Data Science & Artificial Intelligence

Algorithms

Test Series 1500+

Lecture - 07

By- Aditya sir

Recap of Previous Lecture

Topic — Sorting

Topic — Graphs

Heaps

Misc

} Questions

Topics to be Covered

- Topic
- Topic

Miscellaneous Concepts } Questions

Topic : Divide and Conquer

#Q6. What is the total number of comparisons that will be required in worst case to merge the following sorted files into a single sorted file into a single sorted file by merging together two files at a time

Files	F_1	F_2	F_3	F_4
Number of records	60	50	74	86

→ default

0%

Soln :-

$F_1 - 60 - (2)$

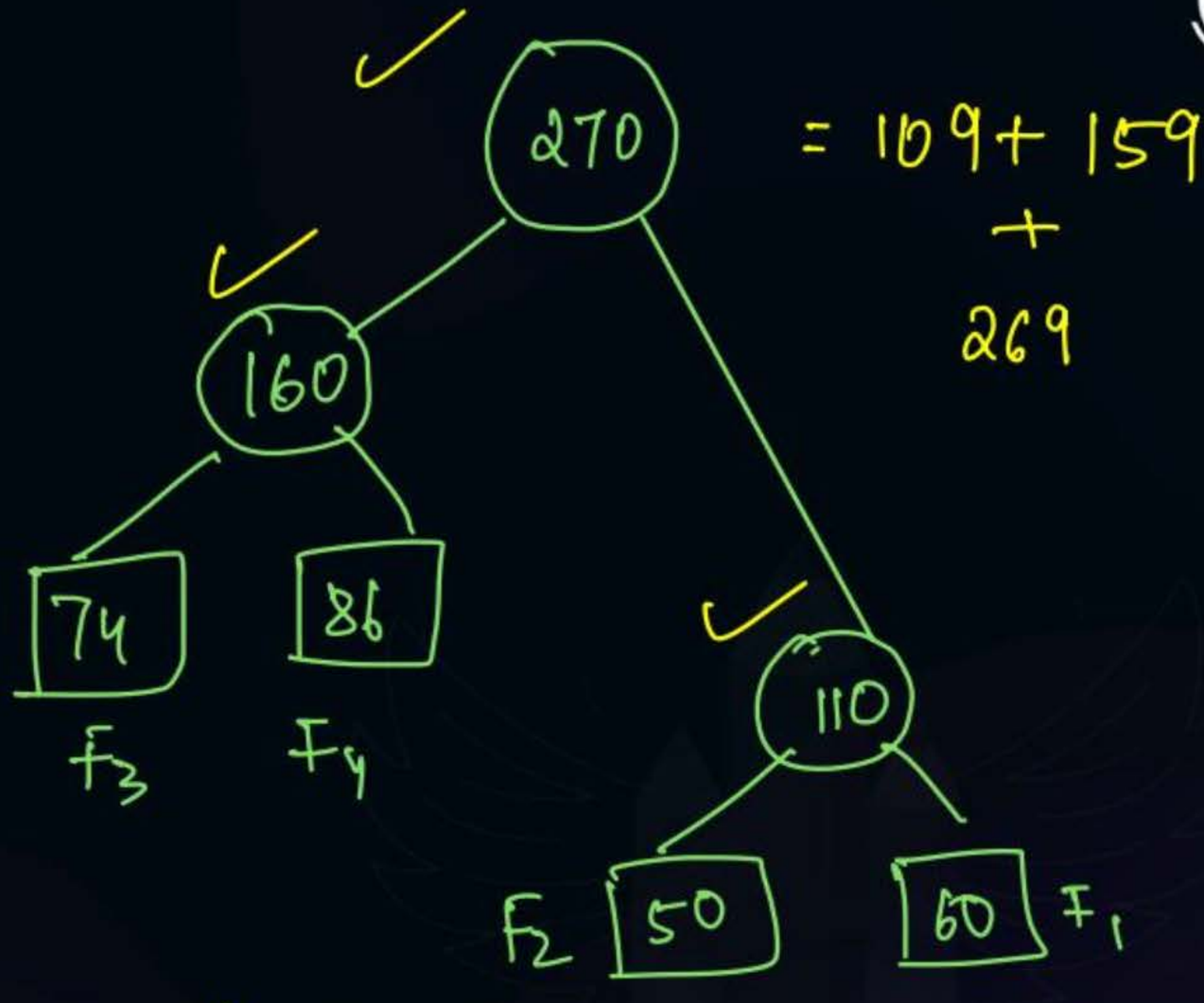
$F_2 - 50 - (1)$

$F_3 - 74 - (3)$

$F_4 - 86 - (4)$

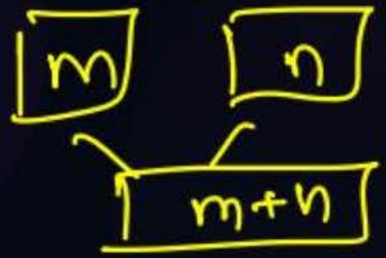
110

160



$= 109 + 159 + 269$

12
109
159
269
537



Total Comp = $(110-1) + (160-1) + (270-1)$

① No. of Record movements

↳ $m+n$

② No. of Comparisons in wc: $(m+n-1)$

② Total no. of Record movements

= Sum of internal nodes

= 110 + 160 + 270

= 540

Standard: No. of Record movements

= External path length

= $\sum d_i r_i$ = 50×2 + 60×2 + 74×2 + 86×2

= 100 + 120 + 148 + 172 = 540

[MSQ]

#Q11. Which sorting technique having the worst case time $O(n^2)$ to sort n elements:

