**DAA-Lab**

**LAB - 4**

LQ1. Aim of the experiment : To implement Binary Search and Modified Binary Search. In Modified Binary Search, middle formula is (last position -2). And also compare the time complexity.

CASE-1

**INPUT:**

Array: 2 5 7 7 9 11 12 12 17 19

Search keys: 3, 2, 15, 12

**OUTPUT:**

Linear search: 3 no, 2 yes, 15 no, 12 yes

CASE-2

**INPUT**

Array: 1 3 5 7 9 11 13 15

Search keys: 5, 8, 13, 20

**OUTPUT:**

Linear search: 5 yes, 8 no, 13 yes, 20 no

CASE-3

**INPUT:**

Array: 4 8 12 16 20 24 28 32

Search keys: 10, 24, 16, 5

**OUTPUT:**

Linear search: 10 no, 24 yes, 16 yes, 5 no

LQ2. Aim of the experiment : To implement Merge Sort and 3-way Merge Sort. And also compare the time complexity. Input to be used : 79 2 11 19 17 12 5 7 12 Expected Output : Merge Sort : [ 257 79 11 12 12 17 19 ] 37 comparisons

CASE-1

**INPUT:**

Array: 8 4 16 2 10 6 12 14

**OUTPUT:**

Merge Sort: [6, 9, 12, 18, 20, 25, 30, 36] 20 comparisons

CASE-2

**INPUT**

Array: 5 8 15 2 10 7 13 12

**OUTPUT:**

Merge Sort: [2, 5, 7, 8, 10, 12, 13, 15] 19 comparisons

CASE-3

**INPUT:**

Array: 30 22 15 10 27 18 35 12

**OUTPUT:**

Merge Sort: [10, 12, 15, 18, 22, 27, 30, 35] 20 comparisons

LQ3. Aim of the experiment : To implement Quick Sort. Element are store into an array in three different ways random or increasing or decreasing order. And also compare the time complexity. Input : 792 11 19 17 12 5 7 12 Expected Output : Quicksort : [ 2 5 7 79 11 12 12 17 19 ] 12 comparisons

CASE-1

**INPUT:**

Array: 5 8 15 2 10 7 13 12

**OUTPUT:**

Quicksort: [2, 5, 7, 8, 10, 12, 13, 15] 18 comparisons

CASE-2

**INPUT**

Array: 30 22 15 10 27 18 35 12

**OUTPUT:**

Quicksort: [10, 12, 15, 18, 22, 27, 30, 35] 20 comparisons

CASE-3

**INPUT:**

Array: 8 4 16 2 10 6 12 14

**OUTPUT:**

Quicksort: [2, 4, 6, 8, 10, 12, 14, 16] 18 comparisons