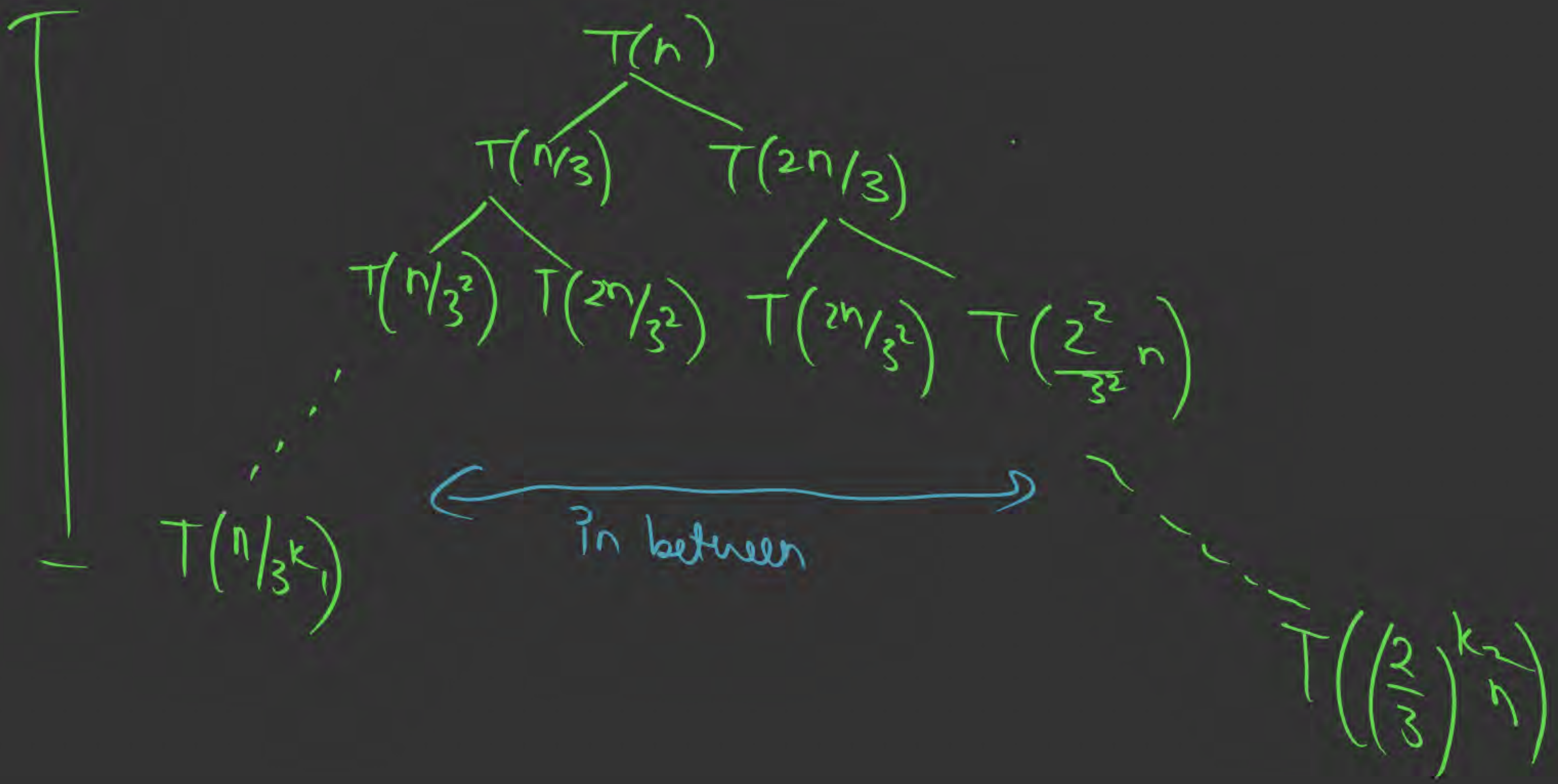
overall TC = no. of levels * TC at each level

