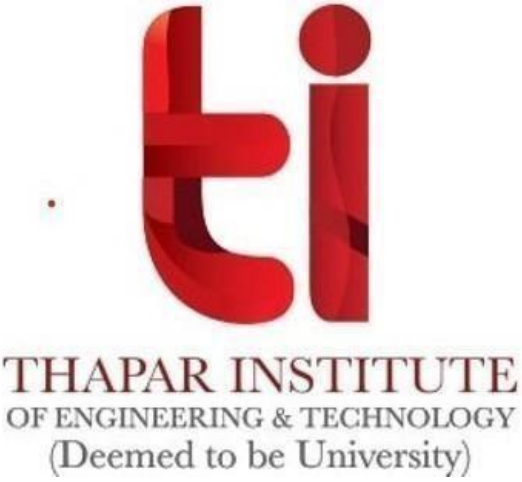
A Practical Activity Report For
Data Structures and Algorithms (UCS406)

Submitted By: **Vivek Arora**

101715178
(ENC 8)

Submitted To:

Dr. Sanjay Sharma



ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR INSTITUTE OF ENGINEERING &TECHNOLOGY, (DEEMED TO BEUNIVERSITY), PATIALA, PUNJAB

ASSIGNMENT-9

Write a program to perform sorting of a given using the following algorithms:
QUESTION 1)Insertion Sort

```
#include<iostream>
using namespace std;
void display(int *array, int size) {
  for(int i = 0; i<size; i++)
        cout << array[i] << " ";
  cout << endl;
}
void insertionSort(int *array, int size) {
  int key, j;
  for(int i = 1; i<size; i++) {
        key = array[i];//take value
        j = i;
        while(j > 0 && array[j-1]>key) {
        array[j] = array[j-1];
        j--;
        }
        array[j] = key;   //insert in right place
  }
}
int main() {
  int n;
  cout << "Enter the number of elements: ";
  cin >> n;
  int arr[n];    //create an array with given number of elements
  cout << "Enter elements:" << endl;
  for(int i = 0; i<n; i++) {
        cin >> arr[i];
  }
  cout << "Array before Sorting: ";
  display(arr, n);
  insertionSort(arr, n);
  cout << "Array after Sorting: ";
  display(arr, n);
}
```

QUESTION 2)Selection Sort

```
#include<iostream>
using namespace std;
void selection_sort(int A[],int n){
   for(int i=0;i<n-1;i++){
         int imin=i;
         for(int j=i;j<n;j++){
               if(A[j]<A[imin])
                     imin=j;
         }
         int temp;
         temp=A[i];
         A[i]=A[imin];
         A[imin]=temp;
   }}



int main(){
   int arr[]={2,7,4,1,5,3},n= *(&arr + 1) - arr;
   selection_sort(arr,n);
   cout<<endl;
   for(int i=0;i<n;i++){
         cout<<arr[i]<<" ";
   }
   cout<<endl;
}
```

QUESTION 3) Quick Sort

```
#include<iostream>
using namespace std;
void swap(int* a, int* b){
   cout<<"swap";
   int temp=*a;
   *a=*b;
   *b=temp;
}
int partition(int* A,int start,int end){
   cout<<"partition";
   int pivot=A[end];
   int pIndex=start;
```

```
    for(int i=start;i<end;i++){
        if(A[i]<=pivot){
            swap(A[i],A[pIndex]);
            pIndex++;
        }
    }
    swap(A[pIndex],A[end]);
    return pIndex;
}
void quick_sort(int* A,int start,int end){
    cout<<"quick";
    if(start<end){
        int pIndex=partition(A,start,end);
        quick_sort(A,start,pIndex-1);
        quick_sort(A,pIndex+1,end);
        }
}
int main(){
    int arr[]={7,2,1,6,8,5,3,4};
    int n=*(&arr+1)-arr;
    cout<<n;
    quick_sort(arr,0,n-1);
    cout<<endl;
    for(int i=0;i<n;i++){
        cout<<arr[i]<<" ";
    }
    cout<<endl;
    return 0;
}
```

QUESTION 4)MergeSort

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

void merge(int a1[],int n1,int a2[],int n2,int a[],int n){
    int i=0,j=0,k=0;
    while(i<n1&&j<n2){
        if(a1[i]<a2[j])
            a[k++]=a1[i++];
        else
            a[k++]=a2[j++];
```

```
    }
    while(i<n1){
        a[k++]=a1[i++];
    }
    while(j<n2){
        a[k++]=a2[j++];
    }
}
void merge_sort(int arr[],int n){
    if(n<2)
        return;
    int mid=n/2;
    int left[mid],right[n-mid];
    for(int i=0;i<mid;i++){
        left[i]=arr[i];
    }
    for(int i=mid;i<n;i++){
        right[i-mid]=arr[i];
    }
    merge_sort(left,mid);
    merge_sort(right,n-mid);
    merge(left,mid,right,n-mid,arr,n);
}
int main(){
    int A[]={902,234,345,67,23,234,64,64444,324,554};
    int n=*(&A+1)-A;
    merge_sort(A,n);
    for(int i=0;i<n;i++){
        cout<<A[i]<<" ";
    }
}
```

QUESTION 5)Shell Sort

```
#include<iostream>
using namespace std;
void swapping(int &a, int &b) {
   int temp;
   temp = a;
   a = b;
   b = temp;
}
```

```
void display(int *array, int size) {
  for(int i = 0; i<size; i++)
        cout << array[i] << " ";
  cout << endl;
}
void shellSort(int *arr, int n) {
  int gap, j, k;
  for(gap = n/2; gap > 0; gap = gap / 2) {          //initially gap = n/2,
        decreasing by gap /2
        for(j = gap; j<n; j++) {
        for(k = j-gap; k>=0; k -= gap) {
        if(arr[k+gap] >= arr[k])
        break;
        else
        swapping(arr[k+gap], arr[k]);
        }
        }
  }
}
int main() {
  int n;
  cout << "Enter the number of elements: ";
  cin >> n;
  int arr[n];    //create an array with given number of elements
  cout << "Enter elements:" << endl;
  for(int i = 0; i<n; i++) {
        cin >> arr[i];
  }
  cout << "Array before Sorting: ";
  display(arr, n);
  shellSort(arr, n);
  cout << "Array after Sorting: ";
  display(arr, n);
}
```