NAME:- SAGAR ADHIKARI
ROLL NO:- PUR078BEI034
FACULTY:- ELECTRONICS(BEI)

DSA LAB SHEET NO. 6

TITLE:- IMPLEMENTATION OF SORTING (Bubble,Selection,Insertion)

THEORY:-

Sorting:-A Sorting Algorithm is used to rearrange a given array or list of elements according to a comparison operator on the elements.
Bubble Sort:- Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. This algorithm is not suitable for large data sets as its average and worst-case time complexity is quite high.
Selection Sort:-Selection sort is a simple and efficient sorting algorithm that works by repeatedly selecting the smallest (or largest) element from the unsorted portion of the list and moving it to the sorted portion of the list.
Insertion Sort:-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.

1.Bubble Sort

PROGRAM CODE:-

```
#include <iostream>
#include <chrono>
#include <cstdlib>
using namespace std;
void swapp(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
void bubbleSort(int A[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        for (int j = 0; j < n - i - 1; j++)
        {
            if (A[j] > A[j + 1])
                swapp(&A[j], &A[j + 1]);
        }
    }
}
```

DSA LAB SHEET NO. 6

```
int main()
{
    int A[100000], n, i;
    do
    {
        cout << "Enter n:- ";
        cin >> n;
        for (int i = 0; i < n; i++)
        {
            A[i] = rand();
        }

        auto t1 = chrono::high_resolution_clock::now();
        bubbleSort(A, n);
        auto t2 = chrono::high_resolution_clock::now();
        auto duration = chrono::duration_cast<chrono::microseconds>(t2 - t1);
        cout << "Time=" << duration.count() << " microseconds" << endl;

    } while (n != 0);
    return 0;
}
```

OUTPUT:-

```
Enter n:- 10
Time=1 microseconds
Enter n:- 100
Time=89 microseconds
Enter n:- 1000
Time=3081 microseconds
Enter n:- 10000
Time=331608 microseconds
Enter n:- 100000
Time=37144552 microseconds
```

NAME:- SAGAR ADHIKARI
ROLL NO:- PUR078BEI034
FACULTY:- ELECTRONICS(BEI)

2.Selection Sort

PROGRAM CODE:-

```
#include <iostream>
#include <chrono>
#include <cstdlib>
using namespace std;
void swapp(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
void selectionSort(int A[], int n)
{
    for (int i = 0; i < n; i++)
    {
        int smallest = A[i];
        int position = i;
        for (int j = i + 1; j < n; j++)
        {
            if (A[j] < smallest)
                smallest = A[j];
            position = j;
        }
        if (i != position)
        {
            swapp(&A[i], &A[position]);
        }
    }
}

int main()
{
    int A[100000], n, i;
    do
    {
        cout << "Enter n:-\t";
        cin >> n;
        for (int i = 0; i < n; i++)
        {
            A[i] = rand();
        }

        auto t1 = chrono::high_resolution_clock::now();
        selectionSort(A, n);
        auto t2 = chrono::high_resolution_clock::now();
        auto duration = chrono::duration_cast<chrono::microseconds>(t2 - t1);
        cout << "Time=" << duration.count()<< " microseconds"<< endl;
    } while (n != 0);
    return 0;
}
```

OUTPUT:-

```
Enter n:-      10
Time=1 microseconds
Enter n:-      100
Time=26 microseconds
Enter n:-      1000
Time=1438 microseconds
Enter n:-      10000
Time=120870 microseconds
Enter n:-      100000
Time=10657245 microseconds
```

3.Insertion Sort

PROGRAM CODE:-

```
#include <iostream>
#include <chrono>
#include <cstdlib>
using namespace std;
void swapp(int *x, int *y)
{
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

void insertionSort(int A[], int n)
{
    for (int i = 0; i < n; i++)
    {
        int j = i - 1;
        int temp = A[i];
        while (i >= 0 && temp < A[j])
        {
            A[j + 1] = A[j];
            j = j - 1;
        }
        A[j + 1] = temp;
    }
}
int main()
{
    int A[100000], n, i;
    do
    {
        cout << "Enter n:- ";
        cin >> n;
        for (int i = 0; i < n; i++)
        {
            A[i] = rand();
        }

        auto t1 = chrono::high_resolution_clock::now();
        insertionSort(A, n);
```

NAME:- SAGAR ADHIKARI
ROLL NO:- PUR078BEI034
FACULTY:- ELECTRONICS(BEI)

```
        auto t2 = chrono::high_resolution_clock::now();
        auto duration = chrono::duration_cast<chrono::microseconds>(t2 - t1);
        cout << "Time=" << duration.count()<<" microseconds" << endl;

    } while (n != 0);
    return 0;
}
```

OUTPUT:-

```
Enter n:- 10
Time=0 microseconds
Enter n:- 100
Time=64 microseconds
Enter n:- 1000
Time=876 microseconds
Enter n:- 10000
Time=81225 microseconds
Enter n:- 100000
Time=6893454 microseconds
```