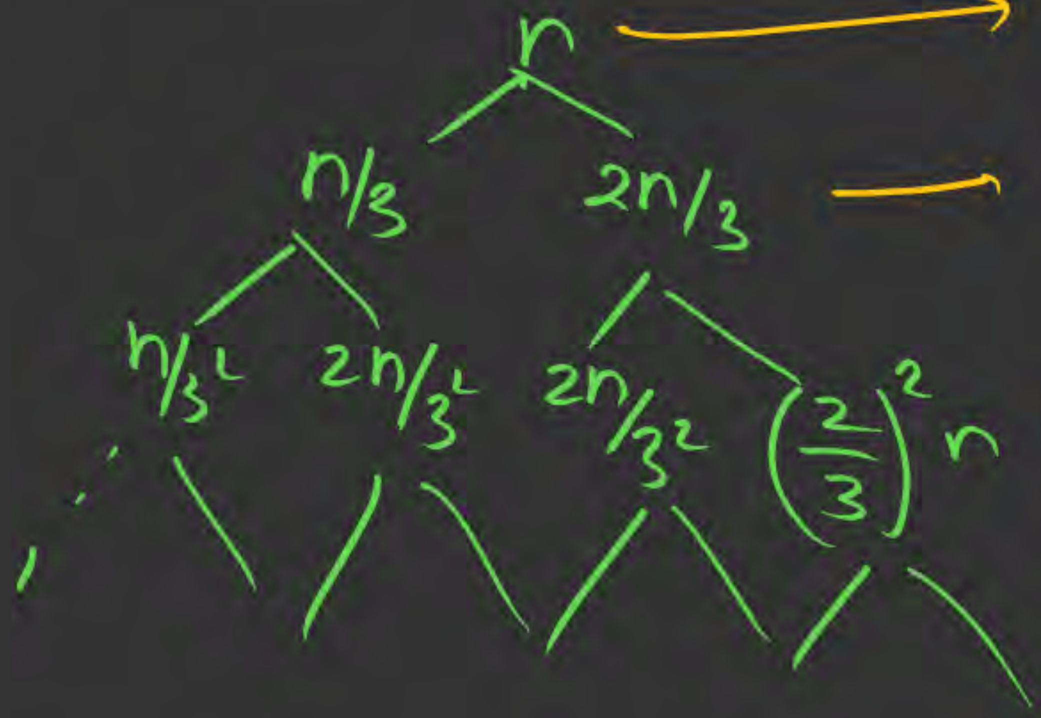
$$= K * n$$

$$= \boxed{n * \log_2 n}$$

$TC = O(n \log_2 n)$

eg 2) $T(n) = T(n/3) + T(2n/3) + n \longrightarrow$ assymetric





$$n$$

$$n/3 + 2n/3 + n$$

$$n/3^2 + 2n/3^2 + 2n/3^2 + \frac{2^2}{3^2} n = n$$

$$n\left(1/9 + 2/9 + 2/9 + 4/9\right) = n$$

$\frac{1}{3} < \frac{2}{3}$

$T(n)$

$T(n/3^k)$ $\qquad$ $T\left(\left(\frac{2}{3}\right)^k n\right)$

Shortest branch

(Lower Bound)

Longest branch

(Upper Bound)

1) <u>For lower Bound</u>: $T(n) \longrightarrow T(n/3^k)$

no. of levels $= k_1 \Rightarrow n/3^{k_1} = 1$

$$\boxed{k_1 = \log_3 n}$$

$TC \geq k_1 * n$

$TC \geq (\log_3 n) * n$

$$\boxed{TC = \Omega(n \log_3 n)} \quad \text{— (1)}$$

2) <u>For Upper Bound</u> :

$$T(n) \longrightarrow T\left(\left(\frac{2}{3}\right)^{k_2} n\right)$$

$$\left(\frac{2}{3}\right)^{k_2} * n = 1 \qquad n = \left(3/2\right)^{k_2}$$

$$\left(k_2 = \log_{3/2} n\right)$$

$$TC \leq n * k_2$$

$$TC \leq n * \log_{3/2} n$$

$$\boxed{TC = O(n \log_{3/2} n)} \text{ --- } 2$$

From 1 & 2

$$\boxed{n\log_3 n \leq T(n) \leq n * \log_{3/2} n}$$

$T(n) = \Omega(n\log_3 n)$

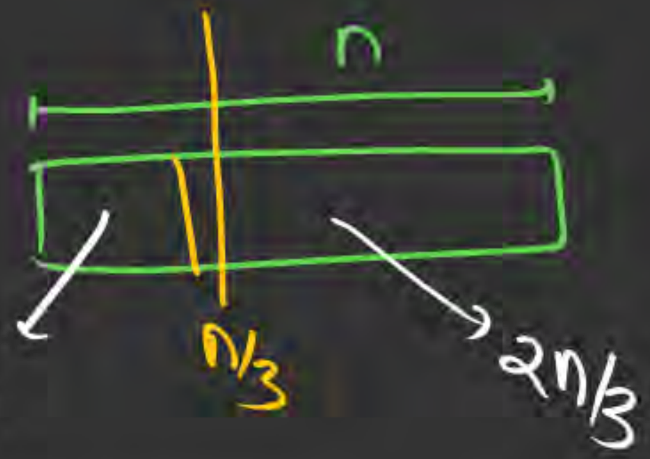
$T(n) = O(n\log_{3/2} n)$

$\Theta(n\log n)$

(Q) **QS** → always $(n/3)^{rd}$ smallest elem is selected as pivot.