A. Bubble sort

B. insertion sort

C. Selection sort

D. Quick sort

WC: $O(n^2)$

Sort	Best Case	Worst Case
(1) Bubble Sort:	$\Omega(n)$	$O(n^2)$
(2) Selection Sort:	$\Omega(n^2)$	$O(n^2) \longrightarrow \Theta(n^2)$
(3) Insertion Sort:	$\Omega(n)$	$O(n^2)$
(4) Quick Sort:	$\Omega(n \log n)$	$O(n^2)$
(5) Merge Sort:	$\Omega(n \log n)$	$O(n \log n) \longrightarrow \Theta(n \log n)$
(6) Heap Sort: $\longrightarrow$	$\Omega(n \log n)$	$O(n \log n) \longrightarrow \Theta(n \log n)$

[MCQ]

#Q12. Which of the following is/are correct?

A. Bubble sort is stable but not inplace sorting technique

B. Insertion sort is stable sorting technique.

C. Selection sort is a stable sorting technique.

D. Bubble sort is a inplace sorting technique

Ans: B, D

Sort	Stable	Inplace
(1) Bubble Sort	✓	✓
(2) Selection Sort	X	✓
(3) Insertion Sort	✓	✓
(4) Merge Sort	✓	X
(5) Quick Sort	X	✓
(6) Heap Sort	X	✓
(7) Radix Sort	✓	X

Topic : Divide and Conquer

#Q7. Assume that there are 8 sorted lists of n/8 elements each, if these lists are merged into a single sorted list of 'n' elements then how many key comparisons are required in the worst case using an efficient algorithm?

wc → efficient algo

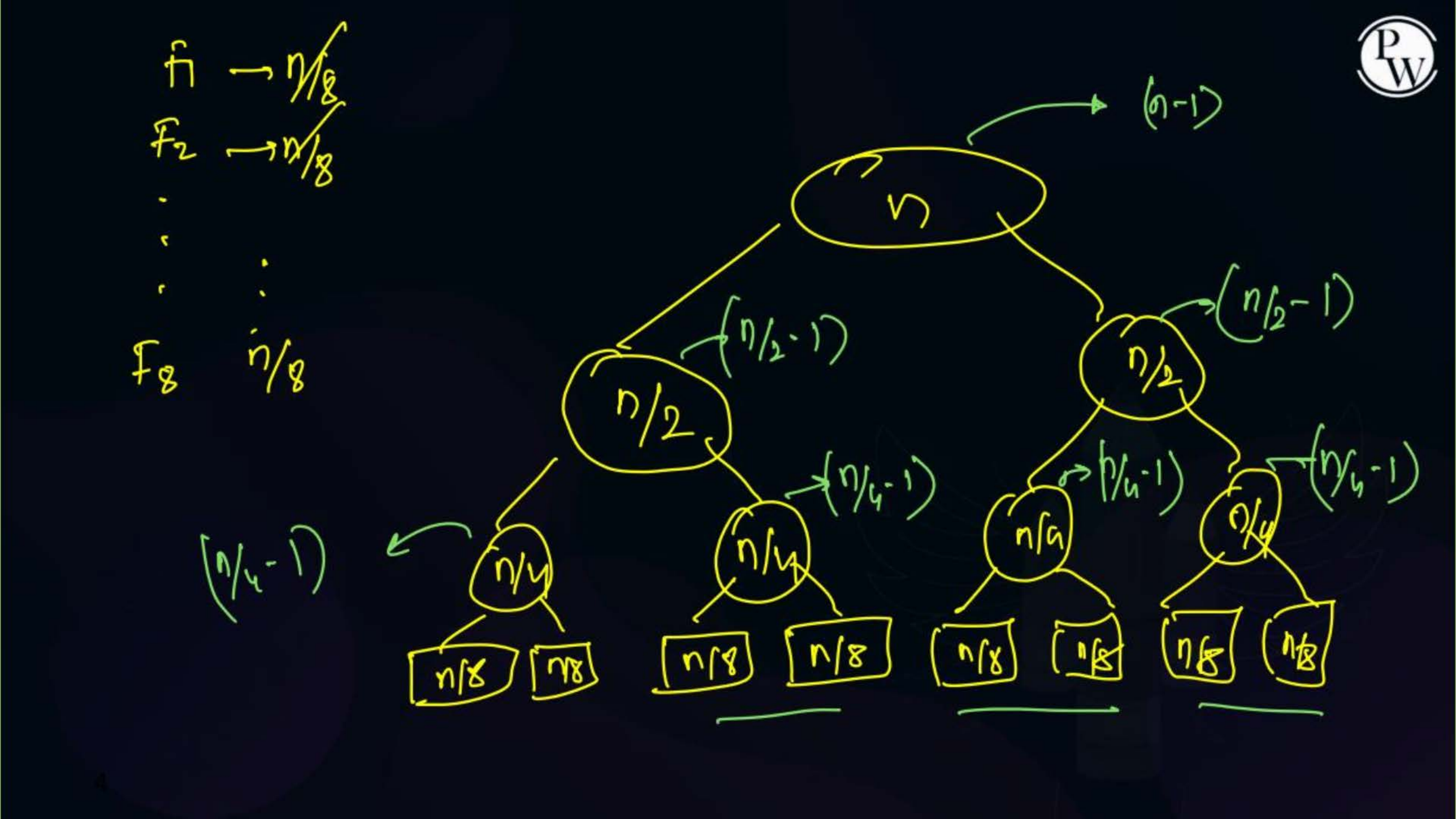
Ans: A

A $3n - 7$

B $\frac{7}{4}n - 3$

C $7n - 3$

D $\frac{6}{4}n - 3$


f1 → n/8
f2 → n/8
f8 n/8
n
(n-1)
n/2
(n/2-1)
n/2
(n/2-1)
n/4
(n/4-1)
n/4
(n/4-1)
n/4
(n/4-1)
n/4
(n/4-1)
n/8
n/8
n/8
n/8
n/8
n/8
n/8
n/8

Total = $\left(\frac{n}{4} - 1\right) \times 4 + \left(\frac{n}{2} - 1\right) \times 2 + (n-1)$

$= (n-4) + (n-2) + n - 1$

$= \underline{\underline{3n - 7}}$

Topic : Divide and Conquer

#Q15. Consider a machine which needs a minimum of 100 seconds to sort 4096 names by quick sort best case , then what is the minimum time required to sort 512 names (approximately)is_____(round off to 2 decimal)

int value

Soln:- QS : BC ⟶ 4096 nums ⟶ 100 sec

BC: 512 nums ⟶ time ?

QS in BC : $O(n \log_2 n)$ time.

