Optimized Search Functionality with Linear Search Algorithm

Md Emran Hossain*, Syed Bayazid Hossain, Md Sha Alam Tutul, Shamsun Nahar

Department of Computer Science and Engineering, World University of Bangladesh, 17 Uttara, Dhaka 1230, Bangladesh

*Corresponding author
doi: https://doi.org/10.21467/proceedings.123.6

ABSTRACT

Linear search is the essential search algorithm used in fact structures. If is likewise called sequential search. Linear search used to discover a specific detail in an array. It is today no longer obligatory to display an array in any order (Ascending or Descending) as in the case of binary search. Linear search initiates advanced with the support of using sequentially scanning the influences within the array and if the detail has been found, if will display the unique detail and the index value of that detail in that array. In this paper we contemporary a unique searching algorithm of modified Linear Search, which is an improved form of linear search algorithm and its created by likening the influences from each termination. We have additionally in comparison the new Linear search set of rules with Linear Search Algorithm. We used a system for implementation and Analysis of time taken with the aid of using each the algorithms. Result shows that Two Way Searching Algorithm is working glowing for all comments standards and it takes lesser time if the detail to be search is after the middle of the array, otherwise it takes same time as if of linear search. Remaining primarily based totally on the linear search and binary search algorithms; one algorithm designed to act on connected linear list.

Keywords*:* Search, item, linear search, index, array.

1 Introduction

Take steps a search function that searches in an app or a website is a project strategy that offers users a technique to find content. Because search is one of the most important techniques, in which human beings decide pleased online, rating advanced in search engines can high to an increase in circulation of visitors to a website Users can trace comfortable by searching for explicit arguments or expressions, without demanding to understand or route through the structure of the app or website. This can be a faster or easier technique to find content, predominantly on an enormous app or website. In this paper, we introduce optimize search functionalities with a basic searching algorithm which is linear search algorithm. A linear search, also recognized as a sequential search, it is an approach of finding an element within a list. It checks every component of the listing sequentially until a suit is observed or the entire listing has been searched. Though linear search is a basic or simple algorithm, we have applied a method in a book-selling app to search each character within a word and display the articles if any section of character has been harmonized.

Well-organized searching is imperative to get maximum output of the system and eventually that system throughput affords maximum effectiveness to our end user [1]. Consecutive search or linear search is a technique in which essential pattern is matched with each word of entire text chronologically. This method finds all matches if consist in the text [1]. Moreover, there are it perform better than other searching algorithms; binary search and brute force. Linear search or sequential

Proceedings DOI: 10.21467/proceedings.123; Series: AIJR Proceedings; ISSN: 2582-3922; ISBN: 978-81-954993-5-9

search is a technique for discovery a particular value in a list that checks each component in sequence until the preferred element is found or the list is exhausted [2]. The list need not be ordered. Linear search is the simplest search algorithm; it is a special case of brute force search. Its worst-case cost is proportional to the number of elements in the list [2]. Its expected cost is also proportional to the number of elements if all elements are searched correspondingly [2]. Working of this algorithm is simple. Throughout the searching of a component in the assumed array, improved linear search algorithm acquired one variable for including the number of manifestations and one transitory array to store the location, every time search is successful [2]. Tri-Search, which is based on allocating the given components into three unsatisfactory parts [3]. The algorithm component is stored in array in either increasing or non-increasing order [3]. Two Way Linear Search, which improved form of linear search algorithm and is constructed on comparing the components from equally ends [4]. Consequence displays that Two Way Searching Algorithm is operational well for all input values and it takes minor time if the component to be search is after the central of the array, else it takes same time as in case of linear Search [4]. The To-Fro Sequential Search algorithm starts skim through the components from the central of the array. The algorithm shares the array into two halves' i.e. by calculating: (Number of array components) / 2. The subsequent is painstaking as median, in-between array into two halves': left portion and right portion. The algorithm navigates from left portion to right portion and from right portion to left portion until the search component or key component is found. As the traverse, moves left to right and right to left, hence the name To-Fro Sequential Search [5].

2 Analysis

2.1 Data Collections

We implement this search functionality in a book-selling app, where books are stored in google firebase database. Now we have 1000 books details stored in the database. Each book has 10 fields, so that total 10,000 field contains in the database. These fields are title, author, publisher, edition, category, condition, price, uploaded date, unique id.

2.2 Research Strategy

Usually when users search a book from database, then the regular searching algorithm check each character of all stored data and its' field in the database. So that checking or comparing each word taking longer time to find out matched data as the input keyword. But with this search functionality, if user give input, the search technique will count the number of characters of each word in a data in database and compare with the number of characters of input keyword, if the number of characters of input keyword matched with a number of characters of a word, then only the words' character will be compared, otherwise the word will be skip and continue to next word. This checking process will be done in all field i.e.: Title, author, publisher etc.

Proceedings DOI: 10.21467/proceedings.123
ISBN: 978-81-954993-5-9

Series: AIJR Proceedings
ISSN: 2582-3922

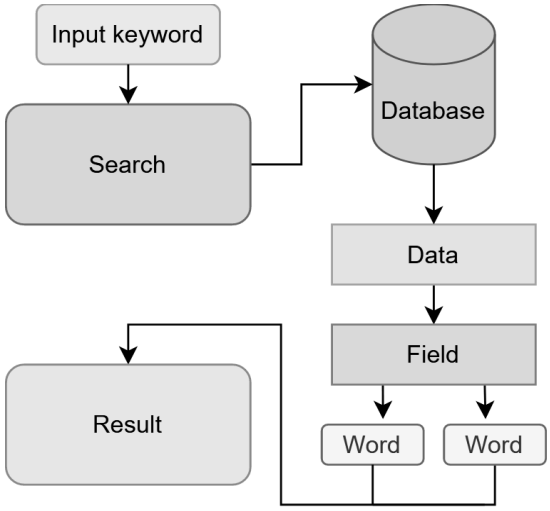

Fig 1: General diagram of the search functionalities.

As shown in the above diagram, when user input a keyword to search a book from database, it will check each data and its' field only if number of characters of a word matched with the number of characters of input keywords.

2.3 Find Keywords

For example, here we take "Computer" as a keyword, so after input this keyword, the function will check how many characters in this word, so here "Computer" has 7 characters. So the algorithm will check a word in the field if the word has 7 characters, otherwise the word will be skip.

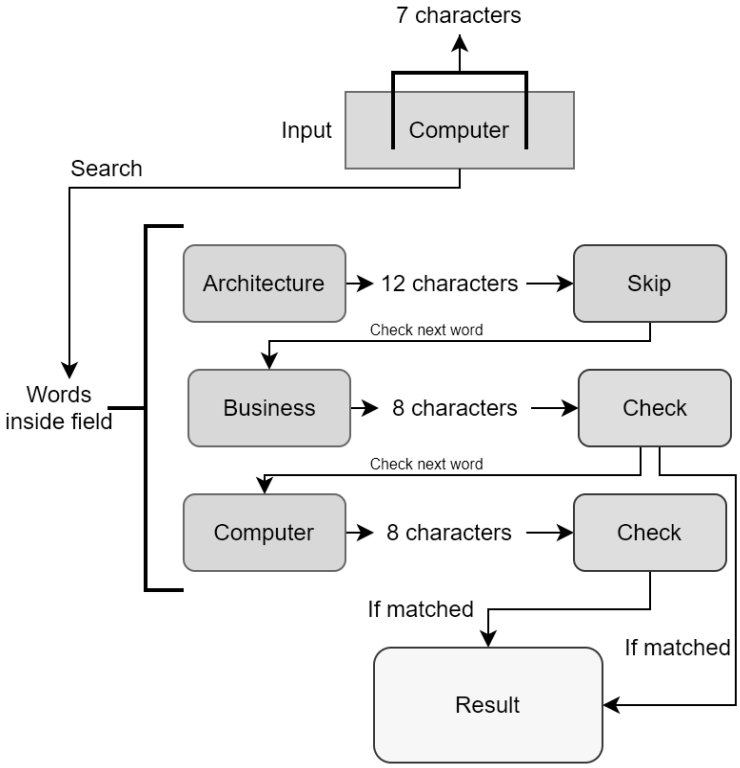

Fig 2: Main diagram of the search functionalities.

So, the first word is “Architecture” which have 12 characters, so as the condition, this word is skip and go next, because of not matching number of characters. Then the second word is “Business”, this have 7 characters, so it meets the condition, so it will be check and it would show in search result If the character matched with the input keyword characters.

Following this the algorithm will check all words in a field in a data. If the number of characters of a word is matched to the number of characters of input keyword, then only the word will be check for matching characters.

