

PRACTICAL-2

Aim: Implement and analyze algorithms given below.(Compare them)

2.1 Bubble Sort:

PROGRAM:

```
#include <iostream>

using namespace std;

int main()
{
    int arr[50];
    int n,i,j,temp,c=0;
    bool swapped;

    cout << "Enter size of array : ";
    cin >> n;

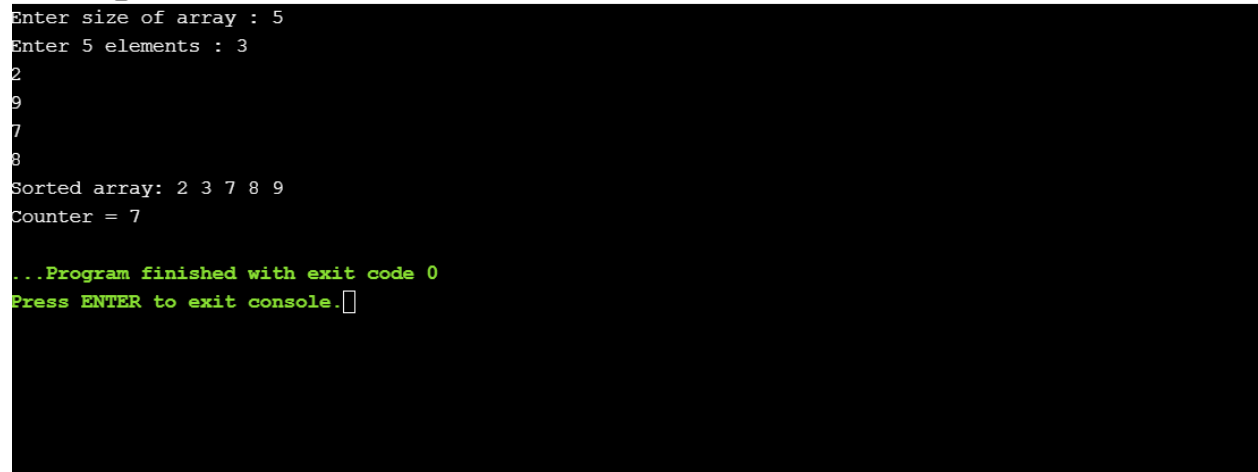
    cout << "Enter " << n << " elements : ";
    for (i=0; i < n; i++)
        cin >> arr[i];

    for (i = 0; i < n-1; i++) {
        swapped = false;
        for (j = 0; j < n-i-1; j++) {
            c++;
            if (arr[j] > arr[j+1]) {
                temp = arr[j];
```

```
        arr[j] = arr[j+1];
        arr[j+1] = temp;
        swapped = true;
    }
}
if (swapped == false)
    break;
}
```



```
cout << "Sorted array:";
for (i=0; i < n; i++)
    cout << " " << arr[i];
cout << "\nCounter = " << c;
return 0;
}
```

