International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 559-564
Published Online August 2020 in IJEAST (http://www.ijeast.com)

AN INDEXED SEQUENTIAL SEARCH AND ITS COMPARATIVE ANALYSIS WITH BASIC SEARCHING TECHNIQUES

Ms. Komal S. Deoghare
Sant Gadge Baba Amravati University,
Amravati, Maharashtra, India

***Abstract*— The data and file structures need the techniques to perform searching or locating operation of elements. In the data structure, list can be sorted or unsorted. Depending upon sorting, different searching algorithms are evolved. In this paper Indexed Sequential Search is studies and compared with basic searching techniques such as Linear Search and Binary Search. These techniques are reviewed with working of the three techniques and their comparison analysis with respect to time and space complexity and finally the basic implementation using C programming.**

***Keywords*— searching, linear search, binary search, indexed sequential search, time and space complexity**

I. INTRODUCTION

To find a particular element from a data structure, Searching is used as the basic operations on both sorted and unsorted list of elements. Depending on the type of data structure, a suitable search algorithm is chosen[1].A search typically answers the user whether the item he searched for is present or not. The research in the Computer systems are carried on to store large amounts of data on a disk from which individual records can be accessed, retrieved according to some of search conditions and the process will be done in less time and complexity. [2] For this purpose some approaches are needed that not only saves our time but also fetches the required data efficiently. In this study we will discuss linear search, binary search, Indexed sequential search algorithms on the basis of their efficiency and time complexity.

Searching falls into two categories:

a) External searching: External searching means searching the records using keys where there are many records, which resides in the files stored on disks. In this kind of searching technique, the data is stored in secondary memory such as hard disk drive or any external storage device. [2][3].

b) Internal searching: Internal searching is that type of searching technique in which there is fewer amounts of data which entirely resides within the computer's main memory. In this technique data resides within the main memory on[2][3].

There are two basic algorithms available:

1.Linear Search

2.Binary Search

Linear Search

Linear search is a one of the basic searching technique. In Linear search, the an element or value in a given array is searched by traversing the array. The traversing is continued till the desired results are found.

Binary Search

The binary search algorithm consists of the principal of method of divide and conquer. To work this technique fine, the data collected and pre-sorted.

To improve the searching efficiency for the sorted file, *Indexed Sequential Search* is introduced. [4]

II. BASIC CONCEPT BEHIND INDEXED SEQUENTIAL SEARCH

Indexed Sequential search method deals with creating the index file. The index file contains the refrences to the group of records. When the index is searched, the partial indexing takes less time because it is located in a specified group or bucket. When the programme searches for specific record, it will find group or bucket of index and then specific record is placed. Indexed Sequential Search actually does the indexing multiple time, like creating the index of an index[5].

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 559-564
Published Online August 2020 in IJEAST (http://www.ijeast.com)

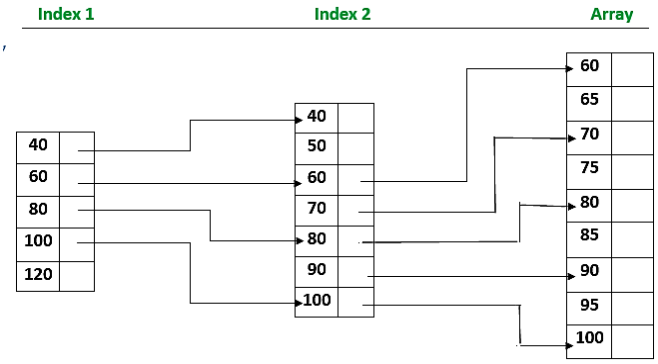

Fig.1. (a) : Indexing in Indexed Sequential search[4].

Actually, the concept of Indexed sequential search is came from ISAM (Indexed sequential Access Method). It is a method for creating, maintaining, and manipulating computer files of data so that records can be retrieved sequentially or randomly by one or more keys. To achieve the quick and efficient retrival of required file records in index files, the Indexes of key fields are maintained. Originally IBM developed Indexed Sequential Access Method (ISAM) for mainframe computers. Nowadays, the implementation of ISAM is available for most of the computers [4].

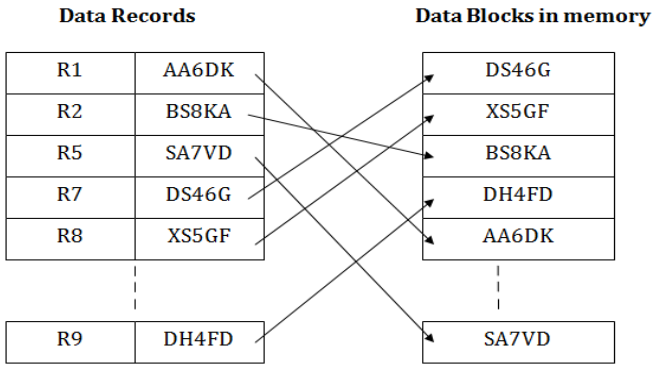

Fig.1. (b) : Indexed Sequential Access Method-Example

Advantages of Indexed Sequential Access Method (ISAM):

- In this method, each record has the address of its data block, searching a record in a huge database is quick and easy.
- This method supports range retrieval and partial retrieval of records. Indexing is based on the value of the primary key.
- The searching process involves the retrival of data for given range. Also, the partial value of the index can be searched effortlessly.

Disadvantages of Indexed Sequential Access Method (ISAM):

- As the data processing is large scale, this method requires extra space in the disk to store the index value.
- When the new records are inserted, then these files have to be reconstructed to maintain the sequence.
- When the record is deleted, then the space used by it needs to be released to maintain the performance of the system[4].

III. BASIC WORKING PROCEDURE

A. Linear Search :

The basic searching techniques is the Linear search. It is also known as sequential search. In this process of searching, the element is searched by comparison of the given element with all the elements of the given list starting from first element to last [6].

As this technique involves searching element from start to end, it is obvious that the time required for searching that element depends upon the number of elements accessed for comparison. If the element found near the starting element. it requires less time as compared with the element found near the end of the list. If the element does not exist at all in the list, this process requires maximum time. Because it needs maximum comparisons This algorithm or technique is used to search an unordered list in which the items are not sorted mostly. [7].

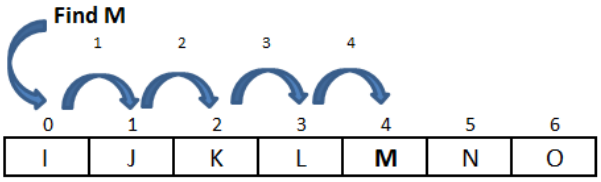

Fig. 2: Linear Search-Example

B. Binary Search :

This binary searching technique involves searching of element within a sorted array. In this searching technique, the given element or value is comapred with the middle element of sorted array. If such element is found, the algorithm ends itself or otherwise it reduces the searching space to half of its total size of array.

If search element is less than the middle element then only first half of the list is searched and in case if it is greater than the middle element then only second half of the list is searched in next iteration. In either of the case the list to be searched is reduced to half at each iteration of the search[7].

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 559-564
Published Online August 2020 in IJEAST (http://www.ijeast.com)

The binary search consists of divide, conquer and combine methods. That can be given as follows:

Divide: The given array is sorted and divided into sub parts. In binary search algorithm, the array or list is sorted and divided into two sub parts.

Conquer : Searhing process is carried out individually in each sub parts. The element to be searched is comapared with the middle element of the sorted list. If the given element is less than the middle element then move towards leftmost item otherwise go to the rightmost item or element in the list.

Combine: The search results that are obtained by divide and conquer then are combined and returned by algorithm wheather the element is found or not found. [8][9].

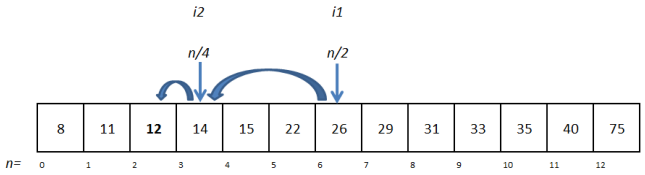

Fig. 3. : Binary Search- Example

In above example of binary search, here the item 12 is to be searched in this array. The array is sorted and divided into two parts in first iteration and compared with the middle element of array at *n/2* that is 26. Then searching is performed on left side, here is second iteration of the algorithm. The left half is further divided into two halves *n/4*. Then compared with the middle item that is 14 and then search left and finally the item is found.

C. Indexed Sequential Search:

In the Indexed sequential search, sorted file is processed along with the additional table called as Index file. The index table consists of a key and reference pointer to the record in the field. Note that, the index file as well as the file to be searched in are sorted on the file. It consists of multiple level index like index of index. It searches through the hierarchical order of index.

