

PAPER • OPEN ACCESS

## Digital Dictionary Using Binary Search Algorithm

To cite this article: Riski Muhamad Fitrian *et al* 2019 *J. Phys.: Conf. Ser.* **1255** 012058

View the [article online](#) for updates and enhancements.



**IOP | ebooks™**

Bringing you innovative digital publishing with leading voices to create your essential collection of books in STEM research.

Start exploring the **collection** - download the first chapter of every title for free.

# Digital Dictionary Using Binary Search Algorithm

**Riski Muhamad Fitrian<sup>1</sup>, Insan Taufik<sup>2</sup>, Muhammad Sabir Ramadhan<sup>3</sup>, Neni Mulyani<sup>4</sup>, Jeperson Hutahaean<sup>5</sup>, Arjon Samuel Sitio<sup>6</sup>, Hengki Tamando Sihotang<sup>7</sup>**

<sup>1,2,6</sup> Teknik Informatika, STMIK Pelita Nusantara Medan, Sumatera Utara, Indonesia

<sup>3,4</sup> Doctoral Program Vocational Technology Education, Postgraduate Engineering Faculty Padang State University Padang West Sumatera, Indonesia

<sup>5</sup> Sistem Informasi, STMIK Royal Kisaran, Sumatera Utara, Indonesia

<sup>6</sup> Doctoral Program, Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan-Indonesia

\* hengki\_tamando@yahoo.com

**Abstract.** Digital Dictionary of Computer and Network Engineering can be used in the learning process of computer and network engineering students. Dictionaries are generally book-shaped is hard to carry because of their thick and heavy, now can be accessed anywhere by development of web technology. Digital dictionaries can be accessed through computers, laptops dan cellphones, this is make student easily in learning process. The digital dictionary can not only be used by students of computer and network engineering, but can also by general public, as long as you have an internet connection, anyone can be use this digital dictionaries. The main process in this digital dictionary is the search process. Binary search algorithm used in the search process of binary search algoritm. Binary search algoritm search is applied to word search in this digital dictionary, because this algorithm is intended for sequential data. Program language that will be used to build this digital dictionary is PHP and MySQL.

## 1. Introduction

Search is a fundamental process in data processing. The search process is to find a certain value (data) in a set of data of the same type (either base type or form type). The search process is an initial activity related to data processing. In the search process the methods used vary, among others linear/sequential search method, binary search method, direct search method, interpolation search method and hash search method. Of the five available methods, the linear search method (linear / sequential search) is the most commonly used method.

The problem that is often encountered in using linear search methods (linear / sequential search) is that if the data sought does not exist, then the search will continue until the last data. This causes linear search to be ineffective applied to large amounts of data. Therefore, the Binary Search method is used to solve that problem.

The advantage of the binary search method (Binary Search) rather than the linear search method is the division of the number of elements. In the linear search method (linear / sequential search), the search is carried out directly within the element, while in the binary search method (Binary Search) the search is done by dividing the element into two parts so that the search can be more effective by dividing the search in two directions, so this method is quite efficient. So that the binary search method is very effective applied to large amounts of data.



## 2. Methodology

### 2.1. Binary Search

There is a search algorithm for the most efficient sorted data, that is binary search algorithm. This algorithm is used to quickly search data. Actually, in everyday life this binary search is often applied. To find the meaning of a particular word in a dictionary (such as an English dictionary), to find the meaning of words, there is no need to open a dictionary from the beginning page to the end of the page one by one, but look for it by separating or dividing it into two parts. If the word you are looking for is not located on the middle page, look again at the left or right side by dividing the two halves. So on until the word is found. This can only be done if the words in the dictionary are sorted. [9]

The principle of searching by dividing data on two parts inspires binary search methods. Data stored in arrays must be sorted. To ease discussion, then the array elements are sorted down (descending). In the search process, two array indices are needed, that are smallest index and the largest index. The smallest index is assumed as the index of the left end of the array and the largest index is the index of the right end of the array. The terms "left" and "right" are expressed by imagining array elements stretched horizontally.

for example the left index is  $i$  and the right index is  $j$ . Initially initialize  $i$  with 1 and  $j$  with  $n$ .

Step 1: Divide into two array elements in the middle element. The middle element is an element with an index  $k = (i + j) \div 2$ . (middle element,  $L[k]$ , divide the array into two parts, that are the left side  $L[i..j]$  and right side  $L[k+1..j]$ )

Step 2: Check if  $L[k] = x$ . If  $L[k] = x$ , search is complete, because  $x$  has been found. But, if  $L[k] \neq x$ , must be determined whether the search will be carried out on the left or right. If  $L[k] < x$ , then the search is done again on the left side array. Otherwise, if  $L[k] > x$ , the search is carried out again in the right-hand array.

Step 3: Repeat to step 1 until  $x$  found or  $i > j$  (that is, the array size is zero).

Example search element with binary search method:

- Illustrate binary search: For example, given an  $L$  array with eight elements that have been sorted down as below:

81	76	21	18	16	13	10	7
$i=1$	2	3	4	5	6	7	$8=j$

- For example the element sought is  $x = 18$ .

Step 1:

Middle element index  $k = (1 + 8) \div 2 = 4$  (shaded element).

81	76	21	18	16	13	10	7
1	2	3	4	5	6	7	8
Left				Right			

Step 2 :

Compare it, if  $L[4] = 18$ ? Ya! ( $x$  found, the search process is complete)

- For example the element sought is  $x = 16$ .

Step 1 :

$i = 1$  and  $j = 8$

Middle element index  $k = (1 + 8) \div 2 = 4$  (shaded element)

16	13	10	7
5	6	7	8
Left'		Right'	

Step 2' :

Compare it: If  $L[6] = 16$ ? No! Must be determined whether the search will be carried out on the left or right side by checking as follow :

Compare : If  $L[6] > 16$  ? No! Do search on the left side array with  $i = 5$  ( fixed ) and  $j = k - 1 = 5$ .

16
5

Step 1'' :

$i = 5$  and  $j = 5$

Middle element index  $k = (5 + 5) \text{ div } 2 = 5$  (shaded element)

16
5

Step 2'' :

$L[5] = 16$  ? Ya! (x found, search proses complete)

### Sample Case :

For example, array K with 5 words that have been sorted as below.

**Table 1.** Sample Case

Index	Word	Definition
1	Access	Activities take or save data from or to memory or to a disk drive.
2	Backbone	Main line (high frequency) communication on computer network systems.
3	Cache	Derived from the word <i>cash</i> , it functions to increase the speed of data transfer both temporarily and permanently.
4	Data	Collection of numbers and characters that have no meaning. Data can be processed to produce information.
5	Edit	Activities modify the format of an output or input by inserting or deleting characters

1. For example the word sought is  $x = \text{"Data"}$ , then:

Step 1 :

Calculate the amount of data in the dictionary database.

$i = 1$

amount  $\rightarrow j = 5$

Step 2 :

Take the middle value of the amount of data in the dictionary database.

middle =  $(i+j) \text{ div } 2$

middle =  $(1+5) \text{ div } 2$

middle = 3

Step 3 :

Check whether position  $K[3] = \text{"Data"}$ ?

No.

Step 4 :

If the position of the searched word is not in the middle position, then :

middle  $\rightarrow i = 3$

amount  $\rightarrow j = 5$

Step 5 :

middle =  $(i+j) \text{ div } 2$

middle =  $(3 + 5) \text{ div } 2$

middle = 4

Step 6 :

Check whether position  $K[4] = \text{"Data"}$ ?

Yes (word found, search stopped).

2. For example the word sought is  $x = \text{"EDP"}$ , then:

Step 1 :

Calculate the amount of data in the dictionary database.

$i = 1$

amount  $\rightarrow j = 5$

Step 2 :

Take the middle value of the amount of data in the dictionary database.

