Journal of Applied & Computational Mathematics

Behera, J Appl Computat Math 2016, 5:4
DOI: 10.4172/2168-9679.1000322

Research Article OMICS International

A Search Algorithm

Behera BC*

Institute of Mathematics & Applications, Ism Dhanbad, India

***Corresponding author:** Bishnu Charan Behera, Institute of Mathematics & Applications, Ism Dhanbad, India, Tel: +91 674 238 6972; E-mail: bishnu_charan_behera@hotmail.com

Received date: May 11, 2015, **Accepted date:** August 15, 2016, **Published date:** August 31, 2016

Abstract

This is an algorithm which has the same time complexity as that of linear search of "O (n)". But still it is better than "linear search" in terms of execution time. Let A[ ] be the array of some size N. If the element which we want to search is at any position before "N/2" than "my-search and linear-search" both will have execution time, but the magic happens when the search element is after "N/2" position. Suppose the element want to search is at nth position, then using the linear search will find the element after nth iteration, but using "my-search" we can search the element after 1st iteration itself. Elements in (N-i)th position can be found in the (I+1)th iteration i.e. suppose size is 1000 than element in 1000th position can be found in 1st iteration, similarly 999 in 2nd iteration and process goes on like this.

Keywords: Algorithm; Linear search; Searching techniques; Data structures

Introduction

You must have read searching techniques like linear search, binary search, etc. which is being taught in data structures to find an element in an array. But this search is a new way different from the above mentioned. You will get details as you proceeds.

When we are dealing with a situation when size is something 10 or 15 its ok. But can you imagine the case when the size is "100000000" or equivalent. If we use this "linear search" technique than the total expenditure you can think off to continue the loop for 100000000 times. But rather if we use "my-search", we get the desired search just after 1 iteration [1-4].

So, now can you imagine how we can prevent such a big loss through "my-search".

Algorithm

This is an algorithm which i have developed and named as "my – search". It will search a required element from the array [5-7].

```
Algo-My search
VAR:A[ ],N,I,J,C,ITEM;
I=1,J=N,C=0;
WHILE((I<N/2)||(J>=N/2))
  {
    IF(A[I]==ITEM)
    {
        WRITE"ELEMENT FOUND";
        C++;
        EXIT();
    }
    ELSE IF(A[J]==ITEM)\
    {
        WRITE"ELEMENT FOUND";
        C++;
        EXIT();
    }
I++;
    J--;
    }
IF(C==0)
    {
    WRITE "ELEMENT NOT FOUND";
    }
```

Structure

56	67	32	23	12	45	78	2
→0	1	2	3	4	5	6	7 ←

Future Needs

As I mentioned above this makes the searching techniques are less complex, so it can be widely be used in industrial purpose. We know that in industries time matters more than anything which can be resolved by this "my-search". Finding element from thousands of entries is not an easy task and we can do it easily.

Conclusion

Hence from the above algorithm we concluded that above algorithm is better than "linear search" algorithm in terms of execution. Using this we can make searching techniques less complex.

References

1. Mehlhorn K (1977) Data Structures and Algorithms. Springer Verlag.
2. Sutton RS, Barto AG (1998) Reinforcement learning: An introduction. MIT press, Cambridge p: 122.
3. Lewis HR, Deneberg L (1991) Data Structures and Their Algorithms. Addison-Wesley 30: 452-462.
4. Schaum S (2016) Data Structures with C. Schaum Series.
5. Wirth N (1978) Algoritms + Data Structures = Programs Prentice Halls, ISBN: 0130224189.
6. International Journal of Data Structures and Algorithms
7. Sedgewick R, Flajolet P (2013) An introduction to the analysis of algorithms. Addison-Wesley.