**Data Structures & Algorithm LAB**

**LAB # 02**



**Spring 2020**

**CSE-210L Data Structures & Algorithms**

Submitted by

**Shah Raza 18PWCSE1658**

**Muhammad Hassan 18PWCSE1669**

**Razi Ullah Durrani 18PWCSE1673**

**Muhammad Adil Khan 18PWCSE1683**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr. Khurram S. Khattak**

Monday, March 9, 2020

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Task 1:**

Implement Selection Sort and analyze its worst, best and average case complexity.

**Code:**

#include <iostream>

using namespace std;

void SelectionSort(int Array[], int SIZE)

{

for(int i=0;i<SIZE;i++)

{

for(int j=i+1;j<SIZE;j++)

{

if(Array[i]>Array[j])

{

int temp= Array[i];

Array[i]=Array[j];

Array[j]=temp;

}

}

}

}

int main()

{

int SIZE;

cout<<"Enter the Size of the Array: ";

cin>>SIZE;

int Array[SIZE];

cout<<"Enter the Elements of the Array: ";

for(int i=0;i<SIZE;i++)

cin>>Array[i];

SelectionSort(Array,SIZE);

cout<<"Sorted Array: ";

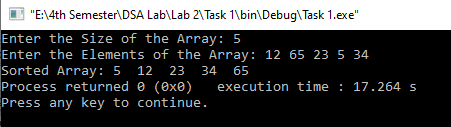
for(int i=0;i<SIZE;i++)

cout<<Array[i]<<" ";

return 0;

}

**Output:**



**Complexity:**

* **Best case/Worst case:**

For selection sort algorithm both best case and worst-case complexity is O [N2] since in this algorithm there are two for loops and both for loops goes from 0 to n even if the array is already sorted.

**Task 2:**

Implement Bubble Sort and analyze its worst, best and average case complexity.

**Code:**

#include <iostream>

using namespace std;

void BubbleSort(int Array[],int SIZE)

{

for(int i=0;i<SIZE;i++)

{

for(int j=0;j<SIZE-1;j++)

{

if(Array[j]>Array[j+1])

{

int temp=Array[j];

Array[j]=Array[j+1];

Array[j+1]=temp;

}

}

}

}

int main()

{

int SIZE;

cout<<"Enter the Size of the Array: ";

cin>>SIZE;

int Array[SIZE];

cout<<"Enter the Elements of the Array: ";

for(int i=0;i<SIZE;i++)

cin>>Array[i];

BubbleSort(Array,SIZE);

cout<<"Sorted Array: ";

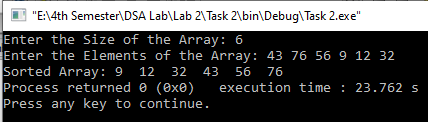
for(int i=0;i<SIZE;i++)

cout<<Array[i]<<" ";

return 0;

}

**Output:**



**Complexity:**

* **Best case:**

For bubble sort algorithm best case complexity is O [N] since in this algorithm there are two for loops and in the best case where the array is already sorted the loop treverses only one time from 0 to n.

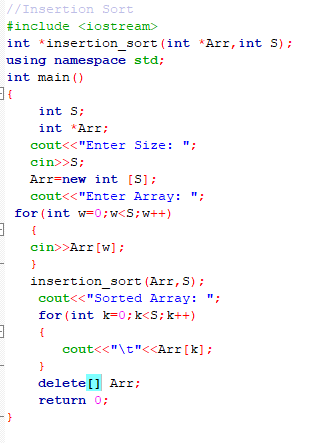
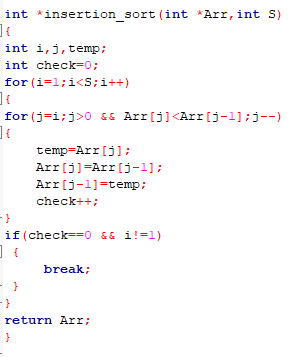
* **Worst case:**

For bubble sort algorithm worst case complexity is O [N2] since in this algorithm there are two for loops and in the worst case where the array is mixed up, the loops treverses n time from 0 to n.

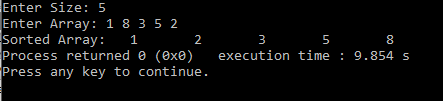
**Task 3:**

Implement Insertion Sort and analyze its worst, best and average case complexity.

**Code:**

** **

**Output:**



**Complexity:**

* **Best case:**

For insertion sort algorithm best case complexity is O [N] since in this algorithm there are two for loops and in the best case where the array is already sorted the loop treverses only one time from 0 to n.

* **Worst case:**

For insertion sort algorithm worst case complexity is O [N2] since in this algorithm there are two for loops and in the worst case where the array is mixed up, the loops treverses n time from 0 to n.