IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13506

Comparative Study of Searching Algorithm: Linear Search and Binary search

Mrs. Asmita Kurhade[1], Ms. Neeta Namdeo Takawale[2]

Assistant Professor, Dept. of Computer Science, Dr. D. Y. Patil Arts, Commerce and Science College, Pimpri, Pune, India[1]

Assistant Professor, Dept. of Computer Science, Dr. D. Y. Patil Arts, Commerce and Science College, Pimpri, Pune, India[2]

Abstract:- Searching is an important operation in data structure helps to find specific data within a collection of data.Searching algorithms helps us in the retrieval of information which is stored within some data structures such as arrays, linked list and trees. Searching is a process of finding an element in the given list.Number of searching algorithms are available, here we have to find which algorithm is best suited according to the situation.This paper gives detailed study of how searching algorithm works and give their performance analysis with respect to time complexity.

Keywords: Linear search, Binary search, Data Structure

I. INTRODUCTION

The process of finding the desired information from the set of items stored in form of data structures is referred as Searching in Data Structure.Search can be done by using searching algorithm to find element in data structure.

The basic searching technique are -

1. LinearSearch
2. Binarysearch

II. SEARCHING

Searching is the process of finding a particular element in a given list or array. It checks whether the element is found in the list or not. It is the logical process of finding a particular item in a list of items.Searching Algorithms are designed to check for an element or retrieve an element from any data structure where it is stored.

LINEAR SEARCH

Linear Search, also known as Sequential Search, is one of the simplest and most straightforward searching algorithms. In linear Search, the algorithm checks each element of the list until a match is found or the end of the list is reached. It works by sequentially examining each element in an array or list until a match is found or the entire list has been traversed.It is suitable for unsorted arrays or lists.If the list contains n number of elements,then the time complexity of linear search is O(n).

ALGORITHM OF LINEAR SEARCH

1. Start from the first element of the list.
2. Compare the target element with the current element.
3. If the current element is equal to the target element, return its index.
4. If the current element and target element both are not equal, move to the next element in the list.
5. Repeat steps 2-4 until either the target element is found or the end of the list is reached.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13506

6. If the target element is not found after examining the entire list, return a value indicating that the target element is not present in the list.

Example:

1)Target element=15

Unsorted array

15	24	2	36	46	8	15
15	24	2	36	46	8	15
15	24	2	36	46	8	15
15	24	2	36	46	8	15
15	24	2	36	46	8	15
15	24	2	36	46	8	15

Conclusion - Element is not found in the list.

2)Target element=20

Sorted array

10	15	25	35	40	20
10	15	25	35	40	20
10	15	25	35	40	20
10	15	25	35	40	20
10	15	25	35	40	20

Conclusion - Element is not found in the list.

BINARY SEARCH

In Binary Search, algorithm repeatedly divides the search space in half.It works within a sorted array or list.Start compare the target element with the mid of the list.Then we have to check first half or second half of the list depending on whether the target element is in less or greater than the mid element.If the middle element is equal to the key element then it return the position of the element in the list if the key element value is less than the middle element then search continues in left half. If the key element value is more than the middle element then the search continues in right half.

Less time is taken by the binary search to search an element from the sorted list or array.So we conclude that, binary search is most efficient method of searching than linear search.Binary search is based on Divide and Conquer strategy, which divides the list into two parts and at a time search the only one part f the list, Hence it reduces the search time.

The time complexity of the algorithm is O(log n) that means it take less amount of time as

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024

DOI: 10.17148/IJARCCE.2024.13506

Compared to linear search.Binary search is faster than linear search .

ALGORITHM OF BINARY SEARCH

1 Compare the key element with the middle element

2 If the key element matches the middle Element Then,return index

3 Else if key<mid then search in left half using recursion

4 Else if key>mid then search in right half using recursion

Example: Suppose a list contains 7 elements

Target element=59

10	20	25	34	46	59	70
0	1	2	3	4	5	6

Solution —

Step 1)LB = 0; UB = 6 ; mid = (0 + 6)/ 2 = 3

A[mid] = A[3] = 34

Step 2)

Since (A[3] < data) - i.e., 34 < 59

LB = 3; UB = 6; mid = (3 + 6)/ 2 = 4

A[mid] = A [4] = 46 Since 59>A[mid]

Target element is in the second half.

Step 3)

Since (A[4] < data) - i.e., 46 < 59

LB = 4; UB = 6; mid = (4 + 6)/ 2 = 5

A[mid] = A [5] = 59

Target element=A[mid]

Element is found at location 5.

III. Discussions

From the above studies we can say that the easiest method to search an element in the list is linear search but the time complexity of linear search isO(n) more than that of binary search(log n)

a)Performance:

Linear search:

1. Worst-case time complexity: O(n), where n is the number of elements in the array.
2. Average-case time complexity: O(n/2).
3. Best-case time complexity: O(1) if the target is found at the beginning of the array.
4. Space complexity: O(1).
5.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 5, May 2024
DOI: 10.17148/IJARCCE.2024.13506

Binary search:

1. Worst-case time complexity: O(log n), where n is the number of elements in the array. This is because binary search divides the array in half with each iteration.
2. Average-case time complexity: O(log n).
3. Best-case time complexity: O(1) if the target is found at the middle of the array.
4. Space complexity: O(1).

b)Complexity:

Linear search:

1. Simple to implement.
2. Works well with unsorted arrays.

Binary search:

1. Requires a sorted array as input.
2. More complex to implement but generally more efficient than linear search for large datasets due to its logarithmic time complexity.

IV. CONCLUSION

After studying these algorithms we can conclude that the time complexity of linear search is more than that of binary search so we can say that binary search is better than linear search.

The selection between the two depends on factors such as the size of the array, whether it's sorted, and the specific requirements of the application.

REFERENCES

[1]. https://ijarcce.com/wp-content/uploads/2012/03/IJARCCE6C-a-arnab-A-Comparative-Analysis-of-Three.pdf
[2]. https://www.researchgate.net/publication/308119139_A_Comparison_and_Selection_on_Basic_Type_of_Searching_Algorithm_in_Data_Structure
[3]. Data structures using c and c++ by Yedidyah langsam,Aaron M. Tenenbaum second Indian printing (Prentice Hall of India private limited), New Delhi-110001
[4]. https://www.irjet.net/archives/V7/i1/IRJET-V7I1275.pdf