CSE 2208
Algorithms Lab

Assignment No:**1**

Assignment Topic:

1. Quick Sort Algorithm
2. Merge Sort Algorithm

Date of Performance: **05.02.2020** **Date of Submission**: **12.02.2020**	**Name**: **Mubina Ashrafi** **Student ID**: **18.01.04.030** **Lab Group**: **A_2** **Department of CSE, AUST.**

1.Merge Sort

```
#include<bits/stdc++.h>
using namespace std;

void Merge(int A[],int l,int m,int u)
{
   int i,j,k;
   int n1=m-l+1;
   int n2=u-m;
   int L[n1],R[n2];
   for(i=0; i<n1; i++)
   {
      L[i]=A[l+i];
   }
   for(j=0; j<n2; j++)
   {
      R[j]=A[m+1+j];
   }
   i=0;
   j=0;
   k=l;
   while(i<n1 && j<n2)
   {
      if(L[i]<=R[j])
      {
         A[k]=L[i];
         i++;
      }
      else{
```

```
        A[k]=R[j];
        j=j+1;
      }
      k=k+1;
    }
    while(i<n1)
    {
      A[k]=L[i];
      i++;
      k++;
    }
    while(j<n2)
    {
      A[k]=R[j];
      j=j+1;
      k=k+1;
    }
}
void MergeSort(int A[],int l,int u)
{
    if(l<u)
    {
      int mid=(l+u)/2;
      MergeSort(A,l,mid);
      MergeSort(A,mid+1,u);
      Merge(A,l,mid,u);
    }
}
int main()
```

```
{
   int n,i;
   cin >> n;
   int a[n];
   for(i=0; i<n; i++)
   {
      cin >> a[i];
   }
   MergeSort(a,0,n);
   for(i=0; i<n; i++)
   {
      cout << a[i] <<" ";
   }
   return 0;
}
```

2.Quick Sort

```
#include<bits/stdc++.h>
using namespace std;
int part(int A[],int l,int h)
{
   int pivot=A[h];
   int i=l;
   int j;
   for(j=l; j<h; j++)
   {
      if(A[j]<pivot)
      {
```

```
            int temp=A[i];
            A[i]=A[j];
            A[j]=temp;
            i=i+1;
        }
    }
    int temp1=A[h];
    A[h]=A[i];
    A[i]=temp1;
    return i;
}
void quickSort(int A[],int low,int high)
{
    if(low<high)
    {
        int p=part(A,low,high);
        quickSort(A,low,p-1);
        quickSort(A,p+1,high);
    }
}
int main()
{
    int n;
    cin >> n;
    int a[n];
    int i;
    for(i=0; i<n; i++)
    {
        cin >> a[i];
```

```
    }
    quickSort(a,0,n-1);
    int j;
     for(j=0; j<n; j++)
    {
       cout << a[j] <<" ";
    }
    return 0;
}
```