Experiment no 10

Linear and Binary Search Algorithm

OBJECTIVES:

- To learn about the concept of linear and binary search.
- To learn different ways for implementing a linear and binary search.

1. LINEAR SEARCH:

Assume that item is in an array in random order and we have to find an item. Then the only way to search for a target item is, to begin with, the first position and compare it to the target. If the item is at the same, we will return the position of the current item. Otherwise, we will move to the next position. If we arrive at the last position of an array and still can not find the target, we return -1. This is called the Linear search or Sequential search.

Below is the code syntax for the linear search.

```
# Linear Search in Python


def linearSearch(array, n, x):

    for i in range(0, n):
        if (array[i] == x):
            return i
    return -1
```

Figure 1: Linear Search

2. BINARY SEARCH:

In a binary search, however, cut down your search to half as soon as you find the middle of a sorted list. The middle element is looked at to check if it is greater than or less than the value to be searched. Accordingly, a search is done to either half of the given list

Below is the code syntax for the binary search.

```
def binarySearch(array, x, low, high):

    # Repeat until the pointers low and high meet each other
    while low <= high:

        mid = low + (high - low)//2

        if array[mid] == x:
            return mid

        elif array[mid] < x:
            low = mid + 1

        else:
            high = mid - 1

    return -1
```

Figure 2: Binary Search

3. DIFFERENCE BETWEEN BINARY AND LINEAR SEARCH:

Linear Search	Binary Search
In linear search input data need not to be in sorted.	In binary search input data need to be in sorted order.
It is also called sequential search.	It is also called half-interval search.
The time complexity of linear search **O(n)**.	The time complexity of binary search **O(log n)**.
Multidimensional array can be used.	Only single dimensional array is used.
Linear search performs equality comparisons	Binary search performs ordering comparisons
It is less complex.	It is more complex.
It is very slow process.	It is very fast process.

Figure 3: Difference between binary and linear search

Let us look at an example to compare the two:

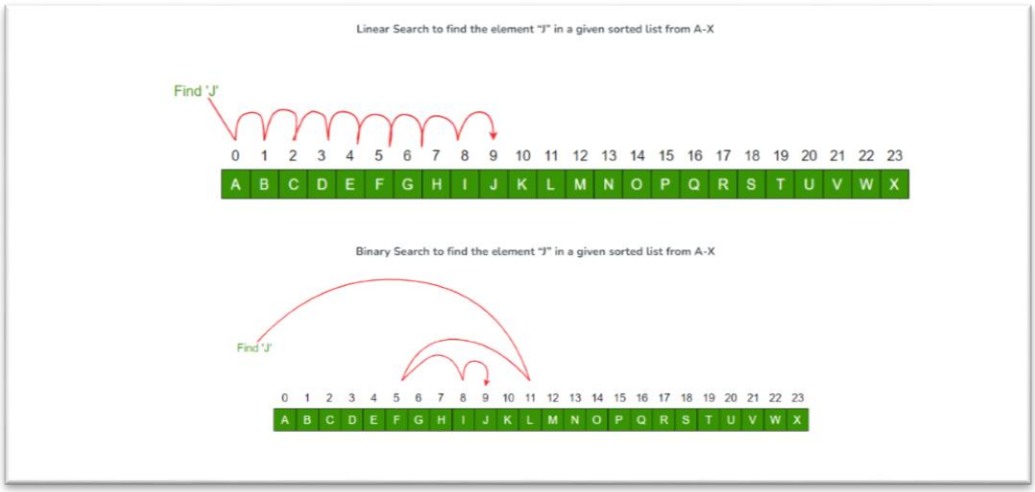

Figure 4: Binary and Linear Search Graphical

4. BINARY AND LINEAR SEACHING EXAMPLES:

```
def linearSearch(array, n, x):

    for i in range(0, n):
        if (array[i] == x):
            return i
    return -1


array = [24, 41, 31, 11, 9]
x = 11
n = len(array)
result = linearSearch(array, n, x)
if(result == -1):
    print("Element not found")
else:
    print("Element is Present at Index: ", result)
```

Figure 5: Linear Search

Time Complexity: O(n), where n is the size of the input array. The worst-case scenario is when the target element is not present in the array, and the function has to go through the entire array to figure that out.
Auxiliary Space: O(1), the function uses only a constant amount of extra space to store variables. The amount of extra space used does not depend on the size of the input array.

```
def binarySearch(array, x, low, high):

    while low <= high:

        mid = low + (high - low)//2

        if array[mid] == x:
            return mid

        elif array[mid] < x:
            low = mid + 1

        else:
            high = mid - 1

    return -1


array = [2, 4, 5, 7, 14, 17, 19, 22]
x = 22

result = binarySearch(array, x, 0, len(array)-1)

if result != -1:
    print(str(result))
else:
    print("Not found")
```

Figure 6: Binary Search

Time Complexity: O(log n) – Binary search algorithm divides the input array in half at every step, reducing the search space by half, and hence has a time complexity of logarithmic order.
Auxiliary Space: O(1) – Binary search algorithm requires only constant space for storing the low, high, and mid indices, and does not require any additional data structures, so its auxiliary space complexity is O(1).

5. ACTIVITIES:

A1) Write a code for Linear search.

A2) Write a code for Binary search.

A3) Write a code for through which you can compare linear and binary search on a same scenario.

IMPORTANT: All the activities` code should be attached to the manual before summary section.

Lab Exercise and Summary

Summary should cover Introduction, Procedure, Data Analysis and Evaluation.

Student's Signature:_____ **Date:**_____

LABORATORY SKILLS ASSESSMENT (Psychomotor)

Total Marks:100

Criteria (Max Marks)	Level 1 0% ≤ S < 50%	Level 2 50% ≤ S< 70%	Level 3 70% ≤ S< 90%	Level 4 90%≤ S ≤100%	Score (S)
Procedural Awareness (30)	Selects inappropriate skills and/or strategies Required by the task.	Selects and applies appropriate skills and/or strategies required by the task with major errors.	Selects and applies the appropriate strategies and/or skills specific to the task without significant errors.	Selects and applies appropriate strategies and/or skills specific to the task without any error.	
Practical Implementation (30)	Makes major critical errors in applying procedural knowledge related to python Linear and Binary Search.	Makes numerous critical errors in applying procedural knowledge related to python Linear and Binary Search.	Makes some non-critical errors in applying procedural knowledge related to python Linear and Binary Search.	Applies the procedural knowledge in optimized ways related to python Lists, Linear and Binary Search.	
Use of Tool/Equipment (30)	Uses tools, equipment and materials with limited competence.	Uses tools, equipment and materials with some competence.	Uses tools, equipment and materials with considerable competence.	Uses tools, equipment and materials with a high degree of competence.	
Safety (10)	Requires constant reminders to follow safety procedures.	Requires some reminders to follow safety procedures.	Follows safety procedures with only minimal reminders.	Routinely follows safety procedures.	
				Marks Obtained	

Instructor Name:_____ **Sign:** _____

LABORATORY SKILLS ASSESSMENT (Affective)

Total Marks: 40

Criteria (Max. Marks)	Level 1 0% ≤ S < 50%	Level 2 50% ≤ S < 70%	Level 3 70% ≤ S < 90%	Level 4 90% ≤ S ≤ 100%	Score
Introduction (5)	Very little background information provided or information is incorrect	Introduction is brief with some minor mistakes	Introduction is nearly complete, missing some minor points	Introduction complete and well-written; provides all necessary background principles for the experiment	
Procedure (5)	Many stages of the procedure are not entered on the lab report.	Many stages of the procedure are entered on the lab report.	The procedure could be more efficiently designed but most stages of the procedure are entered on the lab report.	The procedure is well designed and all stages of the procedure are entered on the lab report.	
Data Record (10)	Data is brief and missing significant pieces of information.	Data provides some significant information and has few critical mistakes.	Data is almost complete but has some minor mistakes.	Data is complete and relevant. Tables with units are provided. Graphs are labeled. All questions are answered correctly.	
Data Analysis (10)	Data is presented in very unclear manner.	Data is presented in ways that are not clear enough.	Data is presented in ways that can be understood and interpreted.	Data is presented in ways that best facilitate understanding and interpretation.	
Report Quality (10)	Report contains many errors.	Report is somewhat organized with some spelling or grammatical errors.	Report is well organized and cohesive but contains some grammatical errors.	Report is well organized and cohesive and contains no grammatical errors. Presentation seems polished.	
				Marks Obtained	

LABORATORY SKILLS ASSESSMENT (Cognitive)

Total Marks: 10

(If any) **Marks Obtained**	

Instructor's Signature:_____ **Date:**_____