GATE

CRASH COURSE

DS & AI

Algorithms

Divide & Conquer (Part 02)
(Lecture 5)

By - Aditya sir

Topics to be covered

1

2 Divide and Conquer (DnC)

About Aditya Jain sir

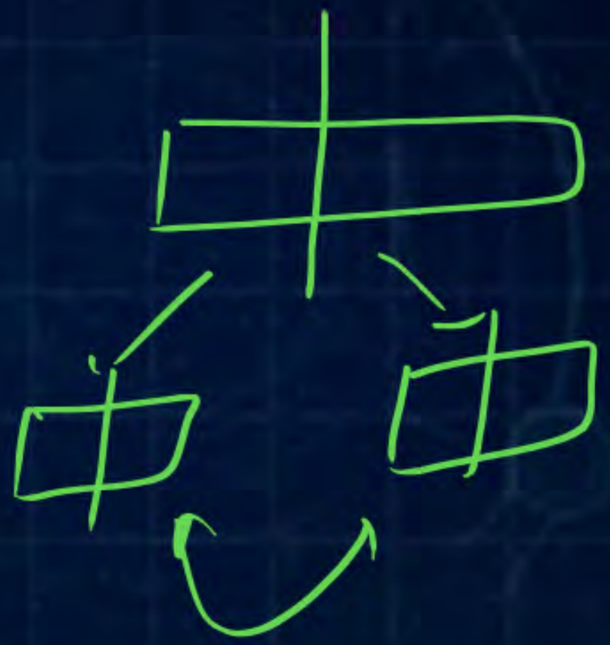
1. Appeared for GATE during BTech and secured AIR 60 in GATE in very first attempt - City topper
2. Represented college as the first Google DSC Ambassador.
3. The only student from the batch to secure an internship at Amazon. (9+ CGPA)
4. Had offer from IIT Bombay and IISc Bangalore to join the Masters 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.

Topic : (Lecture Schedule)

2. Divide & Conquer

1. General Method
2. Max-Min Problem
3. Binary Search
4. Merge Sort
5. Quick Sort
6. Master Method for D and C Recurrences

Part-1 DnC



Part-2 DnC

7. Practice Questions

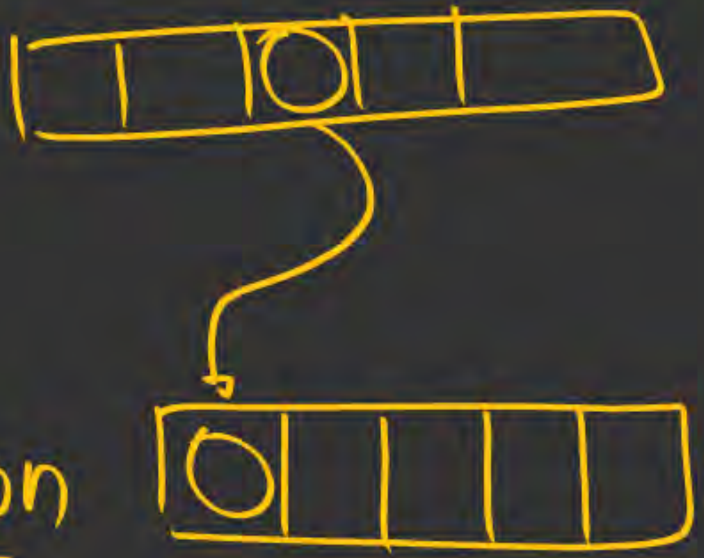
Quick Sort Questions:

1) **Variant of Quick Sort**

↳ Always the middle element is selected as the Pivot. What is the Worst Case Complexity of such an algo?

Soln :

