Name: GOVIND KUMAR
Branch: IT
Roll No: 11912057

Q2. Ans: Let 'A' is array of size '5', and elements are 8, 5, 7, 3, 2.

	0	1	2	3	4
A =	8	5	7	3	2

, n = 5

(a) Bubble sort

For 1st pass

8]	5	5	5	5
5	8]	7	7	7
7	7	8]	3	3
3	3	3	8]	2
2	2	2	2	(8)

} 4 comp & 4 swaps

8 at sorted position.

For 2nd pass

5]	5	5	5
7]	7]	3	3
3	3	7]	2
2	2	2	(7)
8	8	8	(8)

} 3 comp & 3 swaps

7 & 8 are present at sorted position.

For 3rd pass

5]	3	3
3]	5]	2
2	2]	(5)
7	7	(7)
8	8	(8)

} 2 comparisons & 2 swaps.

5, 7, & 8 are present at sorted position.

For 4th pass

3]	2
2]	3
5	5
7	7
8	8

} 1 comparisons & 1 swaps

all elements are present at sorted position.

No of passes = 4 i.e (n-1) passes

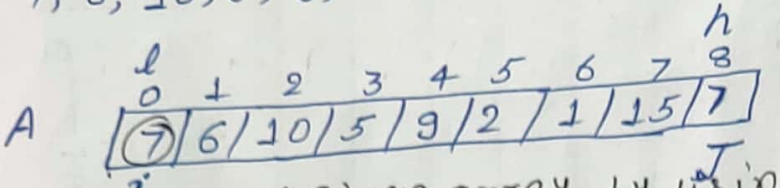
No of comparisons = $1+2+3+\dots+(n-1) = \frac{n(n-1)}{2} = O(n^2)$

Max No of swaps = $1+2+3+4 \to 1+2+\dots+(n-1) = O(n^2)$

hence time complexity of Bubble sort is $O(n^2)$.

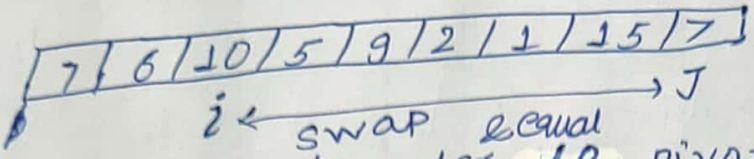
(6) Quick Sort:

Let A is array of size 9, and elements are 7, 6, 10, 5, 9, 2, 1, 15, 7.



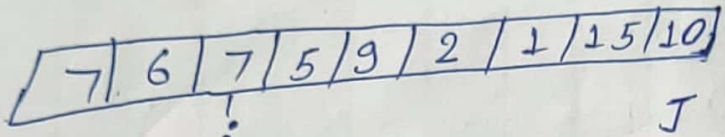
Soot given array by using partition method

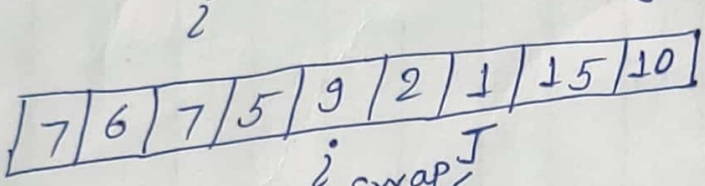
Pivot = A[l] = 7

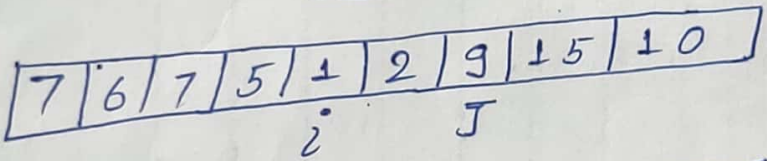


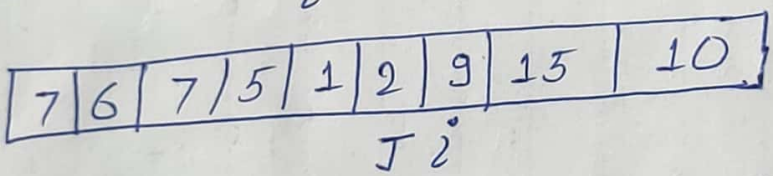
swap equal

If any element greater to pivot then i stop &
If any element less than pivot then J will be stop.

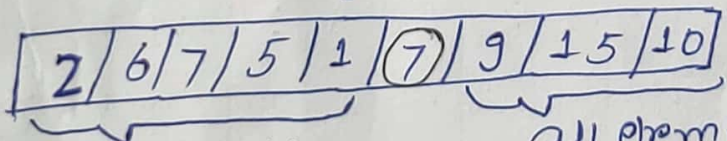








here i > J then swap pivot with A[J],



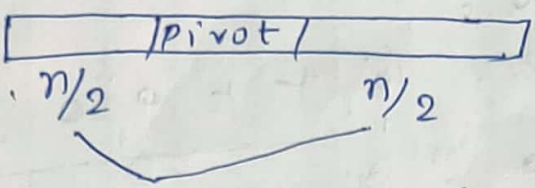
all element in left side of pivot is small & equal to pivot

all element in Right side of pivot is greater than pivot.

Again use partition of left of pivot & Right of pivot Recursively.

⇒ Time complexity of Quick sort:

① Best case time analysis of Quick sort.
It occurs when list is divided into 2 part after placement of pivot at its proper location.



2 sub problems exist of size $n/2$

Recurence Relation

$$T(n) = \begin{cases} c & \text{if } n=1 \text{ b/c signal element are always sorted.} \\ 2T(n/2) + cn & \text{if } n>1 \end{cases}$$

(2 → no of sub problems)

Solving above equation using substitution method

$$T(n) = 2T(n/2) + cn \quad \rightarrow (i)$$

$$T(n/2) = 2T(n/4) + cn/2$$

$$T(n/4) = 2T(n/8) + cn/8$$

$$T(n) = 8T(n/8) + cn + 2cn$$

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

$$\vdots$$

Kth terms $\quad T(n) = 2^K T\left(\frac{n}{2^K}\right) + Kcn$

Let $n = 2^K$

$$\log_2 n = \log_2 2^K$$

$$K = \log_2 n$$

$$T(n) = nT(1) + cn\log_2 n$$

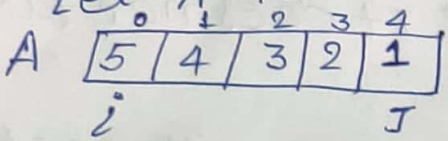
$$T(n) = \Omega(n\log_2 n)$$ → Best case time complexity of quick sort.

↳ omega is use b/c this is best case

(2) worst case time analysis of quick sort

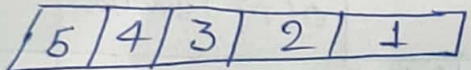
It occurs when the list is already sorted in Ascending order.

Let A is a array of size 5 & elements are



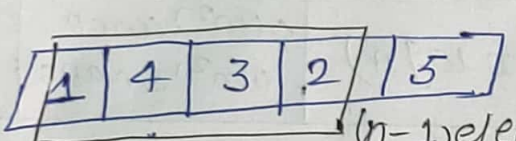
If any element greater & equal to pivot then i should be stop & If any element less than pivot then J should be stop

pivot = 5



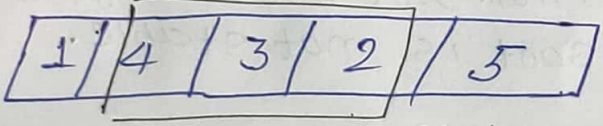
∞ n comparisons

here i > J then swap pivot with A[J]

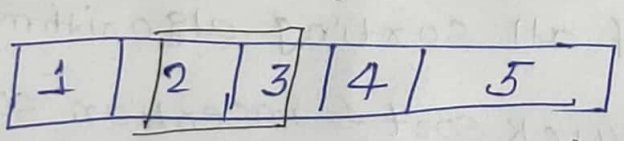


(n-1) element ie (n-1) comparisons

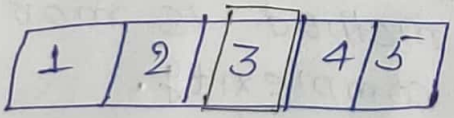
here again i > J then swap pivot with A[J]



(n-2) element ie (n-2) comparison



(n-3) element ie (n-3) comparison



(n-4) element ie (n-4) comparison

the total no of comparisons

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

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

$T(n) = O(n^2)$ ⊗, worst case time complexity of Quick sort.

Average case time complexity of Quick sort is also $\theta(n \log_2 n)$.

⇒ category of both bubble sort & Quick sort is in-place algorithum because It does not take extra space.

⇒ Comparing bubble sort & Quick sork with insertion sort & merge sort.

class	Bubble Sort	Quick Sort	insertion sort	merge sort
worst-case performance	$O(n^2)$ comp $O(n^2)$ swaps	$O(n^2)$	$O(n^2)$ comp $O(n^2)$ swaps	$O(n \log n)$
Best-case performance	$O(n)$ comp $O(1)$ swaps	$O(n \log n)$ (simple partition) or $O(n)$ (three-way partition & equal key	$O(n)$ comp $O(1)$ swaps	$O(n \log n)$ typical, $O(n)$ natural variant
Average performance.	$O(n^2)$ comp $O(n^2)$ swaps	$O(n \log n)$	$O(n^2)$ comp $O(n^2)$ swaps	$O(n \log n)$

Some important points about above four sorting algorithms.

(i) Bubble sort, insertion sort and merge sort allare stable but Quick sort is not stable.

(ii) only Bubble sort and insertion sort are Adaptive out of all sorting algorithms.

(iii) Bubble sort, Quick sort & insertion sort are in-place sorting algorithms but merge sort is outplace sorting method is more efficient in term of time complexity.