For n elems = $c * n \log_2 n$ sec.

$4096 \longrightarrow C * 4096 * \log_2(4096) = 100$

$C \times 2^{12} * \log_2(2^{12}) = 100$

$$C = \frac{100}{2^{12} * 12} \quad \longrightarrow (1)$$

when $n = 512 \longrightarrow$ time ?

for $n = 512 \longrightarrow$ time $= C * n \log_2 n$

Read time $= C \times 512 * \log_2(512)$ Sec

$= \frac{100}{12 * 2^{12}} \times 2^9 * \log_2(2^9)$

$= \frac{100}{12 \times 2^{12}} \times 2^9 \times 9$

$= \frac{100}{12 \times 2^3} \times 9 = \frac{100 \times 3}{4 \times 8}$

$$= \left(\frac{3}{32}\right) \times 100$$

$$= \boxed{9.38}$$

Shortcut

$n = 4096 \longrightarrow 100$

$c \times 2^{12} \times \log_2(2^{12}) = 100$ — (1)

$n = 512 \longrightarrow ?$

$c * 2^9 * \log_2(2^9) = ?$

$\frac{100}{2^{12} \times 12} \times 2^9 \times \log_2(2^9) =$ Read ans

Standard Process :

n elems ⟶ TC : $O(n \log n)$ units

time = $O(n \log n)$

time $\approx$ $n \log n$

time = $c * n \log n$

[MCQ]

#Q.6 Consider the following statements : -

I. The second smallest element in a min heap is always a root of left sub tree.

II. The second largest element in a max heap is always a child of the root node.

III. A max-heap can be constructed from a binary search tree in O(n) time.

IV. A binary search tree can be constructed from a max-heap in O(n) time.

Which of the above statements are TRUE?