Before



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

$$\hookrightarrow \underline{O(n^2)}$$

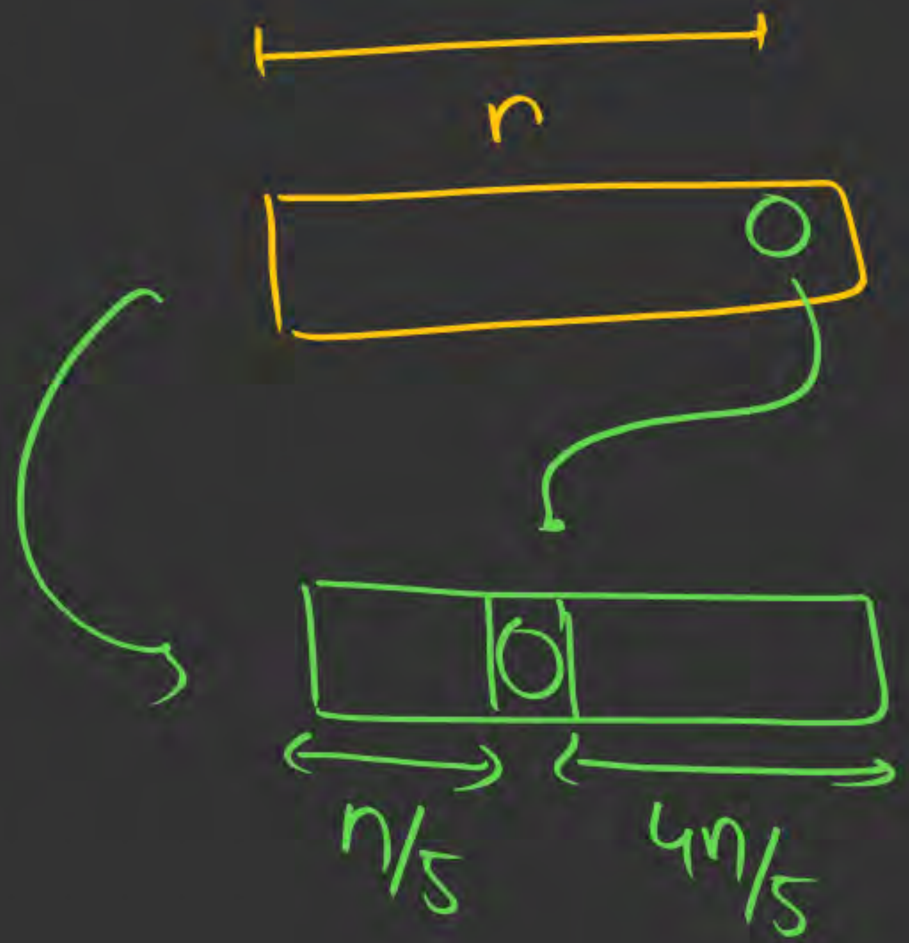
<u>WC</u>

2) Variant of <u>Quick Sort</u>

↳ Always $(n/5)^{th}$ smallest elem is selected as pivot. Then the WC Complexity of such an Algo?

Soln:

after partitn.



Partition algo

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

$$\approx O(n\log(n))$$

3) Variant of Quick Sort

↳ Always the median is selected as pivot, and median can be found in $O(n)$ time

Then the Worst Case Complexity of such an Algo?

median :

10	2	7	50	9

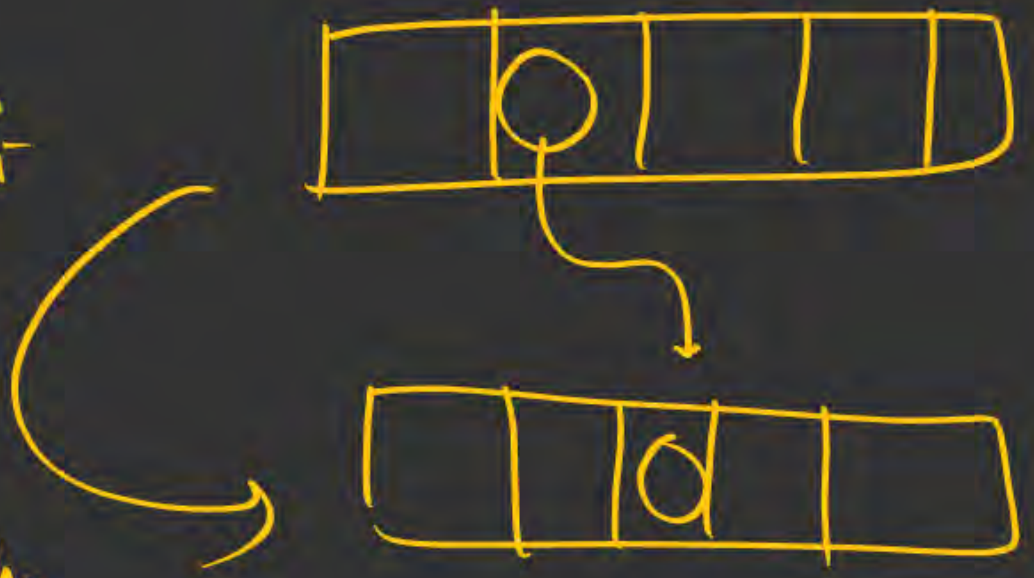
2	7	9	10	50

median

(**Mid** elem in **Sorted Seq**)

Soln :-

after partition



partition

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

median

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

$$O(n \log_2 n)$$

Question

#Q. Assume that merge sort takes 40 sec to sort 64 elements in worst case. What is the approximate number of elements that can be sorted in the Worst case using merge sort using 8 minutes?