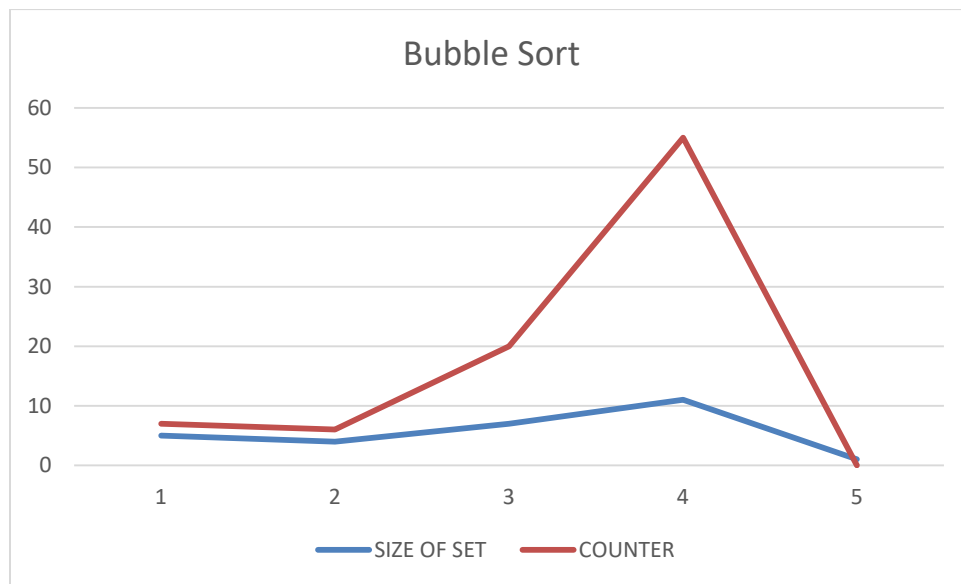
OUTPUT:

```
Enter size of array : 5
Enter 5 elements : 3
2
9
7
8
Sorted array: 2 3 7 8 9
Counter = 7

...Program finished with exit code 0
Press ENTER to exit console.
```

ANALYSIS TABLE:

SIZE OF SET	COUNTER
5	7
4	6
7	20
11	55
1	0

GRAPH:**CONCLUSION:**

In this practical we can conclude that bubble sort algorithm behavior is non-linear.

Aim: Implement and analyze algorithms given below.(Compare them)

2.2 Selection Sort:

PROGRAM:

```
#include<iostream>

using namespace std;

int counter=0;

int main()
{
    int length;
    cout<<"Enter the size of array :";
    cin>>length;
    int arr[length];

    for (int i=0;i<length;i++)
    {
        arr[i]=i+10; cout<<"Enter element"<<(i+1)<<":";
        cin>>arr[i];
    }

    for(int m=0;m<length;m++)
    {
        int minimum=m;
        float temp;
        for(int n=m+1;n<length;n++)
        {
            counter++;
```

```
        if(arr[n]<arr[minimum])
        {
            minimum=n;
        }
    }
    temp=arr[m];
    arr[m]=arr[minimum];
    arr[minimum]=temp;
}
cout<<endl<<endl<<"Sorted Array is:  " <<endl;
for(int i=0;i<length;i++)
cout<<arr[i]<<" ";
cout<<"\nCounter : " <<counter<<endl;

}
```

OUTPUT:

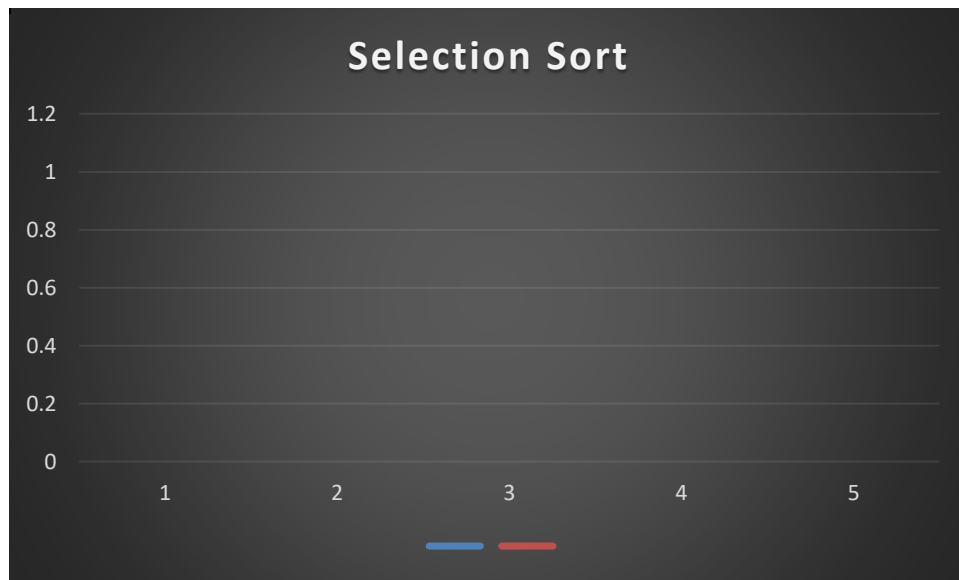
```
Enter the size of array :5
Enter element1:31
Enter element2:12
Enter element3:21
Enter element4:9
Enter element5:0

Sorted Array is:
0 9 12 21 31
Counter :10

...Program finished with exit code 0
Press ENTER to exit console.
```

ANALYSIS TABLE:

Size Of Array	Counter
5	10
3	3
10	45
12	66
9	36

GRAPH:**CONCLUSION:**

In this practical we analyzed the graph of selection sort

Aim: Implement and analyze algorithms given below.(Compare them)

2.3 Insertion Sort:

PROGRAM:

```
#include<iostream>

using namespace std;

int counter=0;

int main()
{
    float temp;
    int i,j,length;
    cout<<"\nEnter the size of Array:";
    cin>>length;
    int arr[length];
    cout<<"Enter Array elements: "<<endl;
    for(i=0;i<length;i++)
    {
        cin>>arr[i];
    }

    for(i=1;i<length;i++)
    {
        temp=arr[i];
        j=i-1;
        counter++;
        while((temp<arr[j])&&(j>=0))
        {
            counter++;
            arr[j+1]=arr[j];
```

```
        j=j-1;
    }
    arr[j+1]=temp;
}
cout<<"\nArray after Being sorted : \n";
for(i=0;i<length;i++)
{
    cout<<arr[i]<<" ";
}
cout<<"\n\nCounter : "<<counter<<endl;
return 0;
}
```

OUTPUT:

```
Enter the size of Array:3
Enter Array elements:
19
16
12

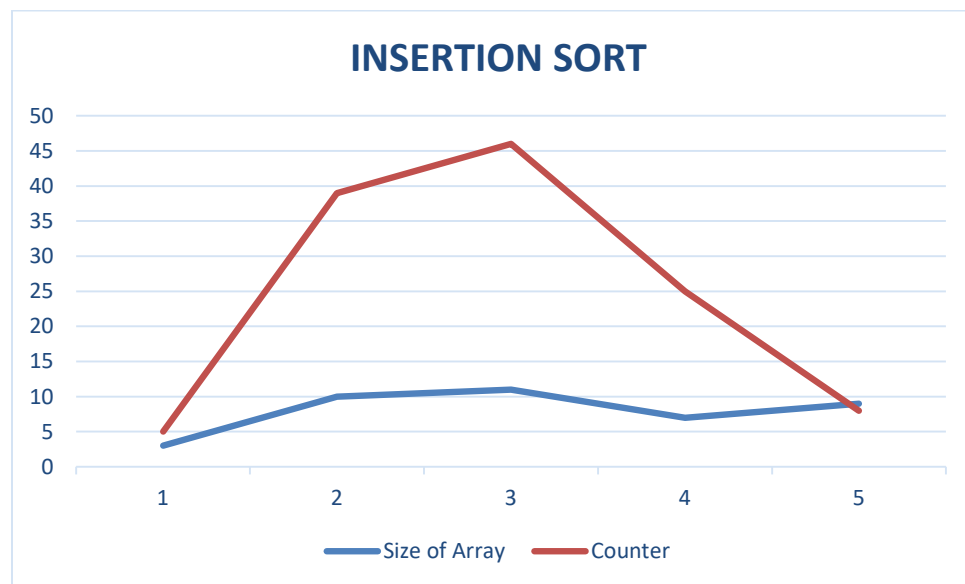
Array after Being sorted :
12 16 19

Counter :5

...Program finished with exit code 0
Press ENTER to exit console.□
```


ANALYSIS TABLE:

Size of Array	Counter
3	5
10	39
11	46
7	25
9	8

GRAPH:**CONCLUSION:**

In this practical we analyzed the behavior of Insertion Sort graph.