iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** IV **Month of publication:** April 2021

DOI: https://doi.org/10.22214/ijraset.2021.33614

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue IV Apr 2021- Available at www.ijraset.com

A Comparative Analysis of Four Distinct Types of Searching Algorithms in Data Structure

Md. Zillur Rahman[1], Samir Dutta[2], Rahul Rawat[3], Simple Sharma[4]
[1, 2, 3]*Student,* [4]*Associate Professor, Dept of Computer Science and Engineering, Faculty of Engineering and Technology, Manav Rachna International Institute of Research and Studies, Faridabad*

Abstract: ***Basically searching algorithms are used to search or find one or more than one element from a data set. Apparently there are a lot of searching algorithms. But here our research intends to demonstrate an overview of comparison between four different types of searching algorithms. We have tried to cover some part of binary search, linear search, hybrid search, and jump search. This research renders a detailed comparison view of four distinct searching algorithms.***
Keywords: Searching algorithms, binary search, linear search, hybrid search and jump search.

I. INTRODUCTION

The Searching algorithms are used to search or find one or more than one element from a dataset. These type of algorithms are used to find elements from a specific data structures.For example, you might need to find someone's phone number on your phone, or a particular business's address in the UK. In this study we will discuss linear search, binary search, Interpolation search, hybrid search, algorithms on the basis of their efficiency and time complexity.

A. Search Algorithms

1) Binary Search

Binary search is the search technique which works efficiently on the sorted lists. Hence, in order to search an element into some list by using binary search technique, we must ensure that the list is sorted.

Binary search follows divide and conquer approach in which, the list is divided into two halves and the item is compared with the middle element of the list. If the match is found then, the location of middle element is returned otherwise, we search into either of the halves depending upon the result produced through the match

Algorithm

a) [Initialize] set beg = lower_bound
end = upper_bound, pos = - 1

b) Repeat steps 3 and 4 while beg <=end

c) Set mid = (beg + end)/2

d) If a[mid] = val
set pos = mid
print pos
go to step 6
else if a[mid] > val
set end = mid - 1
else
set beg = mid + 1
[end of if]
[end of loop]

e) If pos = -1
print "Value is not present in the array"
[end of if]

f) Exit

Illustration

An array with seven elements, search for "9"

10	23	59	99	11	72	9
10	23	59	99	11	72	9
10	23	59	99	11	72	9
10	23	59	99	11	72	9
10	23	59	99	11	72	9
10	23	59	99	11	72	9
10	23	59	99	11	72	9

2) *Linear Search:* Linear search is the simplest search algorithm and often called sequential search. In this type of searching, we simply traverse the list completely and match each element of the list with the item whose location is to be found. If the match found then location of the item is returned otherwise the algorithm return NULL. Linear search is mostly used to search an unordered list in which the items are not sorted. The algorithm of linear search is given as follows.

Algorithm

Linear_search(a, n, val)

a) [Initialize] set pos = -1

b) [Initialize] set i = 1

c) Repeat step 4 while i<=n

d) If a[i] = val
set pos = i
print pos
go to step 6
[end of if]
set i = i + 1
[end of loop]

e) If pos = -1
print " Value is not presenting the array "
[end of if]

f) Exit

Illustration

Searching for 30 in 7-element array

5	10	15	20	25	30	35
L			M			H
5	10	15	20	25	30	35
L	M	H				
5	10	15	20	25	30	35

3) *Hybrid Search:* There are many search algorithms that can be applied to a set of data. Mostly commonly known and used among them are binary search and linear search. While linear search compares every element of the array with the element to be searched, binary search divides the array into sections and compares the middle element of each section with the key element to be searched. Now, while each algorithm is useful and effective in its own way, each one has its own problem. In binary search, the data needs to be sorted in some order, while in linear search, every element is visited and compared with key element sequentially, and hence takes a lot of time. This Hybrid Search provides an entirely new algorithm which combines the advantages of both the algorithms and provides an effective way to search for a given key element in an unsorted array, in limited time.