This algorithm is basic used for searching an indexed sequential file. In the following example, the index file consists of the keys and pointer index that points the record in the file.

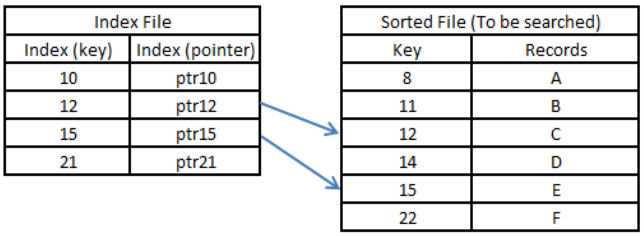

Fig. 4. : Indexed Sequential Search- Example

The index file is maintained is the best advantage of this method. The index is processed and searched sequentially the search time for particular items is reduced. This searching algorithm involves the processing of index file rather than whole big file. This is an advantage of this algorithm. After search of index is performed, a corrosponding record is returend. [4]

IV. Comparison Analysis

In this section, we are going to compare the different searching techniques in complexity point of view. Complexity of algorithm can be determined by the time complexity and space complexity. The comparison among Linear search, Binary Search and Indexed sequential search on the basis of time and space complexity.

Time complexity: Time complexity is the amount of time the computer requires to execute the algorithm. The number of operations performed by an algorithm is used to calculate time complexity, where each operation takes a pre-set amount of time[8] [9].

- Best case : fastest time to complete, with optimal inputs chosen.
- Worst case : slowest time to complete, with pessimal inputs chosen.
- Average case : It is determined by the mean of time taken to compute the particular algotithm with *n* inputs. Simply it is an average of time taken for computation of *n* searches.

Space complexity: An algorithm runs in the memory of computer. It utilizes some portion of memory that depends upon the memory required to store different data type used in the algorithm. So, the amount of memory space the algotithm requires for completing the execution of the algorithm is called as Space Complexity of an algorithm [10] [11].

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 559-564
Published Online August 2020 in IJEAST (http://www.ijeast.com)

Algorithm	Best Time Compl-exity	Average Time Compl-exity	Worst Time Compl-exity	Worst Space Compl-exity
Linear Search	0(1)	0(n)	0(n)	0(1)
Binary Search	0(1)	0(logn)	0(logn)	0(1)
Indexed Sequential Search	0(1)	0(n)	0(n)	0(1)

Table -1 Comparison of different basic algorithms with respect to the Complexity of Time and Space

V. EXPERIMENTAL WORK

The basic operations of searching can be illustrated by simple programmes. In this section the basic programmes are written in C programming language. These three different programmes, provide the searching using Linear Search, Binary Search and Indexed Sequential Search each.

A. Linear Search

```
#include<stdio.h>
void main ()
{
    int a[12] = {8, 9, 12, 15, 22, 17, 24, 20, 10, 13, 27, 6};
    int item, i,flag;
      printf("Enter Item to be Searched in given array:");
      scanf("%d",&item);    /*Takes the input*/
      for (i = 0; i< 10; i++)
    {
                    if(a[i]    ==    item) /*Compares the item with array elements*/

            {
                        flag = i+1;
                        break;
            }
            else
            flag = 0;
    }
            if(flag != 0)
            {
                  printf("\nItem found at location: %d",flag);
            }
    else
    {
       printf("\nItem not found in an array");
    }
```

Output:

```
Enter Item to be searched in given array: 22
Item found at location 5
```

B. Binary Search

```
#include<stdio.h>
int binSearch(int[], int, int, int);
void main ()
{
    int arr[12] = {6, 8, 9, 10, 12, 13, 15, 17, 20, 22, 24, 27};
    int item, location=-1;
    printf("\nEnter the item to be searched in given array: ");
    scanf("%d",&item);
    location = binSearch(arr, 0, 11, item);
    if(location != -1)
    {
        printf("\nRequired item is found at location %d\n",location);
          }
          else
          {
              printf("Item is not found in given array\n");
          }
      }
      int binSearch(int a[], int beg, int end, int item)
      {
          int mid;
          if(end >= beg)
          {
              mid = (beg + end)/2; /*Divide into two parts*/
              if(a[mid] == item)
              {
                  return mid+1;
              }
              else if(a[mid] < item) /*Compare item with middle item*/
              {
                  return binSearch(a,mid+1,end,item);
              }
              else
              {
                  return binSearch(a,beg,mid-1,item);
              }
```