Soln :- Given $n = 64 \longrightarrow$ 40 secs (merge Sort)

$? \longrightarrow$ 8 mins ?

WC complexity to sort n elems in Merge Sort = O(nlogn)

$n \longrightarrow O(n\log_2 n)$ units

$n \Rightarrow c * n\log_2 n$ sec

(given)

$64 \Rightarrow C \times 64 \log_2(64) = 40$ sec

$C \times 64 \times \log_2(2^6) = 40$

$C \times 64 \times 6 = 40$

$$\left(C = \frac{40}{64 \times 6} \right)$$

Reqd

$8 \text{ mins} = 8 \times 60 \text{ sec}$

$n \text{ elms} \longrightarrow O(n \log_2 n) \text{ sec}$

$\Rightarrow c \times n \log_2 n = 8 \times 60$

$\frac{40}{64 \times 6} \times n \log_2 n = 8 \times 60$

$n \log_2 n = \frac{\overset{4}{\cancel{8}} \times \overset{3}{\cancel{60}} \times 6 \times 64}{\cancel{2}\cancel{40}}$

$$\boxed{n \log n = 4 \times 3 \times 6 \times 64}$$

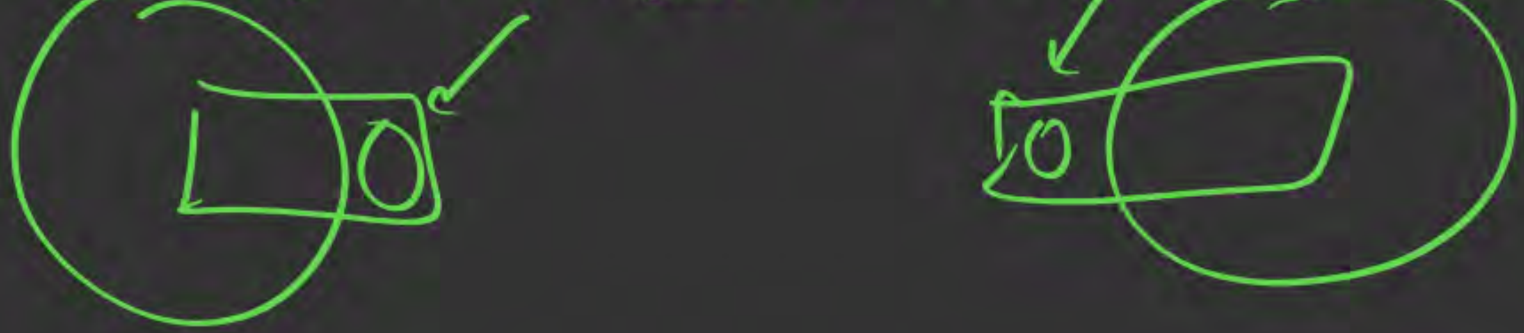
Question

#Q. An array of 30 distinct elements is to be sorted using quicksort. Assume that the pivot element is chosen uniformly at random.
The probability that the pivot element gets placed in the worst possible location in the first round of partitioning (rounded off to 2 decimal places) is ___.

Soln: $n = \underline{\underline{30}}$

$P(\text{elem}) = \frac{1}{30}$

WC $\longrightarrow$ Max elem or Min elem



$$\text{Reqd} = \frac{1}{30} + \frac{1}{30} = \frac{2}{30} = \frac{1}{15} \checkmark$$

Master's Method/ Theorem

Topic : (Design Strategies)

Master Theorem

$\boxed{T(n) = a \,.\, T(n/b) + f(n),\ a \geq 1;\ b > 1;\ f(n) \text{ is +ve}}$ ✓

$T(n)$

Case I: If $f(n)$ is $O\left(n^{\log_b a - \epsilon}\right)$ for some $\epsilon > 0$, then $\theta\left(n^{\log_b a}\right)$

