CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH

Computer Science & Engineering

NAME: PARTH NITESHKUMAR PATEL

ID: 19DCS098

SUBJECT: DESIGN AND ANALYSIS OF ALGORITHM

CODE: CS 351

PRACTICAL-2

AIM:

IMPLEMENT AND ANALYZE THE FOLLOWING ALGORITHMS.

2.1 Bubble Sort

PROGRAM CODE:

```
#include<iostream>
using namespace std;

static int counter=0;

void swap(int *x,int *y){
    int temp=*x;
    *x=*y;
    *y=temp;
}
void bubbleSort(int arr[],int size){
    bool swapped;

    for(int i=0;i<size-1;i++)
    {
        swapped=false;

        for(int j=0;j<size-1-i;j++)
        {
            counter++;
```

```
            if(arr[j]>arr[j+1])
            {
                swap(&arr[j],&arr[j+1]);
                swapped=true;
            }
        }

        if(swapped==false)
        break;
    }
}

void printArray(int arr[],int size){

    cout<<" The sorted array is : "<<endl;
    for(int i=0;i<size;i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
}
int main()
{
    int size;

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

    int arr[size];

    cout<<"Enter the elements : ";
    for(int i=0;i<size;i++){
        cin>>arr[i];
    }
```

```
    bubbleSort(arr,size);
    printArray(arr,size);

    cout<<"COUNTER: "<<counter<<endl;
    cout<<endl;
    cout<<"PARTH PATEL 19DCS098"<<endl;

    return 0;
}
```

OUTPUT:

```
Enter the size of array : 10
Enter the elements : 9 1 7 3 4 5 7 6 5 8
 The sorted array is :
1 3 4 5 5 6 7 7 8 9
COUNTER: 35

PARTH PATEL 19DCS098
```

2.2 Selection Sort

PROGRAM CODE:

```
#include<iostream>
using namespace std;

static int counter=0;

void swap(int *x,int *y){
    int temp=*x;
    *x=*y;
    *y=temp;
}

void selectionSort(int arr[], int n)
{
    int i, j, min_index;


    for (i = 0; i < n-1; i++)
    {

        min_index = i;
        for (j = i+1; j < n; j++){
            counter++;
        if (arr[j] < arr[min_index])
            min_index = j;
        }
```

```
        // Swap the found minimum element with the first element
        swap(&arr[min_index], &arr[i]);
    }
}

void printArray(int arr[],int size){

    cout<<" The sorted array is : "<<endl;
    for(int i=0;i<size;i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
}
int main()
{
    int size;

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

    int arr[size];

    cout<<"Enter the elements : ";
    for(int i=0;i<size;i++){
        cin>>arr[i];
    }

    selectionSort(arr,size);
    printArray(arr,size);

     cout<<"COUNTER: "<<counter<<endl;
    cout<<endl;
    cout<<"PARTH PATEL 19DCS098"<<endl;
```

```
    return 0;
}
```

OUTPUT:

```
Enter the size of array : 10
Enter the elements : 10 9 8 7 6 5 15 13 12 20
 The sorted array is :
5 6 7 8 9 10 12 13 15 20
COUNTER: 45

PARTH PATEL 19DCS098
```

2.3 Insertion Sort

PROGRAM CODE:

```
#include<iostream>
using namespace std;

static int counter=0;

void swap(int *x,int *y){
    int temp=*x;
    *x=*y;
    *y=temp;
}

void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i - 1;

        while (j >= 0 && arr[j] > key)
        {
            counter++;
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
```

```
}

void printArray(int arr[],int size){

    cout<<" The sorted array is : "<<endl;
    for(int i=0;i<size;i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
}

int main()
{
    int size;

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

    int arr[size];

    cout<<"Enter the elements : ";
    for(int i=0;i<size;i++){
        cin>>arr[i];
    }

    insertionSort(arr,size);
    printArray(arr,size);

     cout<<"COUNTER: "<<counter<<endl;
    cout<<endl;
    cout<<"PARTH PATEL 19DCS098"<<endl;

    return 0;
```

```
}
```

OUTPUT:

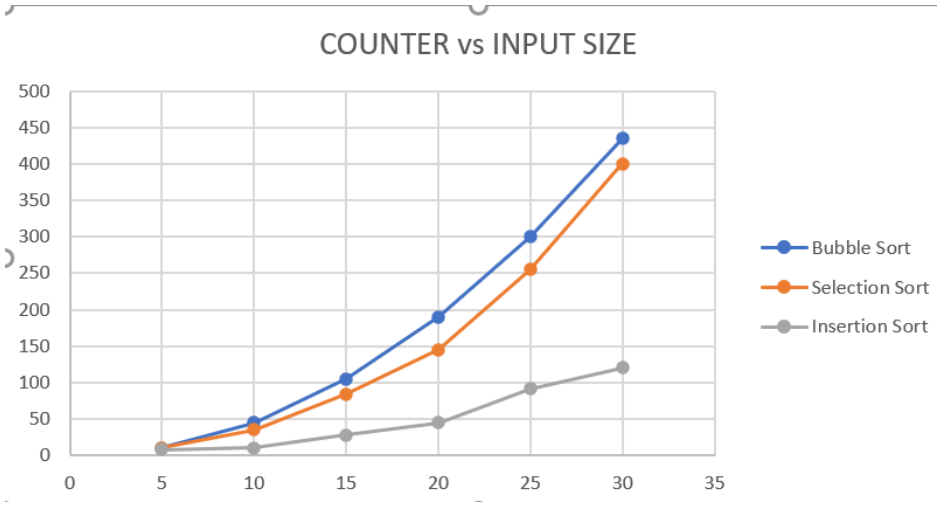
```
Enter the elements : 100 30 1 6 5 2 80 50 22 10
 The sorted array is :
1 2 5 6 10 22 30 50 80 100
COUNTER: 24

PARTH PATEL 19DCS098
```

ANALYSIS TABLE:

INPUT SIZE	COUNTER		
	BUBBLE SORT	SELECTION SORT	INSERTION SORT
5	10	10	8
10	45	35	10
15	105	84	28
20	190	145	45
25	300	255	91
30	435	400	120

GRAPH:



CONCLUSION:

In this practical we learnt about some of the sorting techniques and also, we analysed the difference in the complexity of the algorithms.