iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2 **Issue:** XII **Month of publication:** December 2014

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Performance Analysis of Searching Algorithms in C#

Muhammad Usman[1], Zaman Bajwa[2], Mudassar Afzal[3]
Department of Computing & Technology, IQRA University Islamabad and Pakistan

Abstract--Searching Technique is mostly used in computer sciences and it used as a scale of a system performance and efficiency through implementing different searching algorithm's. In this research paper we have focus on the performance of different searching algorithms such as linear search, Binary search and brute force search which are measured in term of time complexity i.e. execution time of searching algorithm. We have used all above mentioned searching algorithms on different machines and collect the results of time complexity. This varies time to time on different system and size of file. According to results of algorithms, we have found that linear search is better in time complexity and brute force search is best in finding all search patterns.
Keywords-- Binary search, linear search, Brute force, C#, Graphs

I. INTRODUCTION

Searching is the most important task in programming. It's the most methodically studied problem in computer sciences. Searching algorithm is an algorithm that searches a desired element in a huge data. Efficient searching is important to get maximum throughput of the system and ultimately that system throughput provide maximum efficiency to our end user. It is also frequently useful for canonicals statistics and for generating human coherent output. It is the hot area of research because of its importance and complications.

In this paper we implemented various searching algorithms i.e. Binary search, linear search and Brute force search, on Compilers i.e. C family which include C++, C, C# and provided the comparison on the basis of time complexity for the best searching algorithm among the all.

Sequential search or linear search is a method in which required pattern is matched with each and every word of entire text sequentially. This method find all matches if consist in the text.

All sequential algorithms have drawback that they have to walk over the entire text. Improvement work for this problem is to minimize the traversing cost at entire data set. Consider a list in ascending sorted order, in sequential it would start searching at beginning until it reaches at end but it seems more sensible to remove data set as much as possible to find the required pattern quickly. As the list is in order if search starts from the middle and determine which half it should traverse, possibly we could minimize the cost of traversing. By repeating these steps we could find required pattern in quickly manner. This is called binary search.

II. RELATED WORK

We studied different papers for our work. In every paper the binary search is efficient from all other techniques because it divides the list in two parts and determine to which one search but in our work linear search is efficient in time complexity.

A. In this paper a critical analysis is conducted on execution time of linear, binary search without and with inclusion time.
B. The results show that the proposed algorithm can save extra computations, which depends on the size of codebook.
C. In this research paper, we discuss and analyze the performance of different sorting algorithms. The analysis is based on time

In all experiments, there is no best sorting algorithm because in every environment the execution time is change on every execution.

III. EXPERIMENTAL WORKS

For experiment purpose, a program is developed in c# language which includes the algorithms of linear search, binary search and brute force search. The algorithm of linear search is as follow:

```
counter = 0
for i < tokens.Length; i++)
```

```
if (tokens[i].Equals(value))
   counter++;
```

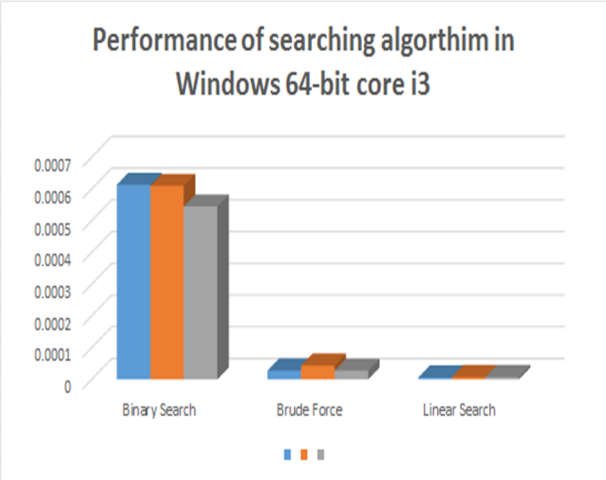
the linear search algorithm is very simple.a for loop is used to travers in the entire datalist . it starts from the begginining and goes till end of sample file.every times if it found a match it will increase counter by one.

