**TITLE OF LAB: (SORTING ALGORITHMS PART B)**

**LAB REPORT NO.03**



**Spring 2022**

**CSE-210L Data Structures and Algorithm Lab**

Submitted by

Name:: **Safi Ullah Khan**

Registration No. **20PWCSE1943**

Class Section: **B**

“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 Shehzad Khattak**

(Friday, July 29th, 2022)

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**OBJECTIVES OF THE LAB**

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

In this lab, we will learn about some more sorting techniques and algorithms.

* Merge Sort
* Quick Sort

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

## **Task 01**

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

**Merge Sort:**

Merge Sort

#include <iostream>

int s;

using namespace std;

void divide(int\* a,int g,int h){

if(g<h){

int mid=(g+h)/2;

divide(a,g,mid);

divide(a,mid+1,h);

combine(a,l,m,hi);

}

}

void combine(int\* a,l,m,hi){

}

void mergeSort(int a[]){

while(sizeof(a)>4){

divide(a,0,s-1);

}

}

int main()

{

cout<<"Enter size of array : \n";

cin>>s;

int \*arr=new int(s);

cout<<"Enter array elements : \n";

for (int i=0;i<s;i++){

cin>>arr[i];

}

mergeSort(arr);

cout<<"\n Sorted array : \n ";

for (int i=0;i<s;i++){

cout<<arr[i]<<endl;

}

delete arr;

return 0;

}

[Best: O(n log n), Worst: O(n log n)]

## **Task 02**

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

**Screenshot of Input:**







**Screenshot of Output:**



**Best Case:**

O(n log n)

**• Worst Case:**

O(n2