Laboratory 7

Sorting & Searching

I. OBJECTIVES:

In this laboratory, you will

- understand and apply the concepts of sorting and searching algorithms
- implement the different sorting techniques such as selection sort, insertion sort, bubble sort, quick sort and binary sort
- implement the different searching techniques; linear search and binary search

II. DISCUSSION:

Sorting

It is a way of arranging the set of items in order to produce a report to simplify manual retrieval of information and to make machine access to data more efficient.

Types of Exchange Sorting:

(1) Bubble Sort

The basic idea is to pass through file sequentially several time. Each pass consists of comparing each element in the file with its successor x[i] with x[i+1] and interchanging the two elements if they are not in proper order.

Consider the given file:

Original file	**25**	**57**	**48**	**37**	**12**	**92**	**86**	**33**

Summary of Iterations:

Iteration 0	25	57	48	37	12	92	86	33
Iteration 1	25	48	37	12	57	86	33	92
Iteration 2	25	37	12	48	57	33	86	92
Iteration 3	25	12	37	48	33	57	86	92
Iteration 4	12	25	37	33	48	57	86	92
Iteration 5	12	25	33	37	48	57	86	92
Iteration 6	12	25	33	37	48	57	86	92
Iteration 7	12	25	33	37	48	57	86	92

(2) Quick Sort

Also known as **partition exchange sort**. Let ***x*** be an array, and ***n*** the number of elements in the array to be sorted. Choose an element ***a*** from a specific position within the array.

Suppose that the elements of ***x*** are partitioned so that ***a*** is placed into position ***j*** and the following conditions hold:

1. Each of the elements in positions ***0*** through ***j-1*** is less than or equal to ***a***.
2. Each of elements in positions ***j+1*** through ***n-1*** is greater than or equal to ***a***.

For example, if an initial array is given as:

25	57	48	37	12	92	86	33

The first element (25) is placed in its proper position:

(12) **25** (57 48 37 92 86 33)

The next element (12) is placed in its proper position and compared with (57 48 37 92 86 33) :

12 and (57 48 37 92 86 33)

Nothing need be done to sort the elements, resulting array

12 25 (57 48 37 92 86 33)

And further repetitions yield

12 25 (48 37 33) **57** (92 86)
12 25 (37 33) **48 57** (92 86)
12 25 (33) **37 48 57** (92 86)
12 25 33 37 48 57 (92 86)
12 25 33 37 48 57 (86) **92**
12 25 33 37 48 57 86 92

Types of Linear Sorting:

(3) Selection Sort

Find the largest element in the array and exchange it with the element in the last position, then find the second largest element in array and exchange it with the element in the second last position, and continue in this way until the entire array is sorted.

Consider the given example:

Initial	**1**	**5**	**0**	**2**	**6**	**3**	**4**

Summary of Iterations:

Iteration 1	1	5	0	2	**4**	3	**6**
Iteration 2	1	**3**	0	2	4	**5**	6
Iteration 3	1	**2**	0	**3**	4	5	6
Iteration 4	1	**0**	**2**	3	4	5	6
Iteration 5	**0**	**1**	2	3	4	5	6

(4) Insertion Sort

The items of the array are considered one at a time, and each new item is inserted into the appropriate position relative to the previously sorted items.

Consider the given example:

Initial	**1**	**5**	**0**	**2**

Summary of Iterations:

Iteration 1	1	5	0	2	(No Interchange)
Iteration 2	1	0	5	2	
Iteration 3	0	1	5	2	
Iteration 4	0	1	2	5	
Iteration 5	0	1	2	5	(No Interchange)

Searching

It is a process of finding a designated target element within a group of items, or determining that the target does not exist within the group. The group of items to be searched is sometimes called the *search pool*.

Linear Search

- If the search pool is organized into a list of some kind, the search is to start at the beginning of the list and compare each value in turn to the target element.
- This approach is called *linear search* because it begins at one end and scans the search pool in a *lin*ear manner.

Binary Search

- A *binary search algorithm* eliminates the large parts of the search pool with each comparison because the search pool is in order (sorted).
- A binary search begins in the middle of the sorted list. If the target is not found in the middle element the search continues. It will either search to the left or right reference to the middle element depending on whether the target is greater than or less than the middle element.

III. Test and Debug

1. Encode and run the given program.

Sorting Program:

```
#include<conio.h>
#include<stdio.h>

int num[3];
int i,k,j;


int main(){

   printf("Input 4 integers: ");
   for (i=0; i<4; i++)
        scanf("%d", &num[i]);

   printf("Summary of Iteration:\n\n");
   for (i=0; i<3; i++) {
        if (num[i] > num[i+1]){
             k = num[i+1];
             num[i+1]=num[i];
             num[i]=k;
        }//end if
   }//end for

   for (j=1;j<4;j++) {
        printf("Iteration[%d] = ",j);
        for (i=0;i<4;i++)
             printf("%d ",num[i]);
        printf("\n");
        }//end for
   return 0;
}
```

1.1 Input the following values and illustrate the output.
Input 4 integers: 34   12   3   15

1.2 Are the values sorted?

2. Modify the given program so that it can sort the values in ascending order using Bubble sort.

Input 4 integers:	34 12 3 15
Summary of Iterations	
Iteration [1]	12 3 15 34
Iteration [2]	3 12 15 34
Iteration [3]	3 12 15 34

IV. Supplementary Exercise

Modify the given program in test & debug so that it can search for a particular value using Binary Search.

SORTED DATA	3 12 15 34

Search for Data	**Found/Not Found**
2	Not found
3	Found

V. Machine Problem

1. Write a simulation program for the following sorting techniques: Quicksort, Selection sort and Insertion sort

Unsorted Data	**5**	**8**	**3**	**2**
Quicksort	2	3	5	8
Selection Sort	2	3	5	8
Insertion Sort	2	3	5	8

The display of values must coincide with the iterations of each sorting technique. Use delay to simulate the iterations.

Note: *The instructor may change the machine problem.*

Sorting & Searching

DATA & RESULTS SHEET

(Tentative Laboratory Report)

Name: ____

Schedule:______________________________ Section:__________________

Date: _________________________________ Grade: __________________

Test & Debug:

Step No.	Answers/Results
1	**Illustrate the output:** *Are the values sorted?*
2	**Input 4 integers:** 34 12 3 15 Summary of Iterations Iteration [1] 12 3 15 34 Iteration [2] 3 12 15 34 Iteration [3] 3 12 15 34

Input 4 integers:	34	12	3	15
Summary of Iterations				
Iteration [1]	12	3	15	34
Iteration [2]	3	12	15	34
Iteration [3]	3	12	15	34

Supplementary Exercise:

SORTED DATA	3 12 15 34

Search for Data	Found/Not Found
2	Not found
3	Found

Topic	Completed?		Remarks
	YES	NO	
Searching			

Machine Problem:

Topic	Completed?		Remarks
	YES	NO	

Note: Check the column YES if completed otherwise check the column NO.(for instructors only)

INSTRUCTOR’S SIGNATURE: ______________________________