Case II: If $f(n)$ is $\theta\left(\cancel{n^{\log_b a} * \log_n^k}\right)$ for some k, such that

$\theta\left(n^{\log_b a} * (\log n)^k\right)$

a) $k \geq 0$, then $T(n)$ is $\theta\left(n^{\log_b a} * \log_n^{k+1}\right)$

b) $k = -1$, then $\theta\left(n^{\log_b a} * \log\log n\right)$

Case III: If $f(n)$ is $\Omega\left(n^{\log_b a + \epsilon}\right)$ for some $\epsilon > 0$, and

$a.f(n/b) \leq \delta.f(n)$ for some $\delta < 1$, then

$T(n)$ is $\theta(f(n))$

1) $T(n) = 9T(n/3) + n$

$$T(n) = a \times T(n/b) + f(n)$$

$$\left.\begin{aligned} a &= 9 \\ b &= 3 \\ f(n) &= n \end{aligned}\right\} \checkmark$$

Case 1: If $F(n) = O\left(n^{(\log_b a - \varepsilon)}\right)$ for some $\varepsilon > 0$,

$$n = O\left(n^{(\log_3 9 - \varepsilon)}\right), \text{ for } \underline{\underline{\text{some}}}\ \varepsilon > 0$$

$$n = O\left(n^{2-\varepsilon}\right),\ \varepsilon > 0$$

$, \varepsilon = 0.1, 0.5$

Case 1 $\rightarrow$ satisfied

Hence $T(n) = \Theta\left(n^{\log_b a}\right) = \Theta\left(n^{\log_3 9}\right) = \underline{\Theta(n^2)}$

2) $T(n) = 4T(n/4) + n\log n$

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

$$\left.\begin{aligned} a &= 4 \\ b &= 4 \\ f(n) &= n\log n \end{aligned}\right\} \checkmark$$

Case 1:

Is $n\log n = O\left(n^{(\log_4^4 - \varepsilon)}\right)$, some $\varepsilon > 0$?

$n\log n = O\left(n^{(1-\varepsilon)}\right)$, $\varepsilon > 0$

$\longrightarrow$ fails

Case 2 : Is $n\log n = \Theta(n^{\log_4 4} * (\log n)^k)$, some k

a) $k \geq 0$? $n\log n = \Theta(n * (\log n)^k)$, some $k \geq 0$?

$k=1$, $n\log n = \Theta(n\log n)$ ✓

Case 2 ✓, $k \geq 0$ ✓ $(k=1)$

Hence $T(n) = \Theta(n^{\log_b a} * (\log n)^{k+1}) \Rightarrow \Theta(n^{\log_4 4} * (\log n)^{1+1})$

$= \Theta(n^1 * (\log n)^2)$

3) $T(n) = T(n/4) + n$

Soln :- $T(n) = a * T(n/b) + f(n)$

$$\left.\begin{aligned} a &= 1 \\ b &= 4 \\ f(n) &= n \end{aligned}\right\} \checkmark$$

Case 1: Is $n = O\left(n^{(\log_4 1 - \varepsilon)}\right)$, some $\varepsilon > 0$?

$n = O\left(n^{0-\varepsilon}\right)$, some $\varepsilon > 0$?

$\hookrightarrow$ fails

Case 2.

Is $n = \Theta(n^{\log_1 1} * (\log n)^k)$, some k

a) $k \geq 0$, $n = \Theta((\log n)^k)$? $\rightarrow$ fails

b) $k = -1$ $\longrightarrow$ fails

Case 3: Is $n = \Omega(n^{(\log_4 1 + \varepsilon)})$, some $\varepsilon > 0$?

$n = \Omega(n^{\varepsilon})$, $\varepsilon > 0$? Yes $\varepsilon = 0.1$

and

$a * F(n/b) \le \delta * F(n)$

$F(n) = n$

$1 * \frac{\cancel{n}}{4} \le \delta * \cancel{n}$, some $\delta < 1$ $\left(\frac{1}{4} \le \delta\right) \longrightarrow$ True

$f(n/4) = n/4$

Case 3 $\longrightarrow$ satisfied.