Algorithm

a) Mid = (low + high)/2
b) If a[low] = key then return low
c) Else if a[high] = key then return high 4. Else if a[mid] = key then return mid
d) Else if low >= high – 2 then return -1
e) Else
f) P = reclinearbinary(a, low + 1, mid – 1, key)
g) If p = -1
h) P = reclinearbinary(mid + 1, high – 1, key)
i) Return p

Illustration

Searching for 1 in 8-elemetnts array

11	3	55	32	49	1	88	9

11	3	55	32	49	1	88	9

11	3	55	32	49	1	88	9

11	3	55	32	49	1	88	9)

4) *Jump Search:* Jump Search Algorithm is a relatively new algorithm for searching an element in a sorted array. The fundamental idea behind this searching technique is to search fewer number of elements compared to linear search algorithm (which scans every element in the array to check if it matches with the element being searched or not). This can be done by skipping some fixed number of array elements or jumping ahead by fixed number of steps in every iteration.

Algorithm

a) Set i=0 and m = √n.
b) Compare A[i] with item. If A[i] != item and A[i] < item, then jump to the next block. Also, do the following:
Set **i = m**
Increment m by √n
c) Repeat the step 2 till **m < n-1**
d) If A[i] > item, then move to the beginning of the current block and perform a linear search.
Set **x = I**
Compare A[x] with item. If A[x]== item, then print x as the valid location else set x++
Repeat Step 4.1 and 4.2 till **x < m**
e) Exit

Illustration

Searching for 30 in 9-elments array

3	9	11	22	30	33	55	66	79

3	9	11	22	30	33	55	66	79

3	9	11	22	30	33	55	66	79

3	9	11	22	30	33	55	66	79

Comparative Analysis of Searching Algorithms

Algorithm	Advantages	Disadvantages
Binary search	Binary search is an optimal searching algorithm using which we can search desired element very efficiently.	This algorithm requires the list to be sorted
Linear search	When a key element matches the first element in the array, then linear search algorithm is best case because executing time of linear search algorithm is 0 (n), where n is the number of elements in an array.	Inversely, when a key element matches the last element in the array or a key element doesn't matches any element then Linear search algorithm is a worst case.
Hybrid search	Higher performance, accuracy than filter. better computational complexity than wrapper and more flexible and robust upon high dimensional data	Classifier specific methods dependents of the combination of different feature selection method
Jump search	Jump search algorithm is more efficient in case of finding a element 600 out of 625 elements in an array.	Jump search algorithm is not preferable for unsorted list or array.

II. CONCLUSION

The paper describes about numerous searching techniques and their algorithm. It demonstrates the methods for various searching techniques.

Searching is one of the most important and required operation of data structure in many places. Searching algorithms allows us to look for a specific data in the entire list of data. The analysis has clearly demonstrated the pros and cons of various searching algorithms. On analysis, we've found

that binary search is precise for average size data items

and is applicable in arrays and in linked list. Linear search is good when we want to do sequential wise searching. Whereas jump search is accurate for huge number of data items. Also we found that Hybrid search used for unsorted list with a large number of elements.

III. ACKNOWLEDGEMENT

At first, we would like to thank our god the creator of all and the most merciful.

Our teacher Simple Sharma, she has immensely helped us to complete the this paper and kept us motivated the whole time.

Our parents for rendering the required serenity to write paper at such time of pandemic.

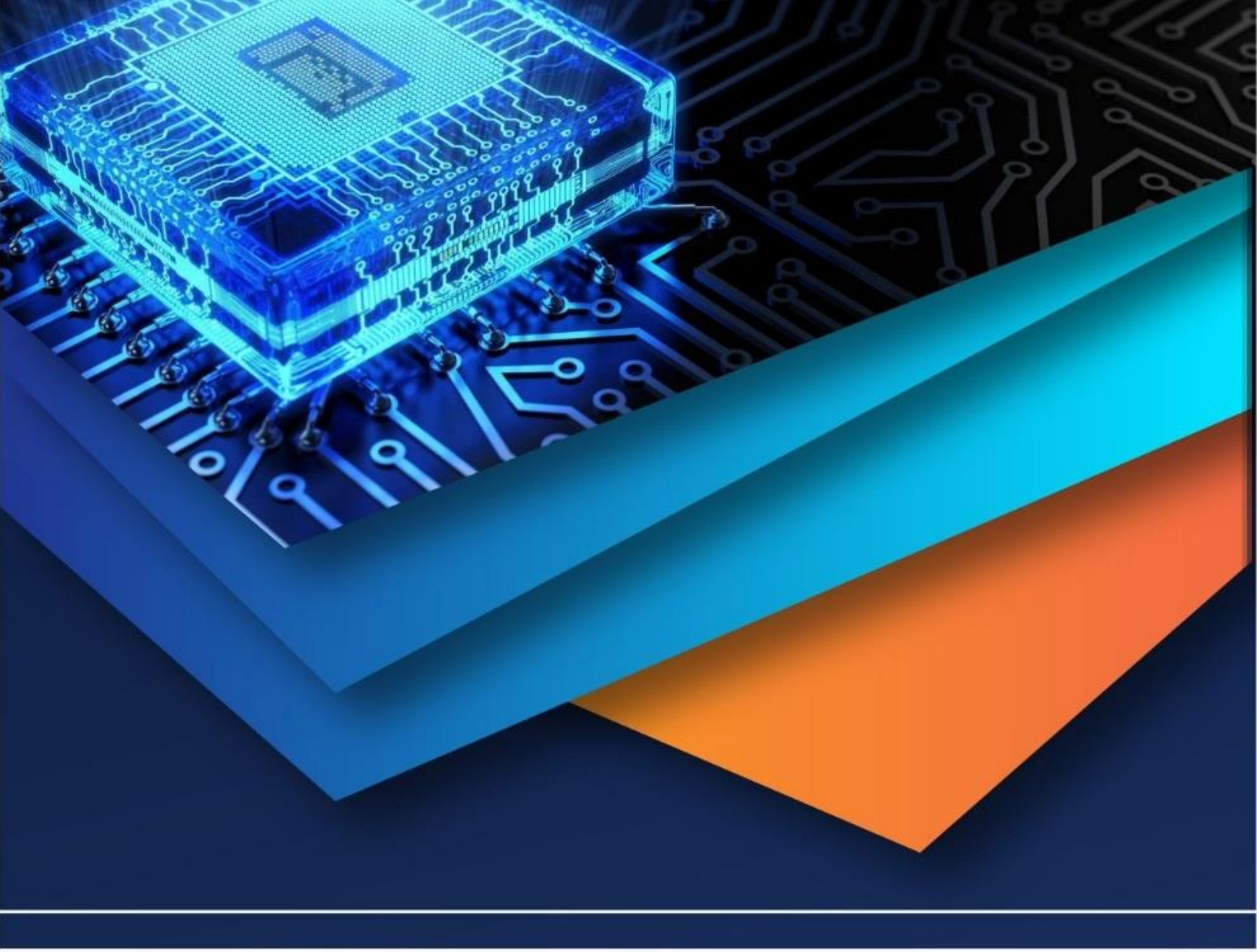
REFERENCES

[1] A comparative analysis of three different types of searching algorithms in data structure, Debadrita Roy, Arnab Kundu, International Journal of Advanced research in computer and communication engineering (IJARCCE).

[2] A comparison Based analysis of different types of sorting and searching algorithms in data structure, Patel Pooja, International Journal of Advance Engineering and research development (IJAERD).

[3] Comparative analysis on sorting and searching algorithms, B Subbarayudu, L Lalitha Gayatri, P Sai Nidhi, P Ramesh, R Gangadhar Reddy, Kishor Kumar Reddy C, International Journal of civil engineering and technology (IJCIET).

[4] A survey on Different Searching algorithms, Ahmad Shoaib Zia, International Research Journal of engineering and technology (IRJET)



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY