Date ______________

Expt. No. ______________ Page No. ______________

Bubble sort

10, 6, 4, 7, 2

first check 1st element with adjace 10 > 6 swap similarly go on ~~to find the correct position of 10~~ till the last element 6 4, 7, 2, 10 next step compare 6 with 4 & find it correct position and compare till the last ~~second~~ element. Similarly go on

Step-1 5 ~~Step~~

10, 6, 4, 7, 2 → 6, 4, 7, 2, 10

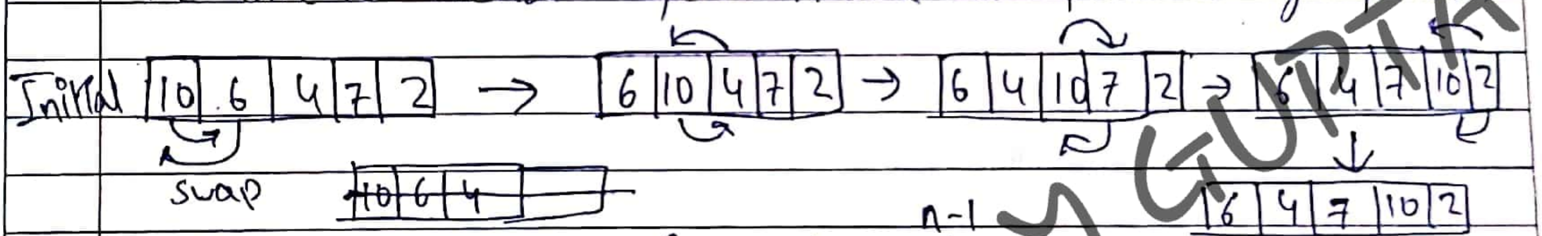
4, 7, 6, 2, 10 → 4 7, 2, 6, 10

4, 2, 7, 6, 10 → 4 2, 6, 7, 10

2, 4, 6, 7, 10

sorted

complexity – $O(n^2)$ as we for each element we have to iterate for n time to place it right position



similarly check for all the ~~n~~ n-1 times for n elements

hence its complexity is $O(n^2)$

since we does not use any extra. space thus it is inplace sorting algorithm

Teacher's Signature ______________

Quick Sort

Method - split and win algorithm technique into which a pivotal element becomes the focal point of division around the given array

let - 1, 3, 8, 9, 4, 5, 7

Pivotal number = 7

Compare 7 with 1 as ~~7>1 no swapping~~ 7 its the first value

no : 8 > 7 no swapping

3<7 Swapp 3, 8

1 3 8 9 4 5 7

9 > 7 no swap

4<7 swap 8 & 4

~~1 3 8 4 9 1 3 4 9 8 7~~

1, 3, 4, 9, 8, 5, 7

5 < 7

swap 9 & 5

1 3 4 5 8 9 7

9 > 7

swap 8, 7

1 3 4 5 7 9 8

now 7 has reach its appropriate place

no divide the array through recursive and again Quick sort 1, 3, 4, 5 & 9 8

1,3,4,5 now 5 become pivot element & checks every element

9,8 now 8 becomes pivot element & checks every elemn

8 < 9 thus swap

8, 9 and now combine all subarrays

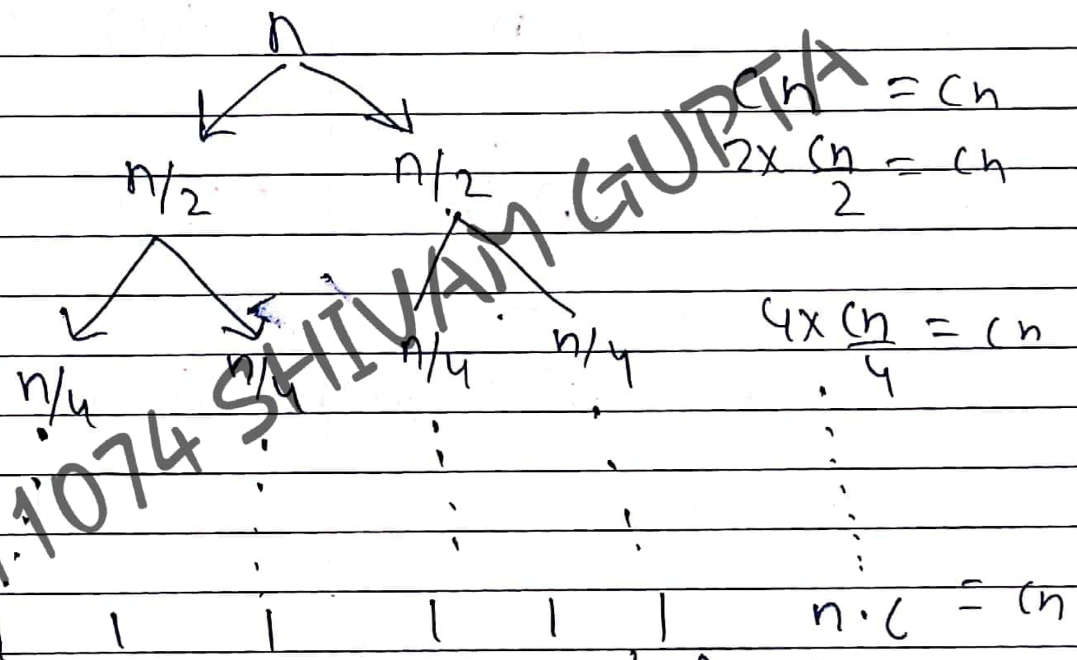
we get 1 3, 4, 5, 7 8, 9

Since we need extra space for sorting divided array hence it is Out of place sorting algorithm

complexity

best case O(nlogn)

worst case O(n²)
Average case O(nlogn)
space complexity O(nlogn)



using big O notation we get (nlog₂n)

Comparison

Quick Sort
Insertion Sort

Comparison

Algorithm	Timecomplexity			Space complexity
~~case~~	Best	Average	worst	worst
Quick sort	~~$\Omega(n)$~~ $\Omega n\log(n)$	$\theta(n\log n)$	$O(n^2)$	$O(n\log n)$
Bubble sort	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$	$O(1)$
Insertion sort	$\Omega(n)$	$\theta(n^2)$	$O(n^2)$	$O(1)$
Merge sort	$\Omega(n\log(n))$	$\theta(n\log n)$	$O(n\log n)$	$O(n)$