International Journal of Scientific & Engineering Research Volume 11, Issue 7, July-2020
ISSN 2229-5518

To-Fro Sequential Search Algorithm – An Alternative to Linear Search

Sumanth.D.R
Department Of Computer Applications
Seshadripuram College
Tumkur, Karnataka,India
Email: drgowda6@gmail.com ,sumanthdr@yahoo.com

Abstract-: This paper explores, a new approach towards finding a particular data from an array. Linear Search is one such algorithm, which is used to find a particular data. In this paper, a new Linear Searching technique is proposed, for search of an element from an unsorted array. The To-Fro Sequential Search algorithm starts scanning the elements from the middle of the array.
The algorithm divides the array into two half's i.e. by calculating:
(Number of array elements) / 2.
The resultant is considered as midpoint, dividing array into two half's: left part and right part. The algorithm traverse from left part to right part and from right part to left part until the search element or key element is found. As the traverse moves left to right and right to left, hence the name To-Fro Sequential Search.

Keywords-:To-Fro Sequential Search, Linear Search, Algorithms, Searching, Traversing.

1. INTRODUCTION

Search technique is one of a major operations of Data Structures, which is to find a particular element from a stack of elements.
In this present scenario, a design to refine search algorithm is essential, since searching an element from a large amount of data, has an implication in Computer Applications.

The To-Fro Sequential Search, initially divides the array into two half's. It is divided by calculating: (Number of array elements)/2.
This resultant is considered as midpoint of the array. Now elements that are present in the left side of midpoint is considered as left part and the elements that are present in the right side of midpoint is considered as right part. The midpoint is always, the first array index scanned with key element. In To-Fro Sequential Search, there are two iterators, first for left part array and the other for right part array. First iterator traverses sequentially the array from right to left, where as other iterator traverses sequentially the array from left to right. When any one of the iterators encounters An element which matches with the key element, then search stops and index of that element is returned.

2. RELATED WORK

Linear Search finds a given element in a list of unsorted elements. This algorithm starts scanning the elements from first element of the list, and traverses until a match is found.
When a match is found, the index value of that particular element is returned.

In Linear Search, if we give an key element that is not present in the list or a key which is the last element of an array, then O(n) is the time complexity, where n is total number of elements in the array.
Algorithm: Linear Search (A[], n, key)

Input: Array A, n, key

Where-: n - Number of array elements
key - key element to be found

Output: Position i,$0 \leq i < n$ such that A[i]=key, or -1 if x does not exist in A.

1. for i=0 to n
2. if a[i]=key then return i
3. else loop
4. if i=n then return -1

3. TO-FRO SEQUENTIAL SEARCH ALGORITHM

To-Fro Sequential Search is a new approach to Sequential Search, where scanning starts from the middle element of the array. It is not compulsory to arrange the array in sorted order only, as in case of binary search. The array is divided into two parts left part and right part. In To-Fro Search, scanning always starts from the last element of left array which is also the mid element, so the best case time complexity input will be the mid element of given array. The worst case input time complexity will always be the first element of the left array and if the array element is not in the list.

Algorithm: TroFro Search(A[], n, key)

Input: Array A,n,key,mid

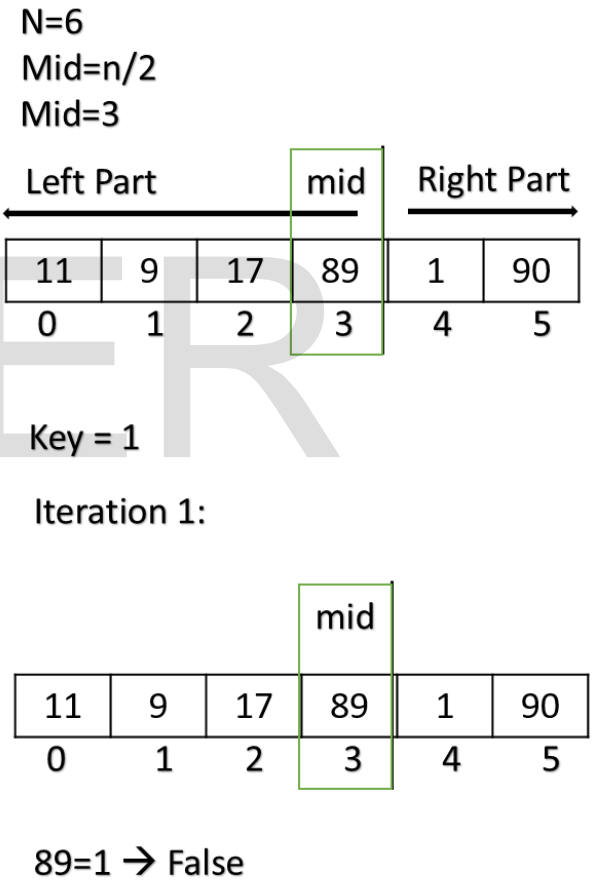
Where-: n - Number of array elements
key - key element to be found
mid=(n/2)