WC TC = ?

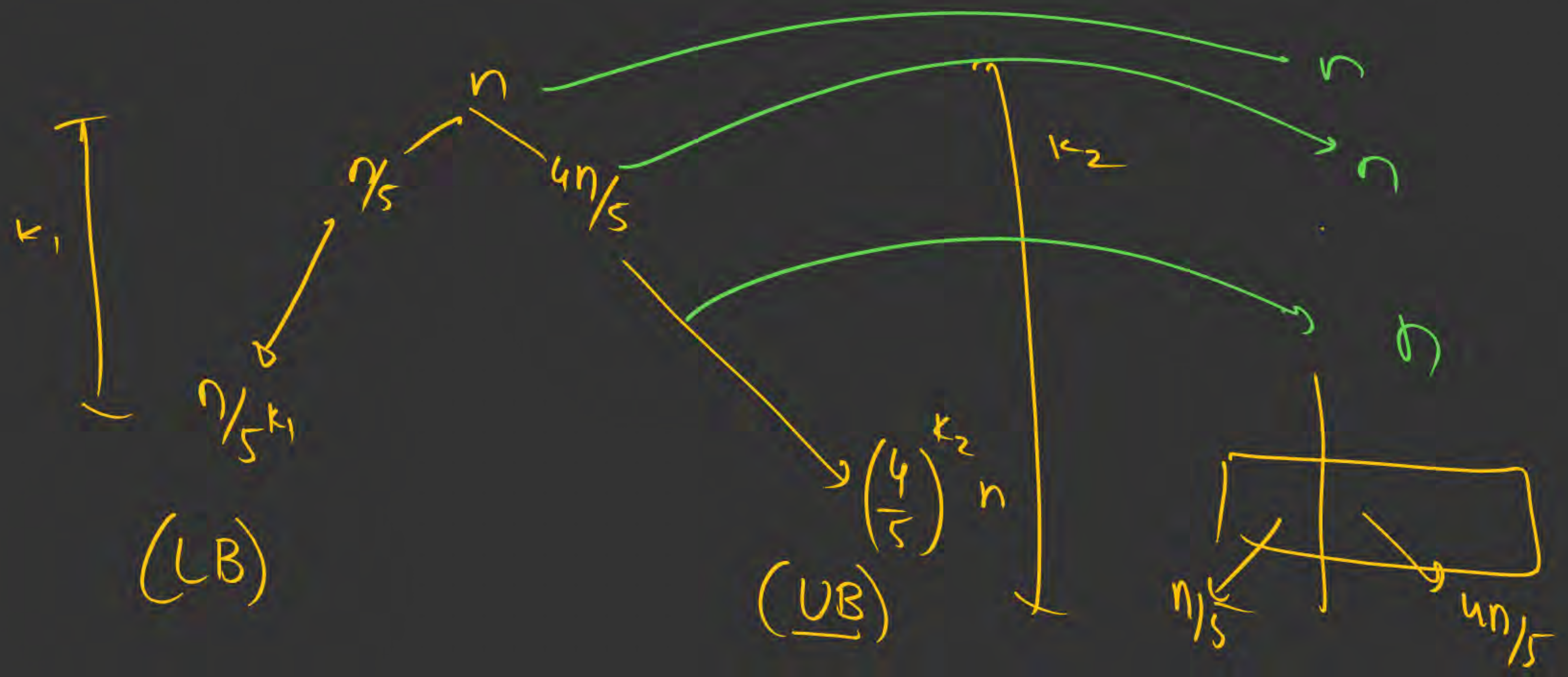
$O(n^2)$ ✗



$$\boxed{T(n) = T(n/3) + T(2n/3) + n}$$

$$= O(n\log_{3/2} n) \longrightarrow \Theta(n\log n)$$

3) $T(n) = T(n/5) + T\left(\frac{4n}{5}\right) + n$



① For lower Bound. (k_1 levels)

