Experiment No.1
Insertion Sort
Date of Performance:
Date of Submission:

Title: Insertion Sort

Aim: To implement Selection Comparative analysis for large values of 'n'

Objective: To introduce the methods of designing and analysing algorithms

Theory:

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

Example:

Algorithm and Complexity:

INSERTION-SORT(A)	*cost*	*times*
1 **for** $j = 2$ **to** $A.length$	c_1	n
2 $key = A[j]$	c_2	$n-1$
3 // Insert $A[j]$ into the sorted sequence $A[1..j-1]$.	0	$n-1$
4 $i = j-1$	c_4	$n-1$
5 **while** $i > 0$ and $A[i] > key$	c_5	$\sum_{j=2}^{n} t_j$
6 $A[i+1] = A[i]$	c_6	$\sum_{j=2}^{n} (t_j - 1)$
7 $i = i-1$	c_7	$\sum_{j=2}^{n} (t_j - 1)$
8 $A[i+1] = key$	c_8	$n-1$

Implementation:

Code:

```
#include<stdio.h>

#include<conio.h>

void main()

{

   int a[]={5,4,3,2,6,8};

   int n=6;

   int temp,i,j;

   for(int i=0;i<n;i++)

   {

      temp=a[i];
```

```
        j=i-1;
        while(j>=0 && a[j]>temp)
        {
            a[j+1]=a[j];
            j--;
        }
        a[j+1]=temp;
    }
    printf("Sorted array:");
    for(i=0;i<n;i++)
    {
        printf(" %d ",a[i]);
    }
getch();
}
```

Output:

Conclusion: In conclusion, the insertion sort algorithm effectively sorts an array of integers in ascending order. This sorting method iterates through the array, comparing each element with the ones preceding it. If an element is smaller than the preceding elements, it shifts those elements to the right to make space for the smaller element, effectively inserting it into its correct position. This process continues until the entire array is sorted. The algorithm has a time complexity of O(n^2) in the worst case scenario, making it suitable for small to medium-sized arrays. Additionally, insertion sort is an in-place sorting algorithm, meaning it doesn't require additional memory space for sorting.