Data Structures
UCS301
Assignment-6

Submitted by:
Name: Ananya Agarwal
Roll No: 102083036
Batch: 2CO14

Q1. Write a program to implement following sorting techniques:

a. Selection Sort

b. Insertion Sort

c. Bubble Sort

d. Merge Sort

e. Quick Sort

f. Counting Sort

Soln.

```
#include<iostream>
using namespace std;


void SelectionSort(int arr[],int n)
{
 int i,j,k;
 for(i=0;i<n;i++)
 {
 int min = arr[i];
 k = i;
 for(j=i+1;j<n;j++)
 {
 if(arr[j] < min)
 {
 min = arr[j];
 k = j;
 }
 }
 int t = arr[i];
 arr[i] = arr[k];
 arr[k] = t;
 }
}
void InsertionSort(int arr[],int n)
{
 int i,j;
 for(i=1;i<n;i++)
 {
 int temp = arr[i];
 for(j=i-1;j>=0 && arr[j]>temp;j--)
 arr[j+1] = arr[j];
 arr[j+1] = temp;
 }
}
void BubbleSort(int arr[],int n)
{
 int i,j;
 for(i=0;i<n-1;i++)
 {
```

```
 int flag = 0;
 for(j=0;j<n-i-1;j++)
 {
 if(arr[j] > arr[j+1])
 {
 int t = arr[j];
 arr[j] = arr[j+1];
 arr[j+1] = t;
flag = 1;
 }
 }
 if(flag == 0)
 break;
 }
}
void merge(int arr[],int l,int m,int r)
{
 int i,j,k;
 int n1 = m - l + 1;
 int n2 = r - m;
 int *arr1 = new int[n1];
 int *arr2 = new int[n2];
 for(i=0;i<n1;i++)
 arr1[i] = arr[l + i];
 for(j=0;j<n2;j++)
 arr2[j] = arr[m + j + 1];
 i = j = 0;
 k = l;
 while(i<n1 && j<n2)
 {
 if(arr1[i] <= arr2[j])
 {
 arr[k] = arr1[i];
 i++;
 }
 else
 {
 arr[k] = arr2[j];
 j++;
 }
 k++;
 }
 while(i<n1)
 {
 arr[k] = arr1[i];
 i++;
 k++;
 }
 while(j<n2)
 {
```

```
 arr[k] = arr2[j];
 j++;
 k++;
 }
 delete[] arr1;
 delete[] arr2;
}
void MergeSort(int arr[],int l,int r)
{
 if(r>l)
 {
 int m = (l + r)/2;
 MergeSort(arr,l,m);
 MergeSort(arr,m+1,r);
 merge(arr,l,m,r);
 }
}

void ShellSort(int arr[],int n)
{
 int gap,i,j;
 for(gap=n/2;gap > 0;gap/=2)
 {
 for(i=gap;i<n;i++)
 {
 int temp = arr[i];
 for(j=i;j>=gap && arr[j-gap]>temp;j-=gap)
 {
 arr[j] = arr[j - gap];
 }
 arr[j] = temp;
 }
 }
}
int Partition(int arr[],int p,int r)
{
 int x = arr[r];
 int i = p - 1;
 int j,t;
 for(j=p;j<r;j++)
 {
 if(arr[j]<=x)
 {
 i++;
 t = arr[i];
 arr[i] = arr[j];
 arr[j] = t;
 }
 }
 i++;
```

```
 t = arr[i];
 arr[i] = arr[r];
 arr[r] = t;
 return i;
}
void QuickSort(int arr[],int p,int r)
{
 if(p < r)
 {
 int q =Partition(arr,p,r);
 QuickSort(arr,p,q-1);
 QuickSort(arr,q+1,r);
 }
}
void CountingSort(int arr[],int n)
{
 int i,j,k;
 k = arr[0];
 for(i=1;i<n;i++)
 {
 if(arr[i] > k)
 k = arr[i];
 }
 int *c = new int[k+1];
 int *b = new int[n+1];
 for(i=0;i<=k;i++)
 c[i] = 0;
 for(i=0;i<n;i++)

 c[arr[i]]++;
 for(i=1;i<=k;i++)
 c[i] += c[i-1];
 for(i=n-1;i>=0;i--)
 {
 b[c[arr[i]]] = arr[i];
 c[arr[i]]--;
 }
 for(i=0;i<n;i++)
 arr[i] = b[i+1];
}
void display(int arr[],int n)
{
 int i;
 cout<<"\nArray elements are as follows : \n";
 for(i=0;i<n;i++)
 cout<<arr[i]<<" ";
 cout<<endl;
}
int main()
{
```

```
int i,n;
cout<<"Enter the required number of elements in the array: ";
cin>>n;
int *arr = new int[n];
cout<<"Enter elements in array : \n";
for(i=0;i<n;i++)
cin>>arr[i];
int arr_size = sizeof(arr) / sizeof(arr[0]);

cout<<"SELECTION SORT:";
SelectionSort(arr,n);
display(arr,n);
cout<<"\nINSERTION SORT:";
InsertionSort(arr,n);
display(arr,n);
cout<<"\n BUBBLE SORT:";
BubbleSort(arr,n);
display(arr,n);
cout<<"\n MERGE SORT:";
MergeSort(arr,0,arr_size - 1);
display(arr,n);
cout<<"\n QUICK SORT:";
QuickSort(arr,0,arr_size - 1);
display(arr,n);
cout<<"\n COUNTING SORT:";
CountingSort(arr,n);
display(arr,n);

delete[] arr;
return 0;
}
```

Output:-

```
C:\Users\User\Desktop\2nd year_3 sem\DS\labs ass\ass 6\q1.exe
Enter the required number of elements in the array: 7
Enter elements in array :
2
3
1
6
4
7
5
SELECTION SORT:
Array elements are as follows :
1 2 3 4 5 6 7

INSERTION SORT:
Array elements are as follows :
1 2 3 4 5 6 7

 BUBBLE SORT:
Array elements are as follows :
1 2 3 4 5 6 7
```

```
 MERGE SORT:
Array elements are as follows :
1 2 3 4 5 6 7

 QUICK SORT:
Array elements are as follows :
1 2 3 4 5 6 7

 COUNTING SORT:
Array elements are as follows :
1 2 3 4 5 6 7

--------------------------------
Process exited after 24.7 seconds with return value 0
Press any key to continue . . .
```

Q2. A slightly improved selection sort

Soln.

```
#include<iostream>
using namespace std;
void display(int a[],int n)
{
 int i;
 cout<<"\nArray elements are : \n";
 for(i=0;i<n;i++)
 cout<<a[i]<<" ";
 cout<<endl;
}
void SelectionSort(int a[],int n)
{
 for(int i=0,j=n-1;i<j;i++,j--)
 {
 int min = a[i], max = a[i];
 int min_i = i, max_i = i;
 for (int k = i; k <= j; k++)
 {
 if (a[k] > max)
 {
 max = a[k];
 max_i = k;
 }
 else if (a[k] < min)
 {
 min = a[k];
 min_i = k;
 }
 }
 swap(a[i], a[min_i]);
 if (a[min_i] == max)
 swap(a[j], a[min_i]);
```

```
  else
  swap(a[j], a[max_i]);
  }
 }
int main()
 {
  int i,n;
  cout<<"Enter the number of elements : ";
  cin>>n;
  int *a = new int[n];
  cout<<"Enter array elements : \n";
  for(i=0;i<n;i++)
  cin>>a[i];
  SelectionSort(a,n);
  display(a,n);
  delete[] a;
  return 0;
}
```

Output:-

```
C:\Users\User\Desktop\2nd year_3 sem\DS\labs ass\ass 6\q2.exe
Enter the number of elements : 7
Enter array elements :
2
3
1
4
7
6
5

Array elements are :
1 2 3 4 5 6 7

--------------------------------
Process exited after 10.14 seconds with return value 0
Press any key to continue . . .
```