Os-Assignment	Anuprect Singh
	11911063
	cse V
9-1- Insertion Sort	at at a for the
Algorithm :-	/
Insertion Sort	(int arr [], size)
ک'	
forsi -> s	ize
j= 201 [133
K=3-1	
while (K>=	-0 & & arr [K] >j)
four [K+D	= arr(K)
K 5	
	194
3	
	Anur
3	Anup
29 2001 [K+1]=j 3	
since Insertion hat it made	
era [K+1]=j 3 Since Insertion Sort is modifily inserting the lower plane	ying the original avora
Ever [K+1]=j Since Insertion Sort is modifily inserting the lower cleme in the original array only	ying the original avora
era [K+1]=j Since Insertion Sort is modified by inserting the lower plane in the original array only require any extra space	ying the original avora ent at the right place y. Thus it does not
era [K+1]=j Since Insertion Sort is modification for the lower clame on the original array only require any extra space. In-Place Sorting" Algorithm	ying the original avora ent at the oright place y. Thus it does not Hence it is an
era [K+1]=j Since Insertion Sort is modifily inserting the lower cleme or the original array only require any extra space. "In-Place Sorting" Algorithm .: Space Complexity	ying the original avora ent at the right place y. Thus it does not Hence it is an = 0(1).
err [K+1]=j Since Insertion Sort is modificate in the lower plane in the original array only require any extra space. In-Place Sorting" Algorithm .: Space Complexity Basic operation takes place	ying the original avora ent at the right place y. Thus it does not Hence it is an = 0(1).
since Insertion Sort is modifically inserting the lower clans on the original array only require any extra space. In-Place Sorting" Algorithm .: Space Complexity Basic operation takes place i) Comparison ii) Susppin	ying the original avorant at the right place y. Thus it does not Hence it is an = 0(1).
since Insertion Sort is modifically inserting the lower clans on the original array only require any extra space. In-Place Sorting" Algorithm .: Space Complexity Basic operation takes place i) Comparison ii) Susppin	ying the original avorant at the right place y. Thus it does not Hence it is an = 0(1).
since Insertion Sort is modifically inserting the lower clans on the original array only require any extra space. In-Place Sorting" Algorithm .: Space Complexity Basic operation takes place i) Comparison ii) Susppin	ying the original avorant at the right place y. Thus it does not Hence it is an = 0(1).
era [K+1]=j Since Insertion Sort is modificate in the original array only require any extra space. In-Place Sorting" Algorithm .: Space Complexity Basic operation takes place i) Comparison ii) Suspin.	ying the original avorant at the right place y. Thus it does not Hence it is an = 0(1).
are [K+1]=j Since Insertion Sort is modificate by inserting the lower cleme of the original array only require any extra space. In-Place Sorting" Algorithm .: Space Complexity Basic operation takes place i) Comparison ii) Swapping 1 Best case i.e. the array is the algorithm only comparison	ying the original avorant at the right place y. Thus it does not Hence, it is an = 0(1). in the g algorithm: already sorted. res 'n' elements.
Since Insertion Sort is Modificy inserting the lower cleme on the original array only require any extra space. In-Place Sorting" Algorithm .: Space Complexity Basic operation takes place i) Comparison ii) Susppin	ying the original avorant at the right place y. Thus it does not Hence, it is an = 0(1). in the algorithm: already sorted. res 'n' elements.

Ans-2- guick sort:
Divide and Conquer Algorithm.
Time Complexity
1) Worst Case:
$T(n) = O(n^2)$
2.) Avg. cose
$T(n) = O(n \log n)$
3) Best Case
$T(n) = O(n \log n)$
→ Bulble Sort
0 0 1
For n elements, (no) comparisones are done
T(n) = 1+2+3 . — $(n-1)$
$T(n) = \frac{n(n-1)}{2} = \frac{n^2 - n}{2}$
Anupre Anupre
> Both guick Sort and Bubble Sort algorithm are "In-Place" Algorithm
"In-Place" Algorithm
> Bubble sort is efficient for small size arrays.
Time complexity for Merge Sort > 0 (nlogn)
1000
Time complexity for Insertion Sout -> O(n2)