**AIM**

To accept an array from the user and search an element using Binary Search technique and sort it using the Bubble and Quicksort methods.

**THEORY**

Searching techniques are used in various applications to look for a particular element in an array and Binary Search is used to do the same. Sorting is to arrange elements or data in an array in a particular order and is being done using Bubble Sort or the Quicksort algorithm.

Linear Search: Linear Search traverses the complete array till we find the desired element. When the element is discovered its index is to be returned.

Average Time Complexity - O(n)

Selection Sort: Starting from the first index as traversing index. Then it traverses through whole array and finds the smallest element and replaces it with the current traversing index. This way it sorts the complete array.

Average Time Complexity - O(n^2)

Insertion Sort: In insertion sort considering index from first element it traverses whole array. Then the elements to the left of current index are sorted in next loop. This way whole array is sorted.

Average Time Complexity - O(n^2)

SOURCE CODE:

/\*

PROG BY ANOM DEVGUN

GR: 21810017

\*/

#include<iostream>

using namespace std;

void dispa(int A[],int n)

{

int l=0;

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

{

cout<<A[l]<<" ";

}

}

int main()

{

char ch;

int n,i=0,j=0,ke=0,k=0;

int temp=0,min=0;

cout<<"Enter the number of elements you want in array:\n";

cin>>n;

int a[n];

cout<<"Enter "<<n<<" numbers\n";

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

{

cin>>a[i];

}

cout<<"1)For selection sorting the input data:\n";

cout<<"2)For insertion sorting the input data:\n";

cout<<">>";

cin>>ch;

switch (ch)

{

case '1':

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

{

min=i;

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

{

if(a[j]<a[min])

{

min=j;

}

}

temp=a[min];

a[min]=a[i];

a[i]=temp;

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

{

cout<<a[k]<<" ";

}

cout<<"\n";

}

break;

case '2':

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

{

ke=a[i];

j=i-1;

while(j>=0 && a[j]<ke)

{

a[j+1]=a[j];

j=j-1;

}

a[j+1]=ke;

dispa(a,i+1);

cout<<"\n";

}

break;

}

return 0;

}