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 559-564
Published Online August 2020 in IJEAST (http://www.ijeast.com)

```
            }
            return -1;
        }
```

Output:

```
nEnter the item to be searched in given array:9
Required item is found at location 3
```

C. Indexed Sequential Search

```
#include <stdio.h>
#include <stdlib.h>
void indexedSeq(int arr[], int n, int k)
{
    int                array_elements[20],
array_indices[20], temp, i;
    int j = 0, ind = 0, beg, end;
    for (i = 0; i < n; i += 3) {
        array_elements[ind] = arr[i];
        array_indices[ind] = i;
        ind++;
    }
    if (k < array_elements[0]) {
        printf("Element not found");
        exit(0);
    }
    else {
        for (i = 1; i <= ind; i++)
            if (k < array_elements[i]) {
                beg  =  array_indices[i  -
1];
                end = array_indices[i];
                break;
            }
    }
    for (i = beg; i <= end; i++) {
        if (k == arr[i]) {
            j = 1;
            break;
        }
    }
    if (j == 1)
        printf("Found at index %d", i);
    else
        printf("Index is Not found");
}
void main()
{
    int arr[] = { 10,11,12,14,17,20};
    int n = sizeof(arr) / sizeof(arr[0]);
    int k = 12;
    indexedSeq(arr, n, k);
}
```

Output:

```
Found at index 2
```

VI. Conclusion

The data and file structures need the different searching techniques to locate and process the data. The files are organised, optimized and particular records are searched. These different searching techniques provide the improvement in performance of the applications. In this paper, the basic searching techniques are described and reviewed with basic concept, their comparison and implementation in C programming language.

VII. References

[1] Sultana N., Paira S., Chandra S. And Alam S. K. S., "A Brief Study And Analysis Of Different Searching Algorithms," 2017 Second International Conference On Electrical, Computer And Communication Technologies (Icecct), Coimbatore, 2017, (Pp. 1-4), Doi: 10.1109/Icecct.2017.8117821.

[2] Zia A. S. 2020 "A Survey On Different Searching Algorithms" International Research Journal Of Engineering And Technology (Irjet) Volume: 07, Issue: 01.

[3] Chandrawat Y. S., Vajpayee A., Pathak A. (2015), "Analysis And Comparative Study Of Searching Techniques" International Journal Of Engineering Sciences & Research Technology(Ijesrt).

[4] F. Zaqout, M. Abbas and A. Hosam, "Indexed Sequential Access Method (ISAM): A Review of the Processing Files," 2011 UKSim 5th European Symposium on Computer Modeling and Simulation, Madrid, 2011, (pp. 356-359), doi: 10.1109/EMS.2011.13.

[5] Samadder S., 2018, https://www.geeksforgeeks.org/ indexed-sequential-search/

[6] Dr. Parmar V. P. , Dr. Kumbharana CK , 2017 "Designing And Implementing Data Structure With Search Algorithm To Search Any Element From A Given List Having Constant Time Complexity" International Education & Research Journal [IERJ] Volume : 3,Issue : 1, (Pg 109-111).

[7] Xavier S. , Jisna V A, Bineesh M., "How to Select a Searching Algorithm-A Comparative Study" ,International Journal of Computer Trends and Technology (IJCTT) ,Volume 48, Number 3, 2017 (Pg 108-111).

[8] Patel P., 2015, "A Comparison Based Analysis of Different Types of Sorting and Searching Algorithms in Data Structures" International Journal of Advance Engineering and Research Development Volume 2,Issue 6, (Pg 401-414).

[9] Pandey K.,Pradhan N., 2014 "A Comparison and Selection on Basic Type of Searching Algorithm in Data Structure" International Journal of Computer Science and Mobile Computing(IJCSMC), Vol. 3, Issue. 7, (Pg 751-758).

[10] Roy D., Kundu A.,2014" A Comparative Analysis of Three Different Types of Searching Algorithms in Data Structure", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 5(Pg.6626-6630).

[11] Subbarayudu B ., Gayatri L. L., Nidhi P. S., Ramesh P., Reddy R. G. Reddy C.K.K., 2017. "Comparative Analysis On Sorting And Searching Algorithms" International Journal of Civil Engineering and Technology (IJCIET), Volume 8, Issue 8, (Pg 955-978).