$$\frac{n}{5^{k_1}} = 1$$

$$\boxed{k_1 = \log_5 n}$$

$$TC \geq k_1 * n$$

$$TC = \Omega(n \log_5 n)$$

UB (k_2 levels)

$$\left(\frac{4}{5}\right)^{k_2} n = 1$$

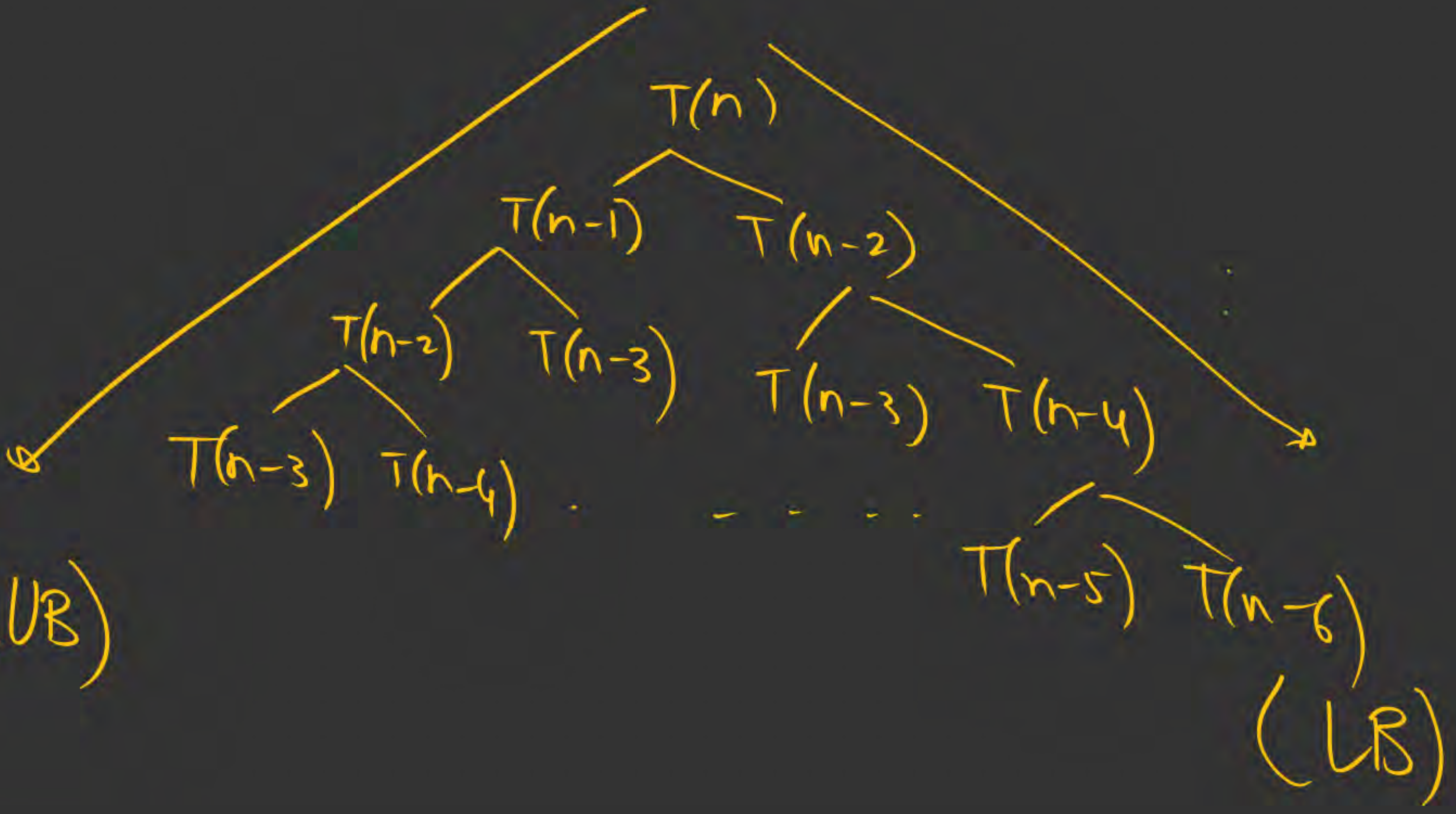
$$\boxed{k_2 = \log_{5/4} n}$$

$$TC \leq n * k_2 = O(\log_{5/4} n)$$

$$TC \leq n \times \log_{5/4} n$$

In general $= \Theta(n \log n)$

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



Left

$T(n) \rightarrow T(n-1) \rightarrow T(n-2) \cdots\cdots T(1)$

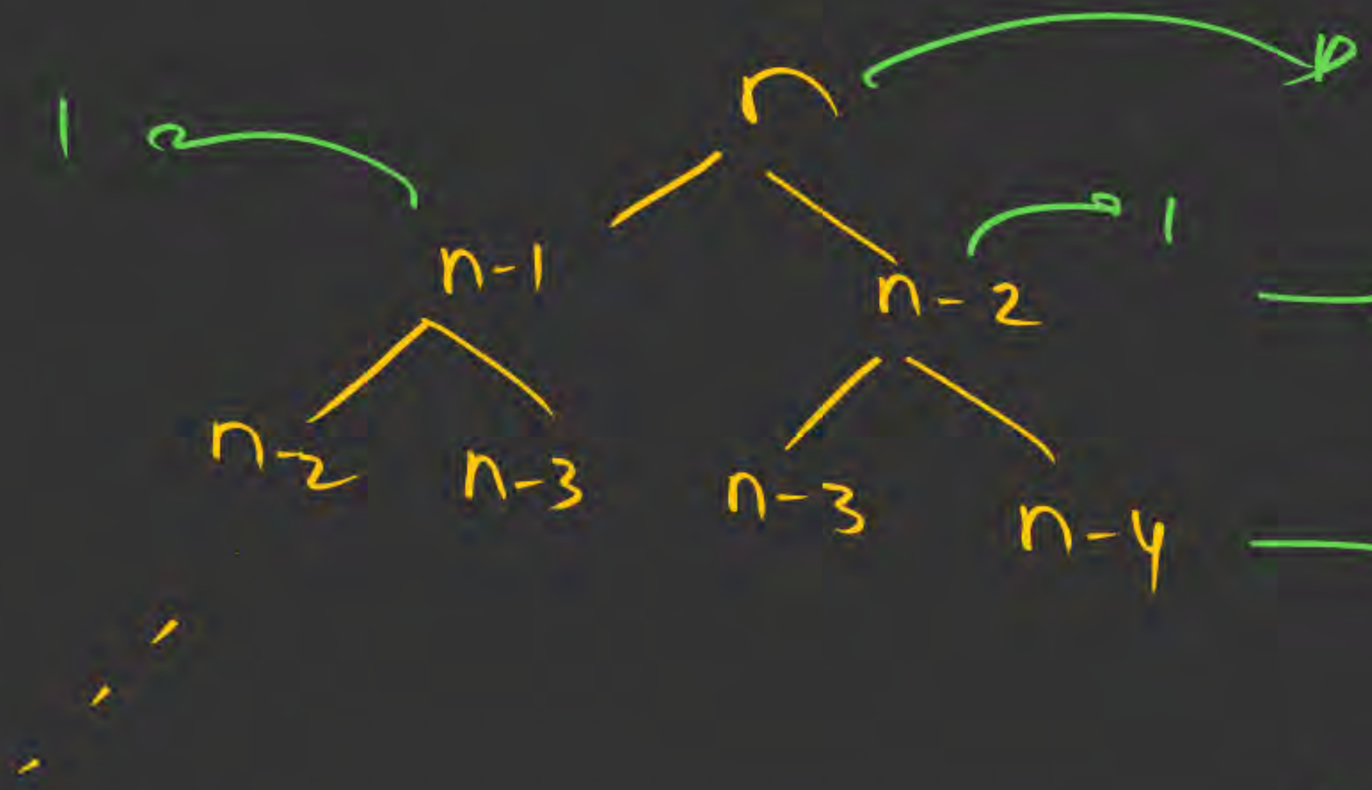
levels $= k_1 = n$ (UB)

Right

$T(n) \rightarrow T(n-2) \rightarrow T(n-4) \cdots\cdots T(0)$

levels $= k_2 = n/2$

LB



1

$1+1=2$

4

k^{th} level $\Rightarrow \underline{\underline{2^k}}$

for lower Bound.

$k_2 = n/2$

$TC \geq 2^0 + 2^1 + 2^2 \cdots 2^{k_2}$

GP. $a = 1$, $r = 2$, $n = k_2 + 1$

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

$$= \frac{1(2^{k_2+1} - 1)}{2 - 1}$$

$$= 2^{k_2+1} - 1$$

$$= (2 \cdot 2^{k_2} - 1)$$

$TC \geq 2 \times 2^{n/2} - 1$

$$\boxed{TC = \Omega(2^{n/2})}$$

UB, $k_1 = n$

$TC \leq 2^0 + 2^1 \cdots 2^{k_1}$

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

$$= \frac{1(2^{k_1+1} - 1)}{1}$$

$$= 2 \times 2^{k_1} - 1$$

$TC = O(2^{k_1})$

$TC = O(2^n)$

fibonacci(n)

$$\boxed{\begin{array}{l} T(n) = \Omega(2^{n/2}) \checkmark \\ T(n) = O(2^n) \checkmark \end{array}}$$

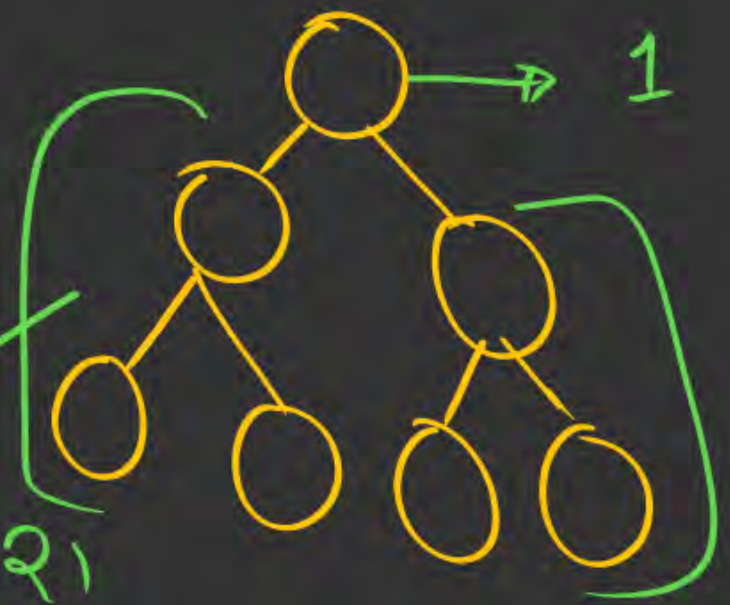
Question

#Q. The number of possible min-heaps containing each value from $\{1, 2, 3, 4, 5, 6, 7\}$ exactly once is ____.

$\{1,2,3,4,5,6,7\}$

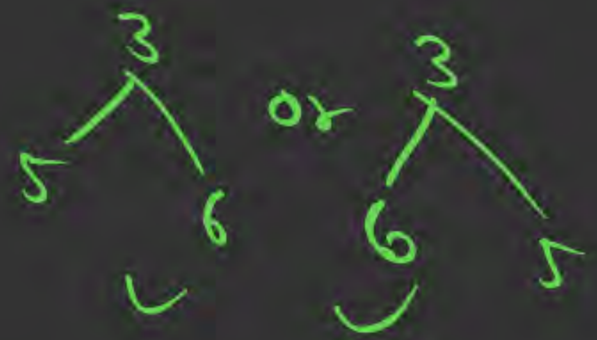
$n=7$

min-Heap



$^6C_3 * 2! * 2!$

3, 5, 6



$= {}^6C_3 \times 2 \times 2$

$= \frac{6!}{3! \times 3!} \times 4$

$= \frac{\cancel{6}\times 5\times 4\times \cancel{3!}}{\cancel{3!}\times \cancel{3!}} \times 4 = 5\times 4\times 4 = 20\times 4 = \underline{80}$

PnC

Count Min-Heap / max-Heap

$n = 15$

Appr2:-

$$T(n) = {}^{n-1}C_k \, T(k) * T(n-1-k)$$

n = Total no. of nodes

k = no. of nodes in the LST.

$n = 15$
$k = 7$

$$T(15) = {}^{14}C_7 * T(7) * T(7)$$

$n = 7$
$k = 3$

$$T(7) = {}^{6}C_3 \times T(3) \times T(3) = 80$$

$$= \boxed{{}^{6}C_3 \times 2 \times 2}$$

$n = 3$
$k = 1$

$$T(3) = {}^{2}C_1 \times T(1) \times T(1)$$

$$= 2$$

$$T(1) = 1$$

Question

#Q. The minimum number of interchanges needed to convert the array into a max-heap is

89, 19, 40, 17, 12, 10, 2, 5, 7, 11, 6, 9, 70

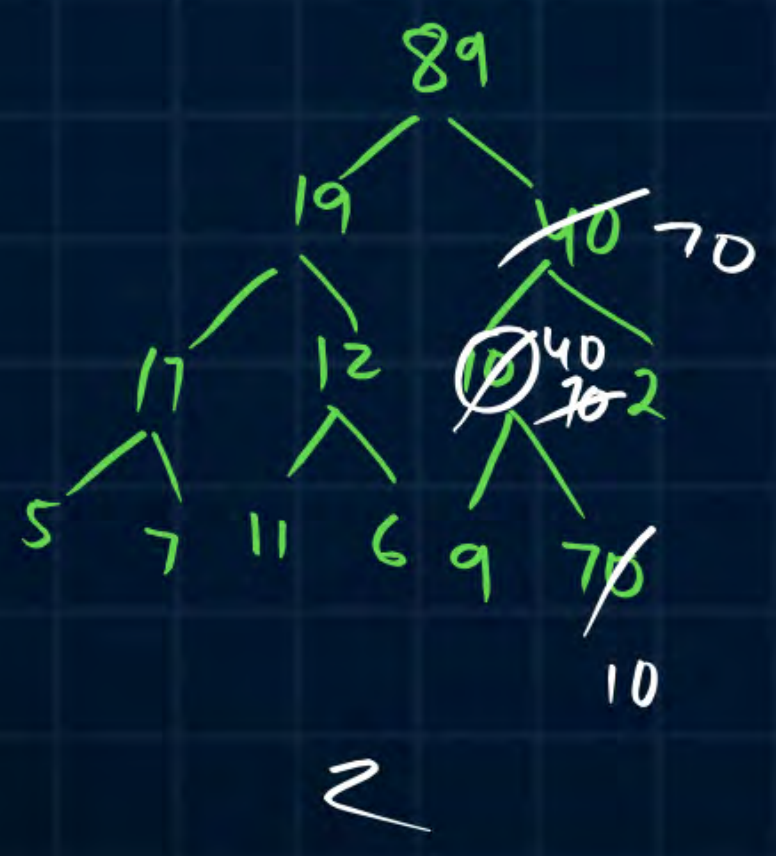
(Heapify)

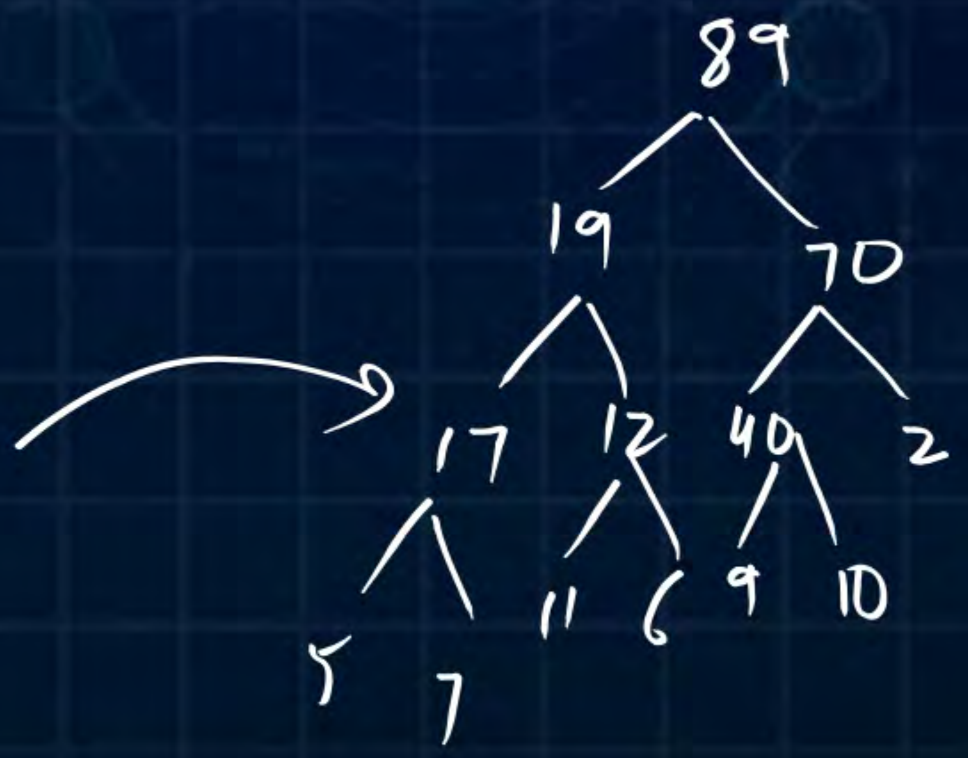
A 0

B 1

C 2

D 3





Question

#Q. Assume that the algorithm considered here sort the input sequence in ascending order. If the input is already in ascending order, which of the following are TRUE ?

✓ I. Quicksort runs in $\Theta(n^2)$ time. ✓ → WC

✗ II. Bubble sort runs in $\Theta(n^2)$ time. → ✗ BC → $O(n)$

✓ III. Merge sort runs in $\Theta(n)$ time. → ✗ WC/BC → $\Theta(n\log n)$

✓ IV. Insertion sort runs in $\Theta(n)$ time. → ✓ BC → $O(n)$

(D)

A I and II only ✗

B I and III only ✗

C II and IV only ✗

D ✓ I and IV only

Question

V. Imp

#Q. The median of n elements can be founded can be found in 0 (n) time. Which one of the following is correct about the complexity of quick sort, in which ~~remains~~ is selected as pivot?

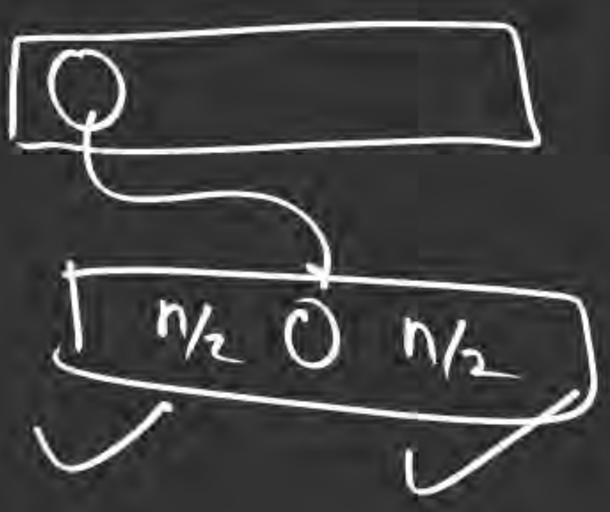
median.

A $\Theta(n)$

B $\Theta(n \log n)$

C $\Theta(n^2)$

D $\Theta(n^3)$



20 19 1 5 10

1 5 10 19 20

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

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

$$= O(n\log_2 n)$$

Question

#Q. Consider the following functions from positive integers to real number:

$$f_1(n) = 2^{100}$$

$$f_2(n) = n$$

$$f_3(n) = n\log_2 n$$

$$f_4(n) = \frac{2^{100}}{n}$$

The correct arrangement of the above functions in increasing order of asymptotic complexity is:

A. f_3, f_4, f_1, f_2

B. f_4, f_1, f_2, f_3

C. f_1, f_4, f_2, f_3

D. f_4, f_1, f_3, f_2

$f_1 = 2^{100} \rightarrow$ Const

$f_2 = n \longrightarrow$ Poly

$f_3 = n\log n \rightarrow$ Polylog

$f_4 = \dfrac{2^{100}}{n} \longrightarrow$ Decr

$$\frac{2^{100}}{n} < 2^{100} < n < n\log n$$

$$f_4 < f_1 < f_2 < f_3$$

$$\frac{1}{2} > \frac{1}{4} > \frac{1}{8} \cdots$$

Question

#Q. $f(n) = \sum_{i=1}^{n} = i^3$ then choices for f(n):

I. $\theta(n^3)$

II. $\theta(n^5)$

III. $O(n^5)$

IV. $\Omega(n^3)$

A I

B II

C III

D IV

Question

#Q. Consider the following array :

23	32	45	69	72	73	89	97

Which algorithm out of the following options uses the least number of comparisons (among the array elements) to sort the above array in ascending order ?

A. Quick sort using the last elements as pivot

B. Selection sort

C. Merge sort

D. Insertion sort

Question

#Q.Consider the following functions:

$f_1 = 2^{2n}$

$f_2 = n!$

$f_3 = 4^n$

$f_4 = 2^n$

What is the correct Decreasing order of above functions?

A) $f_1\ f_4\ f_3\ f_2$

B) $f_4\ f_2\ f_3\ f_1$

C) $f_1\ f_2\ f_3\ f_4$

D) $f_4\ f_3\ f_2\ f_1$

Thank

THANK

Keep Hustling!