**PRACTICAL-2**

**AIM**:

To perform insertion, bubble and selection sort and find their time complexities.

**CODE**:

#include<iostream>

#include<ctime>

using namespace std;

void insertion(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)

{

arr[j+1]=arr[j];

j--;

}

arr[j+1]=key;

}

}

void selection(int arr[],int n){

int i,j,pos,temp;

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

{

pos=i;

for(j=i+1;j<n;j++)

{

if(arr[j]<=arr[pos])

{

pos=j;

}

if(pos!=i)

{

temp=arr[i];

arr[i]=arr[pos];

arr[pos]=temp;

}

}

}

}

void bubble(int arr[],int n)

{

int i,j,temp;

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

{

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

{

if(arr[j+1]<arr[j])

{

temp=arr[j];

arr[j]=arr[j+1];

arr[j+1]=temp;

}

}

}

}

int main(){

int arr[10000],n,i,ch;

clock\_t time;

cout<<"enter size :";

cin>>n;

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

arr[i]=rand();

cout<<arr[i]<<"\t";

}

cout<<"\n 1) insertion \n 2) selection\n 3)=bubble \n 0=exit \n enter your choice:";

cin>>ch;

time=clock();

switch(ch){

case 1:insertion(arr,n);

break;

case 2:selection(arr,n);

break;

case 3:bubble(arr,n);

break;

default : cout<<"wrong choice";

break;

}

time = clock() - time;

cout<<"sorted array is:-";

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

cout<<arr[i]<<"\t";

}

cout <<"Time taken : "<<(double)time/CLOCKS\_PER\_SEC << endl;

return 1;

}

Output:





