



KIET Group of Institutions, Ghaziabad

Department of Computer Applications

(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

DAA Lab

Experiment-No.2

Objective: Implement the INSERTION-SORT algorithm to sort the given list of N numbers and plot graph.

Scheduled Date:	Compiled Date:	Submitted Date:
14-8-2020	18-8-2020	30-8-2020

Algorithm:

INSERTION-SORT(A)

```
1   for  $j \leftarrow 2$  to  $\text{length}[A]$ 
2       do  $\text{key} \leftarrow A[j]$ 
3       Insert  $A[j]$  into the sorted sequence  $A[1 \dots j-1]$ .
4        $i \leftarrow j-1$ 
5       while  $i > 0$  and  $A[i] > \text{key}$ 
6           do  $A[i+1] \leftarrow A[i]$ 
7            $i \leftarrow i-1$ 
8        $A[i+1] \leftarrow \text{key}$ 
```

Program file insertion_sort.c :

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<alloc.h>
int count=0;
void main()
{
    void getdata(int[10],int);
    void putdata(int[10],int);
    void insertion_sort(int a[],int);
    int i,a[100],n;
    clrscr();
    printf("enter the value of n\n");
    scanf("%d",&n);
    getdata(a,n);
    printf("\nbefore soring\n");
    putdata(a,n);
    insertion_sort(a,n);
}
```



KIET Group of Institutions, Ghaziabad

Department of Computer Applications

(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

DAA Lab

```
printf("\nafter sorting\n");
putdata(a,n);
printf("\n for n = %d value of count is %d",n,count);
getch();
}
void getdata(int a[10],int n)
{
    int k;
    printf("enter the value for sorting\n");
    for(k=1;k<=n;k++)
    {
        scanf("%d",&a[k]);
    }
}
void putdata(int a[10], int n)
{
    int k;
    for(k=1;k<=n;k++)
    {
        printf("%d\t",a[k]);
    }
    printf("\n");
}
void insertion_sort(int a[],int n)
{
    int key,j,i;
    count++;

    for(j=2;j<=n;j++)
    {
        count++;
        key=a[j];
        count++;
        i=j-1;
        count++;
        while(i>0 && a[i]>key)
        {
            count++;
            count++;
            a[i+1]=a[i];
            count++;
            i=i-1;
            count++;
        }
    }
}
```



KIET Group of Institutions, Ghaziabad

Department of Computer Applications

(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

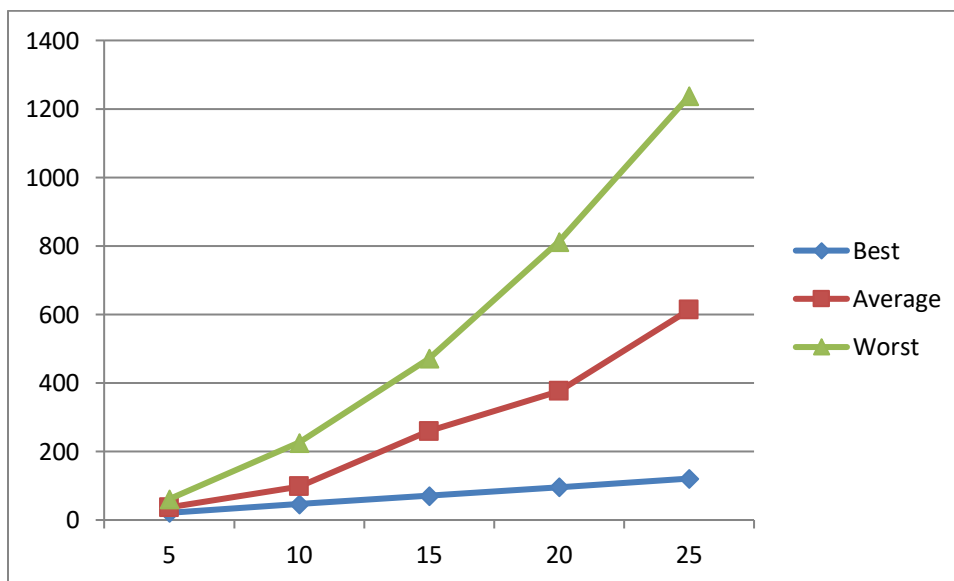
DAA Lab

```
}  
a[i+1]=key;  
count++;  
count++;  
  
}  
  
}
```

Output

Inputs	Best Case	Average Case	Worst Case
5	21	37	61
10	46	98	226
15	71	259	471
20	96	376	812
25	121	613	1237

Graph





KIET Group of Institutions, Ghaziabad

Department of Computer Applications

(An ISO – 9001: 2015 Certified & 'A' Grade accredited Institution by NAAC)

Design and Analysis of Algorithm

RCA 352: Session 2020-21

DAA Lab

Conclusion

Case	Running Time : Growth of Function mathematically	Running Time : Growth of Function after observing graph
Best Case	$O(n)$	$O(n)$
Average Case	$O(n^2)$	$O(n^2)$
Worst Case	$O(n^2)$	$O(n^2)$