Here are the main steps of the binary
search algorithm, expressed in pseudo code:

```
Procedure BINSRCH
// given an array A (1:n) of elements in
nondecreasing order.//
// n ≥ 0, determine if x is present, and if so, set j
such that x = A (j) //
// else j = 0. //
Integer low, high, mid, j, n;
Low ← 1; high ← n
While low ≤ high do
 Mid ← [ (low + high ) / 2]
 Case
 : x < A (mid ) : high ← mid – 1
 : x > A (mid ) : low ← mid + 1
 : else : j ← mid ; return
 Endcase
Repeat
J ← 0
```

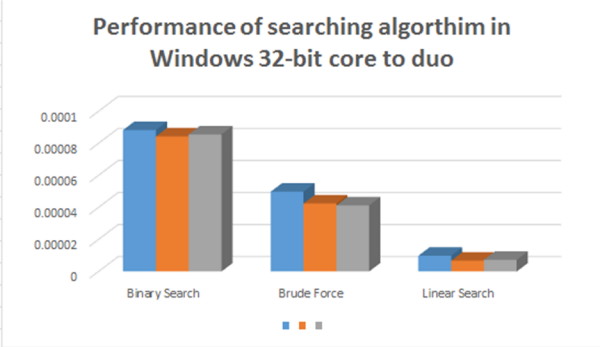
End BINSRCH (Horowitz and Sahni, 1978).

In this paper we perform our experimental work in c sharp compiler which is part of the c family. We took a sample file of 4kb which includes paragraphs of texts and searched for a word router in the entire document on different processing power machines. We did it three times on the machine but every time it gave us new result. The results are showed in graphs below.



In the first experiment we use a machine which has a core i3 processor. In the first run it stated much time in each search because processor was also shared among other process. In second time it took much less time. As per result of processing time, the linear search took lesser time then other two methods. Binary search took comparatively much time in all instances of experiments.

In second experiment we replaced machine with processor of core 2 duo. Result was same in efficiency of algorithms.



IV. CONCLUSION AND FUTURE WORK

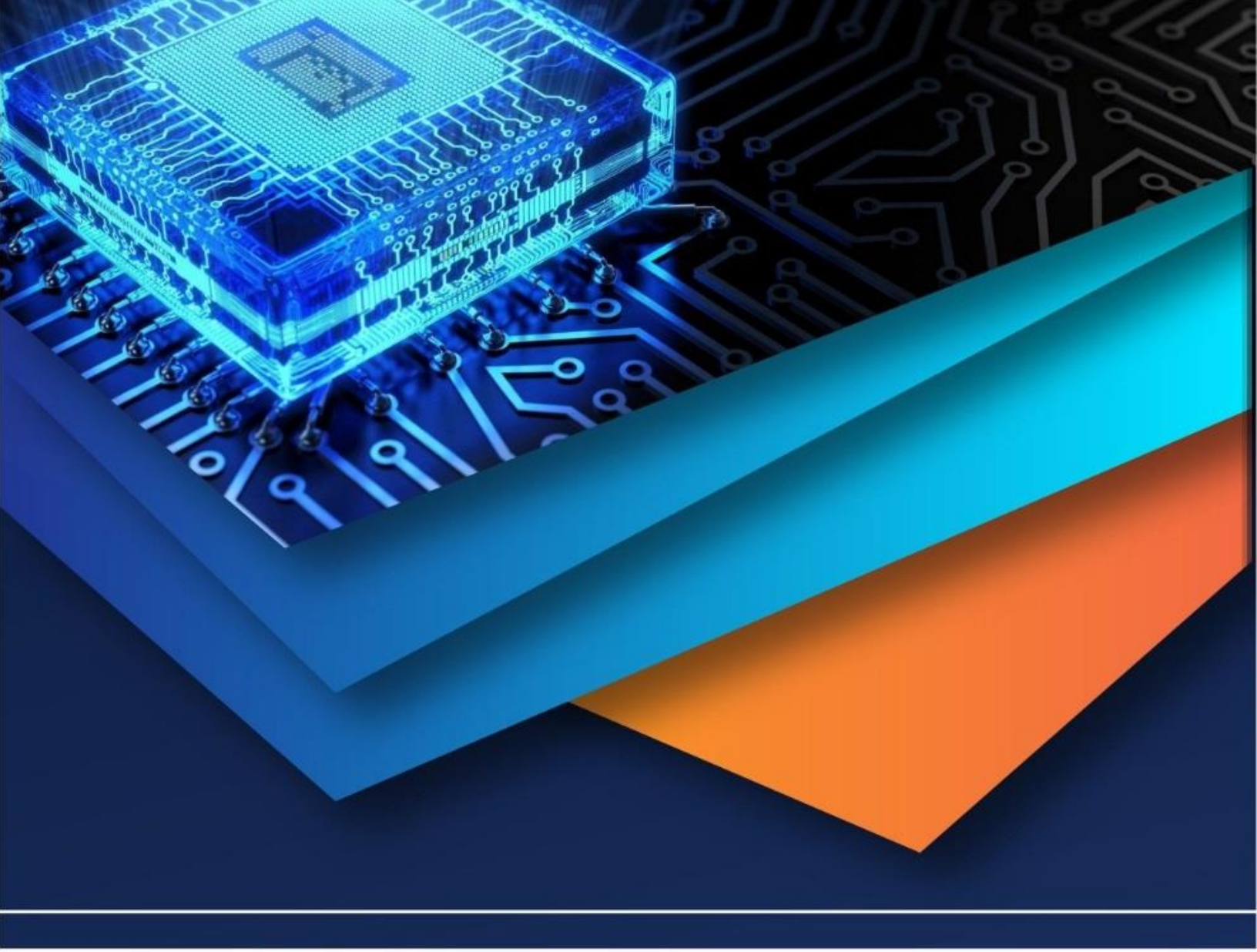
In this research paper, we discuss and analyze the performance of different searching algorithms. The analysis is based on time. In all experiments, the results are different on the same size file. We analyzed the execution time for searching a pattern in a sample text file using linear, binary and brute force.

All existing papers in this field has concluded that binary search is much efficient than others techniques. According to our results the linear search is more efficient as compare to other algorithms because we also considered the sorting time of list which is pre-requisite for the binary search. If sorting time is added to the binary search time then, execution time of

Binary search is not faster than linear search technique. In feature we will implement all these techniques on other compilers .we will see it in term of space complexity too.

REFERENCES

[1] Booch G. (1995), Object Oriented Analysis and Design, second Edition AddisonWesley.
[2] Collins W. J. (1992), Data Structures, first Edition Addison-Wesley publishing company, page 397, U.S.A.
[3] Donald K. (1997), the Act Computer Programming, Volume 3, Sorting and Searching, Third Edition, AddisonWesley.
[4] Horowitz E. and Sahni (1978), Fundamental of Computer Algorithms, Computer science press, Inc, Maryland, U.S.A.
[5] Josuttis N. M. (1999), The C++ Standard Library; a Tutorial and Reference, Addison-Wesley, Reading,M.A.
[6] Kruse R. L. (1999), Data Structures and Program design in C++, Prentice Hall, page 280, USA.
[7] Rodger J. (2007), Searching Arrays: Algorithms and Efficiency linear versus Binary Search, RaiUnversity (http:en.wikipedia.org/wiki/binarysearchalgorithm), last accessed on September 20, 2008.
[8] Stroustrup B. (1994), the Design and Evolution of C++, Addison-wesley Reading M. A.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY