Proc. of the Second Intl. Conf. on Advances in Computer, Electronics and Electrical Engineering -- CEEE 2013

ISBN: 978-981-07-6260-5

A Conceptual Method for Searching

A positive integer searching algorithm

Mohammad Hamid Rafique Shaikh

***Abstract*—In this paper I proposed searching technique for positive integer number. In which time complexity is depends on the number of digit of search element (i.e. if searching number is 141 than time complexity will depends on 3) despite of number of elements a list contain.**

Keywords—Time complexity, List.

I. Introduction *to Searching*

Searching is a fundamental for computer processing, so any attempt to design fast methods for this operation is important. Earlier Searching techniques are depends on the number of element in the list like in binary search algorithm the searching algorithm depends on number of element a list contain i.e. if we want to search 145 and the list contain 10,000 integer number then the time complexity depends on 10,000 and average case will be 10,000/2.

II. My Approach

First, Instead of searching the entire element I started to look at the number that I have to search from given N numbers then I saw that every number that we have to search has some attributes that makes it constant and unique. Every number has its own unique attributes like number of digit that it has, value of 1st digit etc. So based on this attribute I started to store it on the basis of this attribute and search it accordingly.

I have consider an example of library which contain thousands of books and each book has its own unique location and number based on that a librarian can easily find out a book that a reader wants. So I thought why not a positive number which has unique attributes can be placed in such a way that it can be search by its unique attributes.

Mohammad Hamid Rafique Shaikh
MHSSCOE
India

III. Proposed Technique

Proposed technique is based on the attributes (that has described before) of a searching number and placement of the N numbers.

We want to store the entire number based on its unique attributes like number of digit and value of 1st digit by given simple pseudo code.

Number of digit of search element (D):

```
while (x > 0)
   {
     r = x % 10;
    D++;
    x =x/ 10;
   }
```

Value of 1st digit (F):

```
x = Math.abs(x);
F=(int)Math.floor(x/Math.pow(10,Math.floor(Math.log10(x)))
```

And stored accordingly like if we want to store 345 than calculate number of digit i.e. 3 and Value of 1st digit i.e. 3 and placed it accordingly. Now if we want a number that has to be search (like 145) from given stored N numbers then we have to calculate its unique attributes like number of digit and value of 1st digit by given same above pseudo code.

After calculating these attributes send it to there where the three digit number and value of 1st digit is 1. Suppose we want to store only up to 3 digit numbers i.e. maximum element are 999 (or more if the some number is repeated).

Consider if number is not repeated then based on its unique attributes, maximum number of 1 digit element will be 1, maximum number of 2 digit element will be 10, maximum number of 3 digits element will be 100 and so on, despite of numbers of element a List contain.

A. *Abbreviations and Acronyms*

Number of digit is D accordingly if number of digit is 1 then D1 and so on and Value of 1st digit is F accordingly if value of 1st digit is 1 then F1 and so on. Let α be the number of digit of element that we have to search, β be the 1st value of that element and ρ be number of value that we have entered in a list with respect to α and β, σ be the number of element that is repeated in a particular List, List corresponds to Array list.

Proc. of the Second Intl. Conf. on Advances in Computer, Electronics and Electrical Engineering -- CEEE 2013

ISBN: 978-981-07-6260-5

B. *Pseudo code:*

a) Storing:

1) Take input and calculate its unique attributes (D and F)
2) Based on D and F store it to their respective List
3) If list is not present create List
4) If end, stop else go to step 1.

b) Searching:

1) Take the element to be search and calculate its D and F.
2) Based on D and F send it to search respective List.
3) Search the List
4) If element present send success message else not present.

Consider we want to store up to a 3 digit number, then maximum number that we can store are 999 (if element is not repeated). Let D1, D2, D3 are the number of digit i.e. 1, 2, 3 respectively and F1 to F9 are the value of 1st digit then it can be graphically presented as follow

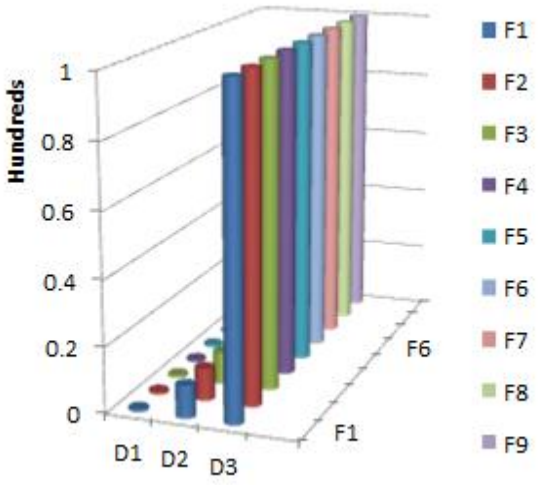

Figure 1: Storing the element in List

IV. Searching from list

After storing 999 elements on the list we can search directly based on their unique attributes.

Example: If we want to search 345 from list we have to calculate its number of digit i.e.3 and value of 1st digit i.e. 3 now send it to that place where it these attributes matches.

Now it searches the particular List if present then it send success message else not found.

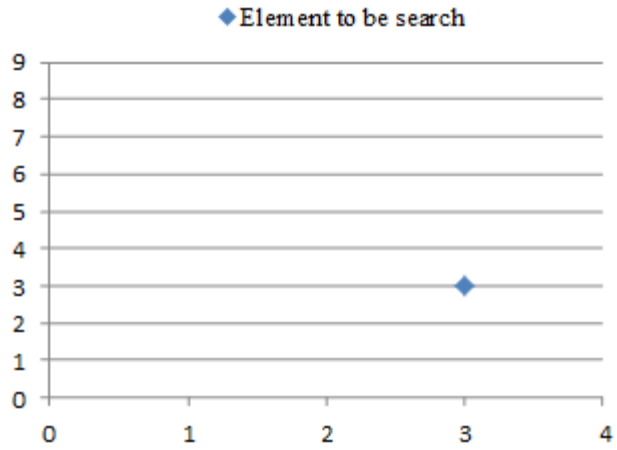

Figure 2: Searching Element from list

Where X-axis is D and Y-axis is F.

V. Time Complexity

The As discussed before that time complexity is depends on the attributes.

Let α be the number of digit of element that we have to search

Let β be the 1st value of that element.

Let ρ be number of element present in a list with respect to α and β, then time complexity will depend on ρ.

Let σ be the number that repeated in a particular List

The worst case would be O(ρ+σ).

For example If we want to search 345 then its α=3 and β=3 and suppose we entered 3 digit number starting with 3 is 40 i.e. ρ=40 then time complexity for worst case would be O(40).

VI. Applications

- Searching of number has many applications like in Data base system using above technique a Data base administrator can easily find out unique id number.
- In Banking system, An account number can easily searchable.
- Based on this searching technique it can be used in Sorting the List.

Acknowledgment

I acknowledge all my college friends and Prof. Nazneen to help me to present this paper.