DATA STRUCTURES:SEARCHING TECHNIQUES

SIDHANT MAHAJAN
CHANDIGARH UNIVERSITY, MOHALI
PUNJAB, INDIA
Sidhant.mahajan23@gmail.com

Hatesh Shyan

Assistant Professor,
Department of CSE,
Chandigarh University

hatesh.cse@cumail.in

***Abstract*—**In computer science a search algorithm is any algorithm which helps us in the retrieval of information which is stored within some data structure.

Keywords—Linear search, Binary search, data structure

I. INTRODUCTION

In computer science a search algorithm is any algorithm which helps us in the retrieval of information which is stored within some data structure. Search algorithm can be based on the mechanism or method of searching. Some of the basic searching technique includes

- Linear Search
- Binary search

In linear Search, algorithm checks every record for the one with a target key in a linear fashion. It works on the technique of traversing a list from beginning to end by checking properties of all the elements that are found in thelist.

In Binary Search, algorithm repeatedly targets the center of the search structure and divides the search space in half

2. SEARCHING

Searching is the process of finding a given value. It decides whether the key is present in the data or not. It is the logical process of finding a particular item in a list of items.

2.1 LINEAR SEARCH

Linear Search is also known as Sequential search having complexity O(n). The search algorithm starts at the beginning of the list and checks every element of the list. It is a simple search algorithm. Linear search

compares the element with all the other elements given in the list. If the element is matched, it gives the value of index; else it print that element is not present in the list.

ALGORITHM OF LINEAR SEARCH

- 1 Set b to 1
- 2 if b>n print element not found
- 3 if A[b]=x print element found at index b
- 4 set b= b+1;
- 5 go to 2
- 6 exit

PSEUDO CODE OF LINEAR SEARCH

```
For b=0 to length of list
If list[index]==search item, then
Return index
Else
Return item not found
```

Logic of linear search

```
For(b=0;i<n;b++)
{
If (arr[b]==x)
Retrun 1
Else
Return -1
}
```

2.2BINARY SEARCH

In computer science binary search is also known as logarithmic search. Most easy search in searching algorithm is linear search but the time complexity of linear search is O(n)which means it will take n amount of time, where n is the number of element. But in case of binary search the time complexity of the algorithm is O(log n)that means this algorithm(binary search)will take less amount of time as compared to linear search. Binary search is faster than linear search but there is a condition of binary search, that elements in

the list must be sorted .it can only be implemented only if list is sortedBinary search works on the algorithm of divide and conquer. It works on sorted arrays . Binary search starts with comparing the middle element of list with the key. 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 middle element then the search continues in right half.

2	4	5	6	8	23	45	67	458	500	800

In this example we have to search five. First we will find the middle of the list if key value is less than mid then find the mid in lower list recursively until the value is not found

2	4	5	6	8

In this we have found the value at position 3:

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 lower half using recursion

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

Pseudo code of binary search

1 set beginning=1 end=n mid =(beginning +end)/2

2 while x not found

3 if beg>end, then x does not exist

4 if mid<x set beg =mid+1

5 if mid>x set end =mid-1

6 if mid=x; item found

End while

Logic of binary search

```
Function binary (A, n, key):
beg := 0
end := n −1
whilebeg<= end:
m :=(beg+end)/2
if A[m] <key:
beg := m + 1
else if A[m] >key:
end:= m - 1
else:
return m
returnitem not found
```

3 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)

4 Conclusion

After studying these algorithm 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 that of linear search.

References

1. W. Sarada, Dr. P. V. Kumar, “A Comparative Study and Analysis of Searching and Sorting algorithms” , International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). Volume 5, Issue 5
2. Debadrita Roy, ArnabKundu,” A Comparative Analysis of Three Different Types of Searching Algorithms in DataStructure” International Journal of Advanced Research in Computer and Communication Engineering) Vol. 3, Issue 5, 2014
3. Taneja, Kavita, HarmunishTaneja, and Rohit Kumar. "Multi-channel medium access control protocols: review and comparison." *Journal of Information and Optimization Sciences* 39.1 (2018): 239-247.
4. S.K. Shrivastava, DeepaliShrivastava,”Searching, Hashing and Storage Management”, “Data Structures Through C InDepth”, BPB Publications, Eighth Edition,ISBN:81-7656- 741-8
5. Savina&SurmeetKaur, “Study of Sorting Algorithm to Optimize Search Results”, International Journal of Emerging Trends & Technology in Computer Science, Volume 2, Issue 1.

6. LUEKER, G. "A transformation for add- mg range restriction capabdlty to dynamtc data structures for decomposable search-mg problems," Tech. Rep. 129, U of Califorma at Irvine, 1979.
7. Yehoshua Perl Bar-Ilan University and The Weizmann Institute of Science,AlonItaiTechnion--Israel Institute of Technology, HaimAvni The Weizmann Institute of Science. Interpolation Search A LogLogNSearch.
8. Falley. P “Categories of Data Structures”, Journal of Computing Sciences in Colleges - Papers of the Fourteenth Annual CCSC Midwestern Conference and Papers of the Sixteenth Annual CCSC Rocky Mountain Conference. Volume 23 Issue 1, October 2007. PP. 147-153, 2007-10-01
9. Pogue, C. & Willett, P. (1987). Use of Text Signatures for Document Retrieval in a Highly Parallel Environment. Parallel Computing, 4, 259-268.