ADA Lab-2

1. Sort a given set of elements using Quick sort. Time analysis to be done by varying the input size and pattern.

**code:**

#include<iostream>

using namespace std;

int count=0;

int pos(int a[],int l,int h){

    int i = l+1;

    int j = h;

    int pivot = a[l];

    while(i<=j){

        while(a[i]<=pivot)

        { count++;

          i++;

        }

        while(a[j]>pivot){

          j--;

          count ++;

        }

        count ++;

        if(i<j)

         swap(a[i],a[j]);

    }

    // if(j<=i)

    swap(a[j],a[l]);

    return j;

}

// int partition(int \*a,int l,int h) //taking last element as pivot

// {

//     int p=a[h];

//     int i=l;

//     int j=h-1;

//     while(i<=j)

//     {

//         while(a[i]<p)

//         {

//             i++;

//         }

//         while(a[j]>p)

//         {

//             j--;

//         }

//         if(i<j)

//         {

//             swap(a[i],a[j]);

//         }

//     }

//     swap(a[i],a[h]);

//     return i;

// }

void quickSort(int a[],int l,int h){

    if(l<h){

        int p = pos(a,l,h);

        quickSort(a,l,p-1);

        quickSort(a,p+1,h);

    }

}

int main(){

   int n;

    cout<<"Enter size of array\n";

    cin>>n;

    int arr[n];

    cout<<"Enter "<<n<<" numbers\n";

    for(int i=0;i<n;i++){

        cin>>arr[i];

    }

    cout<<"Entered array is : ";

    for(int i=0;i<n;i++){

        cout<< arr[i]<<" ";

    }

    cout<<endl;

    quickSort(arr,0,n-1);

    cout<<"Sorted array is : ";

    for(int i=0;i<n;i++){

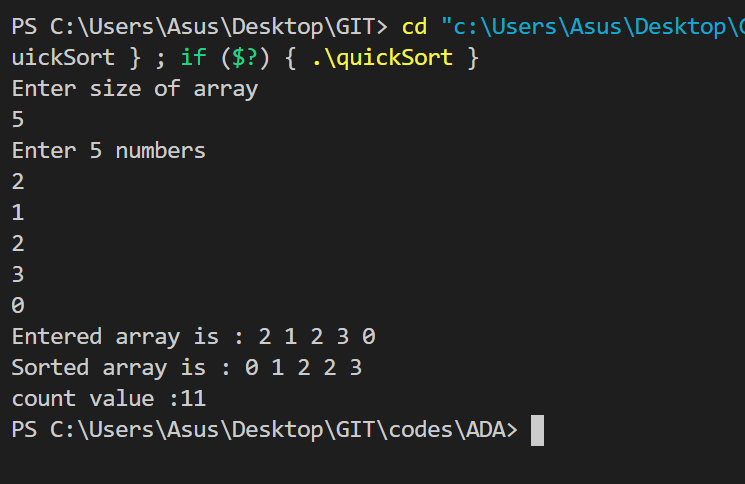
        cout<< arr[i]<<" ";

    }

     cout <<"\ncount value :" <<count;

    }

Output:



Analysis:

