CET 214 – Data Structures & Algorithms

Experiment # 7

Experiment Title

Searching Algorithms

Assessment of CLO(s): IV

Performed on 01-11-2024

Student Name			
Roll No.		**Group**	
Semester		**Session**	

Total (Max)	**Criteria 1 (2.5)**	**Criteria 2 (2.5)**	**Criteria 3 (2.5)**	**Criteria 4 (2.5)**	**Total (10)**
Marks Obtained					
Remarks (if any)					

Experiment evaluated by

Instructor's Name	**Engr. Muhammad Asad Husain**		
Date		**Signature**	

Department of Engineering Technology
(UIT University)

Course Code: CET214 Course Title: Data Structures & Algorithms Course Credits: 2+1 Session: Fall 2024

Rubric for assessment criteria to perform experiment number 7.

Level / Criteria	UNSATISFACTORY 1	COMPETENT 2	PROFICIENT 3	DISTINGUISHED 4
Capability of writing algorithm/ Procedure	None of the steps are implemented of an algorithm.	Few steps are implemented correctly of an algorithm.	Most of the steps are implemented correctly of an algorithm.	All the steps are implemented correctly of an algorithm.
Capability of writing Program	Programs not completed.	Completeness of code, consistent variable naming and unformatted.	Completeness of code, inconsistent variable naming and well formatted.	Completeness of code, consistent variable naming and well formatted.
Completion of target in Lab	25% target has been completed	50% target has been completed	75% target has been completed	100% target has been completed
Output	None of the outputs are correct.	Few outputs have been found correctly.	Some of the outputs are correct and well formatted.	Most of the outputs are correct and well formatted.

Practical Objective(s):

1. Learn how to search an element in an array using Linear Search
2. Learn how to search an element in an array using Binary Search

Theory

Linear Search

Linear search, also known as sequential search works by checking every element of a list one at a time in sequence until a match is found. Linear search runs in O(N). If the data are distributed randomly, on average (N+1)/2 comparison will be needed. The best case is that the value is equal to the first element tested, in which case only 1 comparison is needed. The worst case is that the value is not in the list (or is the last item in the list), in which case N comparisons are needed.

Algorithm 1: Linear Search

Here **DATA** is a linear array with N elements and **ITEM** is a given item to be searched. This algorithm finds the location **LOC** of **ITEM** in **DATA**, or sets **LOC: = -1** if the search is unsuccessful.

Step 1. Read ITEM
Step 2. Set LOC: = 0
Step 3. Repeat step 4 while DATA[LOC]! = ITEM and LOC<N:
Step 4. LOC: =LOC+1
[End of loop]

Step 5. If LOC = N:
Write "Item is not in array"
Step 6. Else:
Write LOC, "is the location of", ITEM
Step 7. Exit

Binary Search

A binary search algorithm (or binary chop) is a technique for finding a particular value in a sorted list. It makes progressively better guesses, and closes in on the sought value by selecting the median element in a list, comparing its value to the target value, and determining if the selected value is greater than, less than, or equal to the target value.

Algorithm 2: Binary Search

Here **DATA** is a sorted array with lower bound **LB** and upper bound **UB**. **LB** is the index of first element and **UB** is the index of last element. **ITEM** is to be searched. This algorithm fins the location **LOC** of **ITEM** in **DATA** or sets **LOC=NULL** if **ITEM** is not found.

Step 1. Set BEG: = LB AND END: = UB
Step 2. Set MID = (BEG + END) / 2
Step 3. Repeat 4 and 5 while (BEG<=END AND ITEM! = DATA[MID])

Step 4. **If ITEM > DATA[MID] then**
BEG: = MID + 1
Else
END: = MID – 1
[End of If structure]
Step 6. **Re-calculate MID = (BEG+END)/2**
Step 7. **If DATA[MID] != ITEM, then**
Display "Location Not Found
Else "ITEM Found at Location", MID
Step 8. **Exit**

Do It Yourself:

1. Implement algorithm 1.
2. Implement algorithm 2.

(Note: Show both the scenarios in each of the exercise output i.e. search for an element that exists in the array and also the one that does not exist)