A I, III and IV

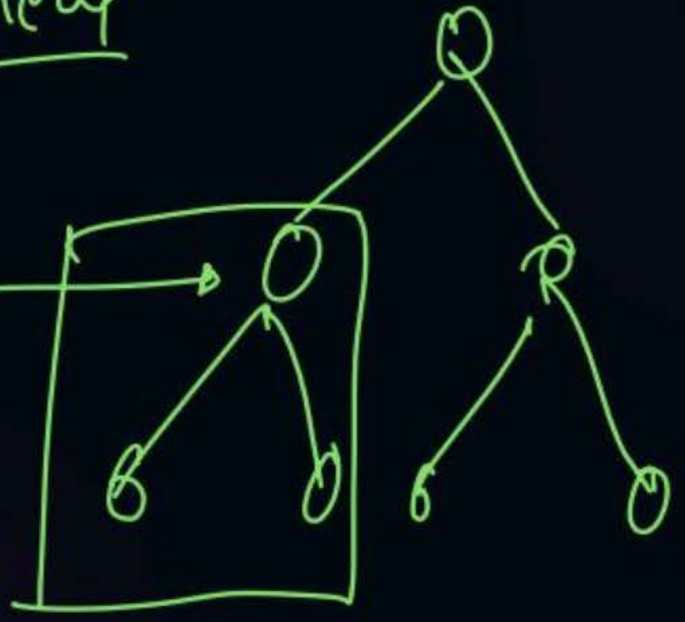
B II, III

C I, II and III

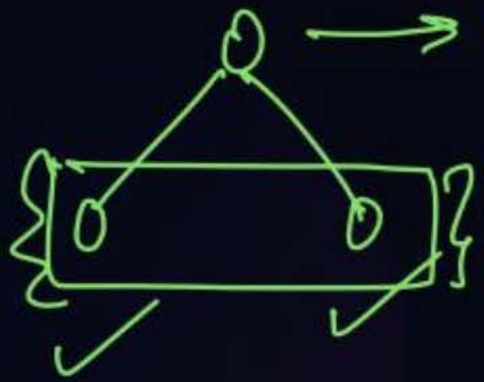
D I, II and IV

Soln :-

min-Heap

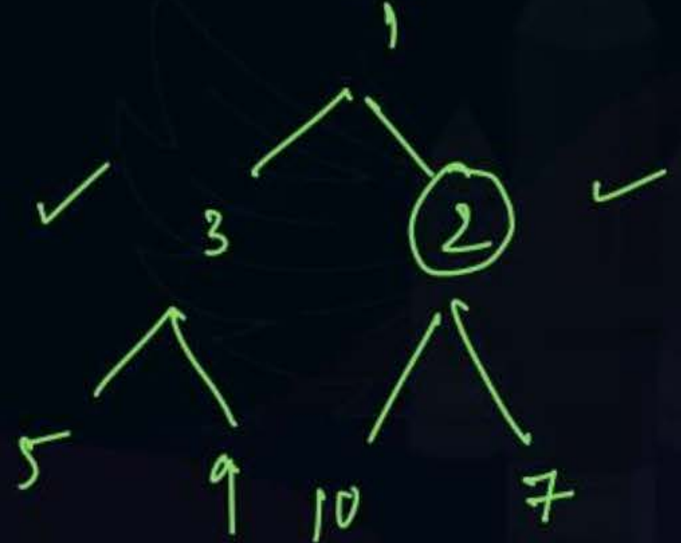


2nd smallest elem



→ max child

→ 2nd max child



BST ⟶ Max-Heap : $O(n)$

Topic : Divide and Conquer

#Q14. Assume that, quick sort implementation is used to sort an array in ascending order. After the first partition step has been completed, the contents of the array are in the following order. 150, 330, 440, 220, 880, 790, 1200, 1000, 180, 110, 160

Which of the following elements could be a pivot element?

A 440

B 330

C 880

D None of these

Ans: D

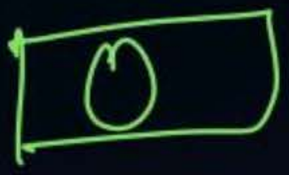
Soln:-

1	2	3	4	5	6	7	8	9	10	11
150	330	440	220	880	790	1200	1000	180	110	160

×	×	×	×	×	×	×	×	×	×	×
110	150	160	180	220	330	440	790	880	1000	1200

Sorted:

Concept :- Partition Algo :



1) Pivot gets placed at its correct position
2) All the elems on left of pivot, will be lesser than the pivot.
3) all the elems on the right of pivot, will be greater than pivot.

Q2 :- After 1st partition iteration, the seq is

200 300 110 500 650 720

which of the following can be a pivot element.

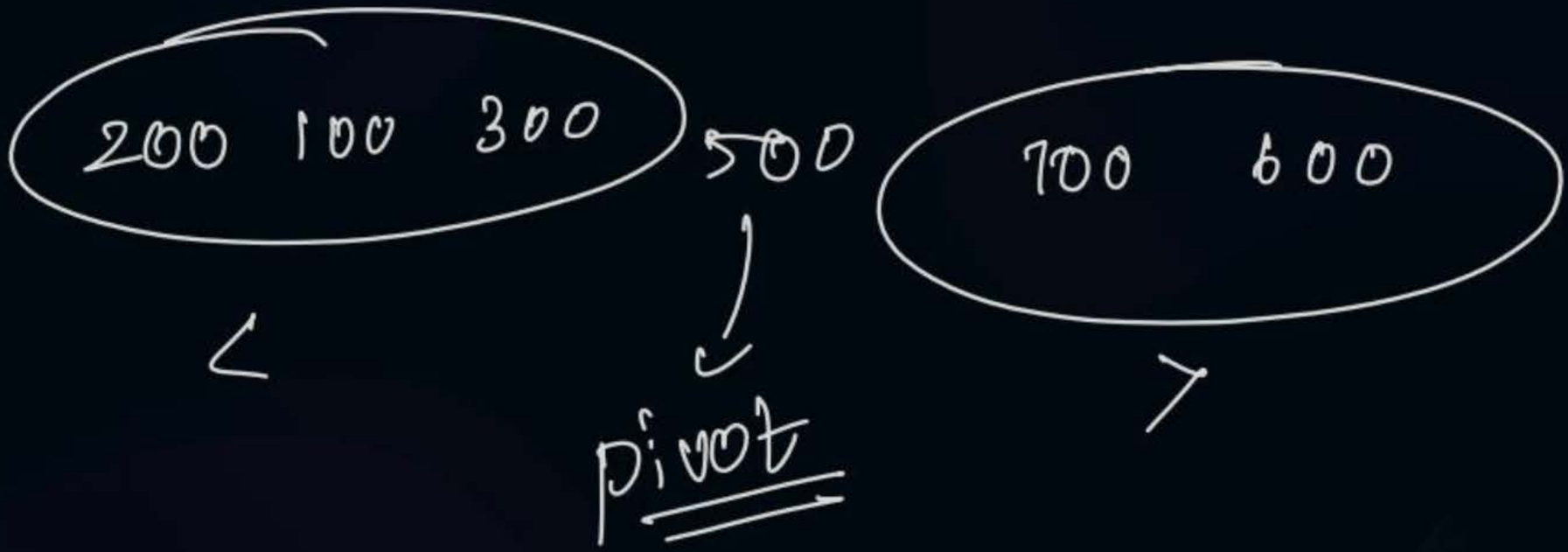
A) 110 B) 500 C) 650 D) 720

Given: 200 300 110 500 650 720

× × × ✓ ✓ ✓

Sorted: 110 200 300 500 650 720

Ans: 500, 650, 720


200 100 300
500
700 600
pivot

Topic : Divide and Conquer

#Q16. Consider a modified version of Quick sort where we have an input as an sorted array X[1 ..... n], all element of the array is distinct and $n \geq 3$. Pivot is median of set of 3 elements (first, middle, last). Then what is the worst-case time complexity of this algorithm?

Ans: B

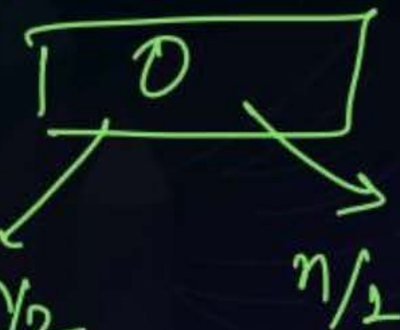
A $O(n^2)$ ✗

B $O(n \log n)$ ✓

C $O(n^2 \log n)$ ✗

D $O(n \log n \log n)$ ✗

→ $O(n \log_2 n)$



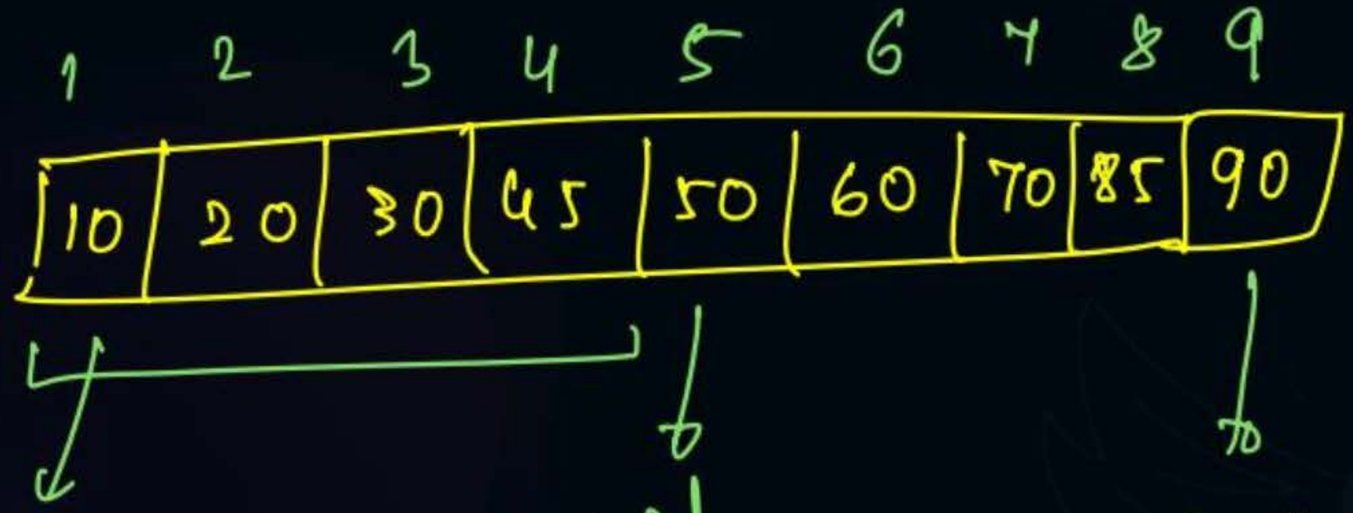
$T(n) = T(n/2) + T(n/2) + O(n)$ → partition

$T(n) = 2T(n/2) + n$ ✓

Soln :-

i/p: A: Sorted

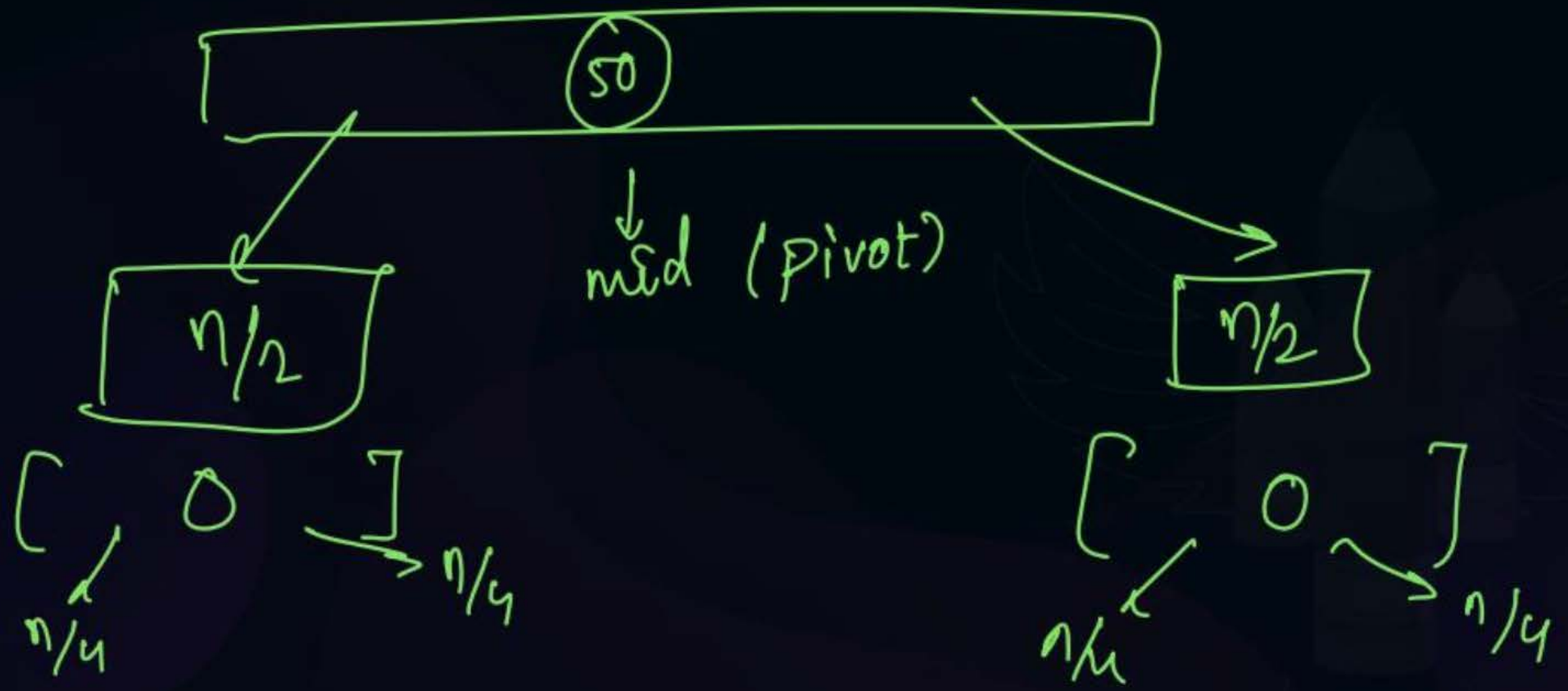
pivot : median [First, mid, last]



40
20
80

median → mid elem in the Sorted seq. (20 40 80]

median : [10, [50], 90] → 50

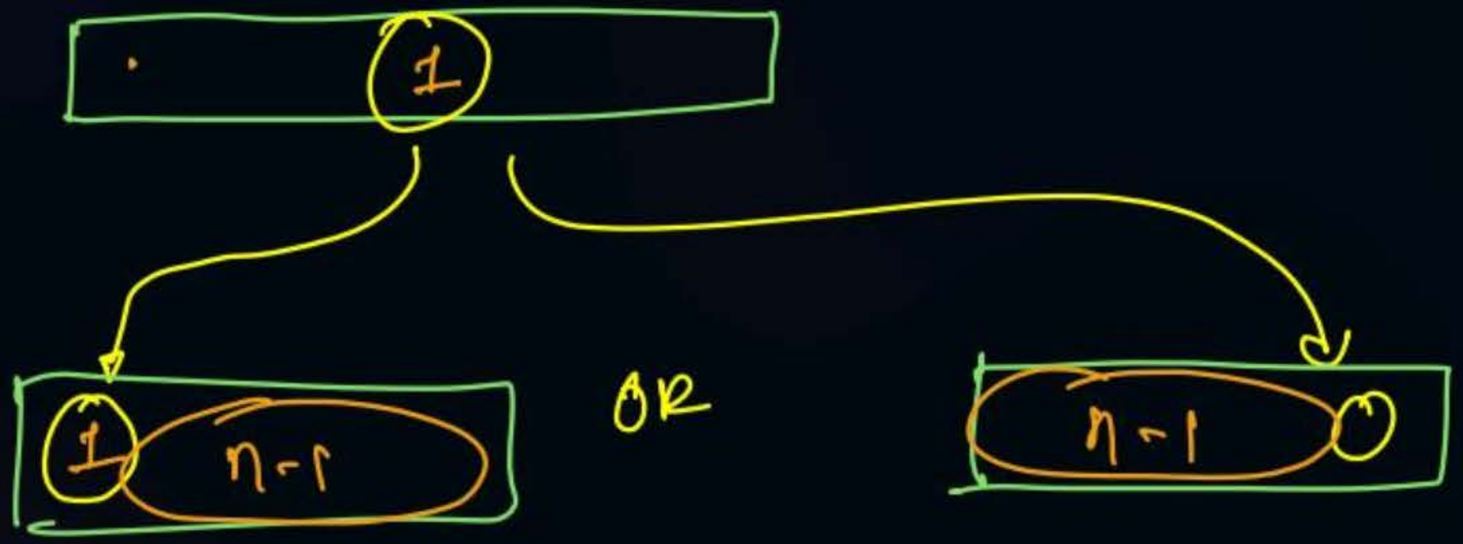


(Q) In an array, always middle elem is taken as pivot.

Then what is the WC complexity of this version of QS ?

Imp

Before partition



Imp: Selecting the mid is not same as BC. what matters is, that after partition, where does pivot gets placed (mid for BC)

[MCQ]

#Q13. What is the number of comparisons (element comparison) needed to sort n elements using radix sort?

A. 0

B. O(n)

C. O(logn)

D. O(log logn)

Ans: A

Radix Sort → non-Comparison based Sorting.

12 80

0 Comparisons

Topic : Analysis of algorithm

#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 ✗

Ans: B

Soln :-

$f_1 = 2^{100}$ ⟶ Const

$f_2 = n$ ⟶ Poly

$f_3 = n \log_2 n$ ⟶ Polylog

$f_4 = \frac{2^{100}}{n}$ ⟶ Decr

$n < n \log n$

$=$

In general

Decr < Const < Poly

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

$[f_4\ f_1\ f_2\ f_3]$

2 mins Summary

- Topic
- Topic
- Topic
- Topic

Misc Questions

THANK - YOU

Telegram Link for Aditya Jain sir:
https://t.me/AdityaSir_PW