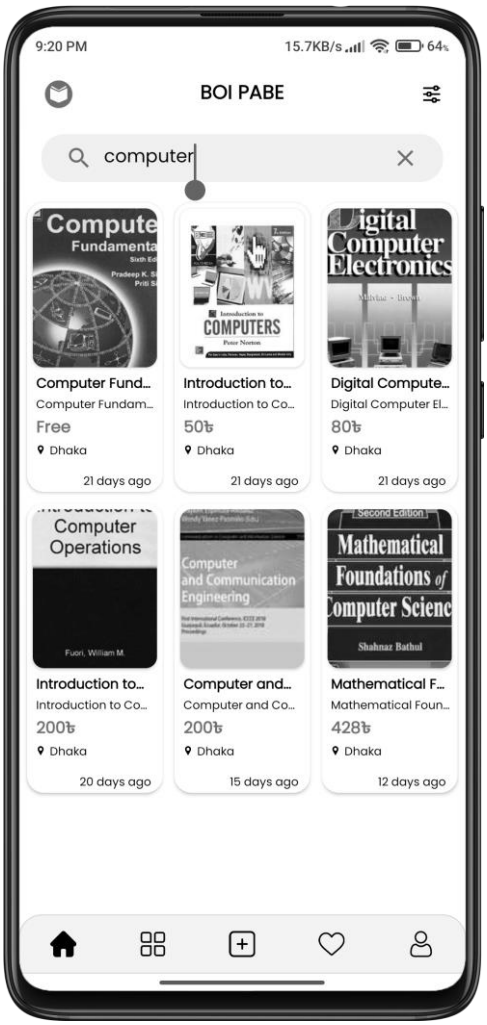

Fig 3: Example of a search result in a system

2.4 Proposed Algorithm

Usually, linear algorithm sequentially checking all characters in an array. We slightly modify the linear search algorithm to improve time complexity. Below we are showing our modified algorithm.

As we compare number of characters, so here *n* is the number of characters. **A** is keyword; **B** is stored word or data from field.

1. Start
2. Check n in **A**
3. Check n in **B**

Proceedings DOI: 10.21467/proceedings.123
ISBN: 978-81-954993-5-9

Series: AIJR Proceedings
ISSN: 2582-3922

4. If *n* of **A** and *n* of **B** are equal, then check each character of the word and show in the search result if matched.
5. If *n* of **A** and *n* of **B** not equal, then skip the word.
6. Forward next word
7. Repeat from 1 to 4
8. End

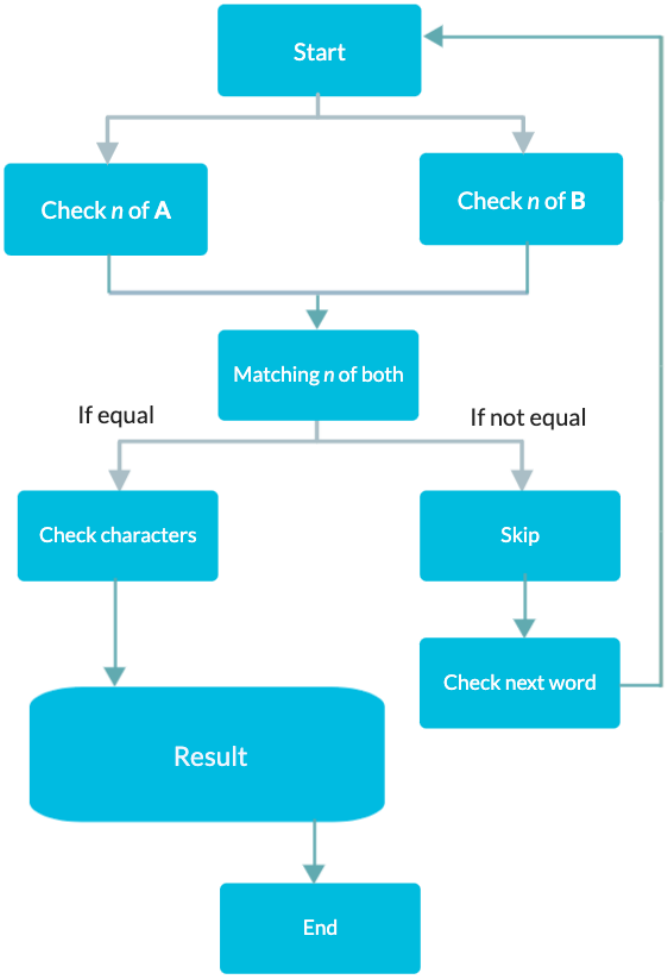

Fig 4: Flowchart of the algorithm

3 Results

We have almost expected search result. Our focus and expectation was reducing time to search a book or content.

3.1 Compare with existing linear search algorithm

According to general linear search, we compare this proposed modified algorithm with the linear search algorithm.

TABLE I: COMPARISON AMONG DIFFERENT APPROACHES WITH PROPOSED ALGORITHM

	Existing Search Algorithm	**Proposed Algorithm**
Time Complexity	0(n)	0(nLog(n))

In this situation we ignore n=1, and our proposed algorithm shows better result than existing one.

	Some Modified Linear Search Algorithm		**Proposed Algorithm (N!=1)**
	Modified 1[8]	**Modified 2**[9]	
Time Complexity	$C_{worst}(N)=N/3$	T (n) = θ (n)	0(nLog(n))

Set	Existing Algorithm	Proposed Algorithm	Modified 1	Modified 2
50	0.3727	0.2936	0.4729	0.4929
100	0.0032	0.0026	0.0077	0.0089
200	0.0051	0.0039	0.0092	0.0103
500	0.0083	0.0071	0.0103	0.0119
1000	0.0147	0.0136	0.0189	0.0213
2000	0.0256	0.0229	0.0298	0.0317
5000	0.0603	0.0501	0.0762	0.0889
7000	0.0830	0.0739	0.0969	0.0987
9000	0.0907	0.0833	0.1087	0.1140
10000	0.1174	0.0998	0.1293	0.1345

Since there are few drawbacks like autosuggestion, entering full keyword as input, so in future we will improve the algorithm according to the autosuggestion in search and partial character of keywords in search. In addition, we will work on this algorithm to improve and more time efficiently.

4 Conclusion

We focused on time complexity in this proposed algorithm. Which is generally slightly modified algorithm of linear search algorithm. We faced many complexities to modify the algorithm and find out the shortest time complexity. People can implement this algorithm to improve user experience to search content. An algorithm called linear time, or O (n) time, if its time complexity is O (n). Casually, this means that the consecutively time increases most linearly with the size of the input. More exactly, this means that there is a constant c such as the maximum running time for each input of size n. there are it perform better than other searching algorithms; binary search and brute force.

References

[1] Muhammad Usman, Zaman Bajwa and Mudassar Afzal "Performance Analysis of Searching Algorithms in C#", *IJRASET*, Volume 2 Issue XII, December 2014, ISSN: 2321-9653

[2] Chandra Andola "Simple Search Technique with enhanced feature", *International Journal on Emerging Technologies*, ISSN No. (Online): 2249-3255

[3] Hriday Kumar Gupta and Rajesh Pandey "Tri-Search: A New and Efficient Searching Algorithm: An Extension of Ternary Search Approach with Variable Partitioning", *International Journal of Computer Applications (0975 – 8887)*, Volume 181 – No. 8, August 2018

[4] Nitin Arora, Garima Bhasin and Neha Sharma "Two way Linear Search Algorithm", *International Journal of Computer Applications (0975 – 8887)*, Volume 107 – No. 21, December 2014

[5] Sumanth.D.R "To-Fro Sequential Search Algorithm – An Alternative to Linear Search", I*nternational Journal of Scientific & Engineering Research* Volume 11, Issue 7, July-2020

[6] Ahmad Shoaib Zia "A Survey on Different Searching Algorithms", *International Research Journal of Engineering and Technology*, Volume: 07 Issue: 01 | Jan 2020

[7] G. B. Balogun "A Modified Linear Search Algorithm", *Afr. J. Comp. & ICT*, Vol. 12, No. 2, June 2019, pp. 43 - 54 ISSN 2006-1781

[8] Subir Saha, Mridul K. Bhaumik and Supratim Das "A New Modified Linear Search Algorithm", International Journal of Mathematics Trends and Technology (IJMTT) – Volume 65 Issue 12 - Dec 2019

[9] Deep Chandra Andola "Simple Search Technique with enhanced feature", nternational Journal on Emerging Technologies (Special Issue NCETST-2017) 8(1): 527-530(2017), ISSN No. (Print): 0975-8364 ISSN No. (Online): 2249-3255

[10] G. B. Balogun "Evaluating the Time Efficiency of the Modified Linear Search Algorithm", Proceedings of the World Congress on Engineering and Computer Science 2017 Vol I WCECS 2017, October 25-27, 2017, San Francisco, USA