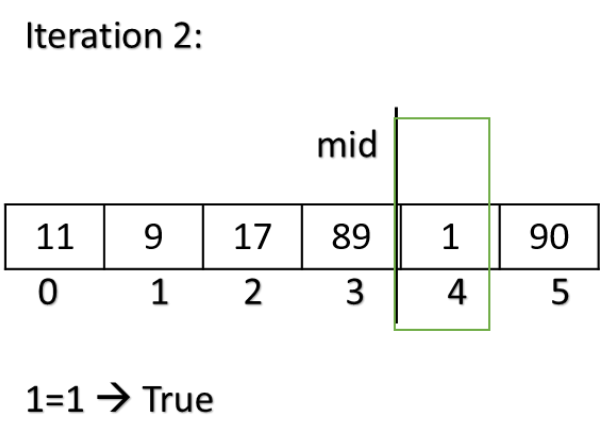
Output: Return i if num to belongs A[i], or Return j if num belongs to A[j].
Steps:

1. **START**
2. mid ← n / 2
3. i ← mid
4. j ← mid + 1
5. flag ← 0
6. **while** ($i \geq 0$)
7. **while**(j < n)
8. **do**
9. **if** (a[i] = key) **then** return i + 1 and; **flag ← 1;** break
10. **if**(a[j] = key) **then** return j + 1 and; **flag ← 1;** break
11. i ← i - 1
12. j ← j + 1
13. **if** (flag = 0) **then** print "Not found"
14. **STOP**

3.1.1. WORKING OF TO-FRO SEQUENTIAL SEARCH

Example:





In the above figure explains how To-Fro Sequential Search works. This search technique iterates only two times to find the 4th index array, where in the traditional linear search it would take five iterations to search the 4th index array.

4. PROOF OF CORRECTNESS FOR TO-FRO SEQUENTIAL SEARCH

In order to prove that an algorithm is correct, we need to show that the algorithm terminates and it produces correct output.

The variable 'i' takes value 'mid' and variable 'j' takes value 'mid+1'. After finite number of iterations the value of 'i' keeps on decreasing and the value of 'j' keeps on increasing, however infinite loop is not formed, because of the condition $i \geq 0$ and $j < n$.

Here the left part array which is indexed as 'i' terminates when it reaches the first element. The right part array which is indexed as 'j' terminates when it reaches the last element.

4.1. TO-FRO SEQUENTIAL SEARCH PRODUCES THE CORRECT OUTPUT

In To-Fro Sequential Search if the key element if present in left part of the array then (i+1) value is printed or, if the key element is present in right part of the array then (j+1) value is printed.

5. EFFIENCY ANALYSIS

Linear Search and To-Fro Sequential Search were implemented in Visual Studio 2013 using C#, and was tested for the input length 100000, 500000. Both the algorithms were executed on a machine with 64-bit Intel(R) Core(TM) i5-7200 CPU @2.50GHz, and installed memory (RAM) 8.00GB.The values of input length and CPU time taken (msec) is shown in Table 1.Result shows that To-Fro Sequential Search works well for all input values and it takes lesser time, to search the element which are after the middle of the array, otherwise it takes more time than Linear Search.

Table 1. CPU Time (msec) for different length of input sequences

	Element present at starting position. Input size-100000	**Element present at middle position. Input size-100000**	**Element present at last position. Input size-100000**	**Element present at starting position. Input size-500000**	**Element present at middle position. Input size-500000**	**Element present at last position. Input size-500000**
Linear Search	**1**	**0**	**0**	**3**	**3**	**2**
To-Fro Sequential Search	**14270**	**0**	**0**	**381138**	**2**	**2**

6. CONCLUSION AND FURTURE SCOPE

To-Fro Sequential Search is a searching technique that is used to find a particular element in a array. Table 1 shows that the To-Fro Search works best when the array elements are searched after the middle of the array, and takes maximum time to search array elements which are before the middle of the array. Even though the To-Fro Search takes more time to search the first element, it takes very less time in case of finding the middle and the last element of the array.

7. ACKNOWLEDGMENT

This research paper is made possible through the help and support of the faculty members of the Dept. Of Computer Applications, SDC. I would like to extend my gratitude to Prof.Prathibha.S.B (HOD, SDC) which would have not been possible, without their immense support and guidance.

8. REFERENCES

[1] Nitin Arora, Garima Bhasin and Neha Sharma. Two Way Linear Search Algorithm. Internation Journal Of Computer Applications. Volume 107 – No. 21, December 2014.

[2]Mridul Bhaumik, Subir Saha, Supratim Das. A New Modified Linear Search. https://www.reasearchgate.net/publications/309609322. Working Paper - November 2016.