$T(n) = \Theta(f(n)) \Rightarrow \boxed{T(n) = \Theta(n)}$

4) $T(n) = T(\sqrt{n}) + 10$

→ (change of variable mtd)

$$T(n) = T(\sqrt{n}) + 10 \quad \text{—— ①}$$

Let $n = 2^k \Rightarrow \sqrt{n} = 2^{k/2}$

$$T(2^k) = T\left(2^{k/2}\right) + 10$$

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

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

$$S(k) = S\left(\frac{k}{2}\right) + 10 \quad \text{—— ②}$$

$$S(k) = S\left(\frac{k}{2}\right) + 10 \quad \text{—— (2)}$$

$$\left.\begin{aligned} a &= 1 \\ b &= 2 \\ f(n) &= 10 \end{aligned}\right\} \checkmark$$

Check case 1: Is $10 = O\left(k^{\log_2 1 - \varepsilon}\right)$, some $\varepsilon > 0$?

$10 = O\left(k^{-\varepsilon}\right)$? — fails

Case 2: $10 = \Theta\left(k^{\log_2 1} * (\log k)^{k'}\right)$

a) $k' \geq 0 \rightarrow k' = 0$ $\quad 10 = \Theta\left((\log k)^0\right) = \Theta(1)$ ✓

Yes

$S(k) = \Theta\left(k^0 * (\log k)^{k'+1}\right) = \Theta(\log k)$

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

$n = 2^k$

$\underline{k = \log_2 n}$

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

$$T(n) = \underline{\Theta(\log(\log_2 n))}$$

Question

#Q. Consider the following array

A

[illegible]	500	700	130	90	850	210	160

How many comparisons are needed to ~~start~~ Sort the above array using insertion sort?

Soln: 500 700 130 90 850 210 160

pass1: 500 | 700 130 90 850 210 160

pass2: 500 700 | 130

pass3: 130 500 700 | 90

pass4: 90 130 500 700 | 850

pass5: 90 130 500 700 850 | 210

pass6: 90 130 210 500 700 850 | 160

→ [90 130 160 210 500 700 850]

Comp
1 ✓
2 ✓
3 ✓
1
4 ✓
5

16 ✓

Question

#Q. Consider a list which contains np sorted array each of size n/p and is merged using merge sort, then what is the tightest upper bound worst case complexity?

(2-way merging)

A. $O(np^2 \log np)$ ✗

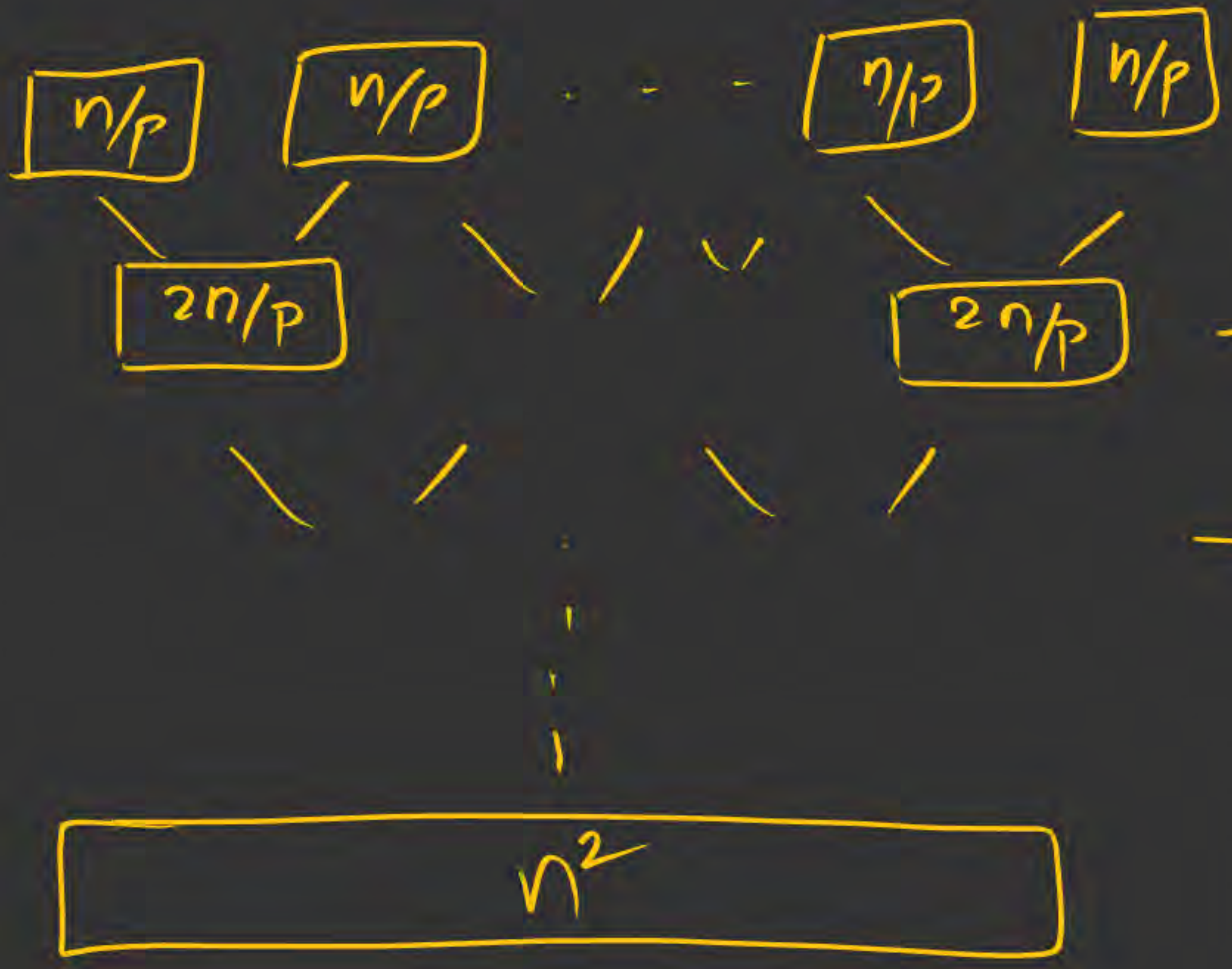
B. $O(n^2 \log n)$ ✗

C. $O(n^2 \log np)$ ✓

D. None of these ✗

C.

Soln :-



$\longrightarrow$ np arrays

$\longrightarrow \left(\frac{np}{2}\right)$ arrays

$\longrightarrow \left(\frac{np}{4}\right)$ arrays

$\longrightarrow$ 1 array

$\rightarrow$ np arrays $\longrightarrow$ n/p each

Total elms = $n\cancel{p} \times \frac{n}{\cancel{p}} = \boxed{n^2}$ ✓

TC at every level $\longrightarrow$ $O(n^2)$

Total no. of levels ?

$$\left[np \rightarrow \frac{np}{2} \rightarrow \frac{np}{4} \cdots\cdots 1 \right]$$

$\log_2(np)$

Total / overall TC = no. of levels × TC of each level

$$= O\left(\log_2(np) * n^2\right)$$

$$= \underline{O\left(n^2 * \log(np)\right)}$$

Question

#Q. Consider the following array with 8 elements:

60	50	45	85	55	90	65	12

What is result after 3rd pass of bubble sort?

(A) 50, 60, 55, 45, 12, 65, 85, 90 ×

(B) 12, 45, 50, 60, 90, 65, 55, 85 ×

(C) 45, 50, 55, 60, 12, 65, 85,90 ✓

(D) 50, 55, 45, 60, 12, 65, 85, 90 ×

C

Soln :- A = 60 50 45 85 55 90 65 12

50 45 60 55 85 65 12 $\boxed{90}$

pass1 : o/p

pass2 o/p → 45 50 55 60 65 12 85 90

pass3 o/p → (45 50 55 60 12 65 85 90) → ans

Question

#Q. Consider the following recurrence relation (T(n)):

$$T(n) = 9T\left(\frac{n}{3}\right) + C$$

What is the time complexity of above recurrence relation?

(A) $\theta(\log n)$ ✗

(B) $\theta(n^2 \log n)$

(C) $\theta(n^2)$ ✓

(D) $\theta(n^3)$

C

Soln :-

$$T(n) = 9T(n/3) + c$$

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

$$\left.\begin{array}{l} a = 9 \\ b = 3 \\ f(n) = c \end{array}\right\} \checkmark$$

Case 1 :- Is $c = O\left(n^{\log_3 9 - \varepsilon}\right)$, some $\varepsilon > 0$?

$$c = O\left(n^{(2-\varepsilon)}\right), \text{ some } \varepsilon > 0 \text{ ?}$$

$$T(n) = \Theta\left(n^{\log_3 9}\right) = \underline{\Theta(n^2)}$$

True

Question

#Q. Let's suppose you are given an array of n elements in which, the few elements in the beginning are $ and remaining elements are @, then what is the complexity of most efficient algorithm to find the first @ symbol?

A $\theta(n \log n)$

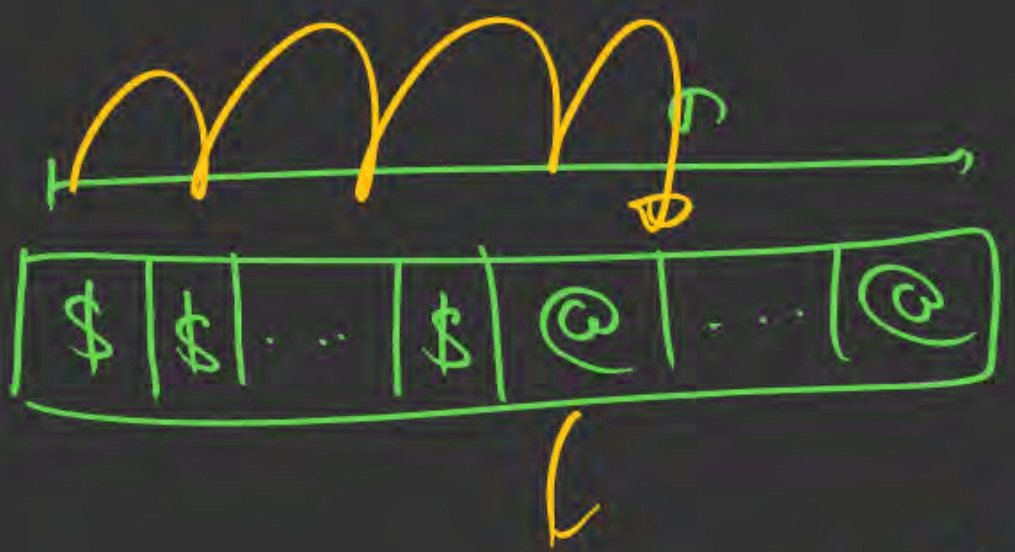
B $\theta(n)$

C $O(1)$

D $\theta(\log n)$

D

Soln :-



1) Linear Search   WC : $O(n)$

2) Binary Search $\rightarrow$ WC : $O(\log_2 n)$

Question

#Q. Consider the following array with 98 as the first element, all other elements can be in any order.

{98, 66, 77, 105, 100, 96, 136, 64}

Quick sort partition algorithm is used by choosing 1^{st} elements as pivot, then what is the total number of arrangements of integer is possible to preserve the effect of first pass of partition algorithm?

Soln: $A = \{98, 66, 77, 105, 100, 96, 136, 64\}$

pivot

				98			

77, 66, 64, 96 　　 105, 136, 100

$4!$ 　　 $3!$

$4! \times 3!$

$24 \times 6 = \boxed{144}$

(Q) A = [10 20 50 60 70 65 55 25 15]

after 4th pass → position of 70? (Selection Sort)

Soln:

P1 : [10 20 50 60 70 65 55 25 15]

P2 : 10 | 20 50 60 70 65 55 25 15

P3 → 10 15 | 50 60 70 65 55 25 20

P4 : → 10 15 20 | 60 70 65 55 25 50

→ 10 15 20 | 25 70 65 55 60 50 → pass 4 O/P

ans = 5

Question

#Q. Let the number of 84, 98, 142, 284, 362, 999, 738, 393 and 561 be sorted using radix sort what will be 8th number in the sequence of number after ~~sorting the~~ 3rd ~~digit~~ _______?

pass?

738

Shortcut :- max digits → $\boxed{3}$ → passes to sort

84 98 142 284 362 393 561 738 999

Radix Sort

Sol: A = 84 98 142 284 362 999 738 393 561

0	1	2	3	4	5	6	7	8	9
	561	142	393	84				98	999
		362		284				738	

Pass 1 → O/p: [561 142 362 393 84 284 98 738 999]

0	1	2	3	4	5	6	7	8	9
			738	142		561		84	393
						362		284	98
									999

pass 2 O/p: [738 142 561 362 84 284 393 98 999]

0	1	2	3	4	5	6	7	8	9
84	142	284	362		561		738		999
98			393						

84 98 142 284 362 393

561 738 999

Thank

THANK

Keep Hustling!