$\text{middle} = (i+j) \text{ div } 2$

$\text{middle} = (1 + 5) \text{ div } 2$

$\text{middle} = 3$

Step 3 :

Check whether position  $K[3] = \text{"EDP"}$  ?

No! Continue search.

Step 4 :

Because  $x$  isn't in position  $K[3]$ , then :

$\text{middle} \rightarrow i = 3$

$j = 5$

Step 5 :

$\text{middle} = (i+j) \text{ div } 2$

$\text{middle} = (3+5) \text{ div } 2$

$\text{middle} = 4$

Step 6 :

Check whether position  $K[4] = \text{"EDP"}$ ?

No! Continue search.

Step 7 :

If  $x$  isn't in middle position  $K[4]$ , then:

Check whether the order of values  $\text{"EDP"}$  greater than  $K[4]$ ? Ya.

Then,  $i = \text{middle} + 1 = 5$

$j = 5$

Step 8 :

$\text{middle} = (i+j) \text{ div } 2$

$\text{middle} = (5+5) \text{ div } 2$

$\text{middle} = 5$

Step 9 :

Check whether position  $K[5] = \text{"EDP"}$ ?

No! (Search stopped, word that search is not in the database).

## 2.2. Search algorithm

Algorithms are at the heart of computer science or informatics. Many branches of computer science are referred to in the terminology of algorithms, such as message routing algorithms in computer networks, Bresenham algorithms for drawing straight lines (fields of computer graphics), Knuth-Morris-Pratt algorithm for finding a pattern in text (fields information retrieval), and so on [9].

According to Donald E. Knuth in his book entitled The Art of Computer Programming [KNU73][6], an algorithm must have five important features :

- An algorithm must always terminate after a finite number of steps.
- Each step of an algorithm must be precisely defined. The action to be carried out must be rigorously and unambiguously specified for each case.
- An algorithm has zero or more inputs, quantities that have given to it initially before the algorithm begins, or dynamically as the algorithm runs.
- An algorithm has one or more outputs. Quantities that have a specified relation to the inputs.

The algorithm will be effective if run by a processor (processor). The processor can be human, computer, robot, machine etc. The processor reads every instruction in the algorithm and then does it. According to Les Goldschlager [GOL88], a processor must :

- Understand each step in the algorithm.
- Work on operations that correspond to these steps.

Search is a fundamental process in data processing. The search process is to find a certain value (data) in a set of data of the same type (either base type or form type). For example, to change (update) certain data, the first step that must be done is to find the existence of data in the collection. If the data sought is found, the data can be changed in value with new data. The same initial activity is also carried out in the process of adding new data. The process of adding data begins by finding out whether the data to be added is already in the collection. If it already exists and considers there is no duplication of data, then the data does not need to be added, but if it doesn't already exist, then add it [9]

### 3. Result and Analysis

#### 3.1. Implementation System

After the system is analyzed and designed in detail, then go to the implementation stage. Implementation is the stage of placing the system so that it is ready to operate. The implementation aims to configure the design into the system. This is so that users can provide input to system developers.

#### 3.2. Digital Dictionary Application Display

Main page



Figure 1. Main Page

Search results if the word exists



Figure 2. Search result if the word exists

Search result if the word doesn't exists



**Figure 3.** Search result if the word doesn't exists

#### 4. Conclusion

From the results of the discussions that have been carried out, it can be concluded as follows:

- The search process on the “Kamus Digital Teknik Komputer dan Jaringan” web-based built using binary search algorithms works well.
- The Binary Search Algorithm successfully finds every word that is searched if the word is available in the dictionary database.
- The basic principle of the binary search algorithm is to repeat the search space repeatedly until data is found or until the search space cannot be shared (data may not exist).
- The purpose of searching using a binary search algorithm is to reduce the number of operations that must be compared between the data sought and data in the database, especially for large amounts of data.
- This digital dictionary is built to facilitate students, especially computer and network engineering students in finding foreign computer vocabulary without having to carry a dictionary in book form everywhere.
- This digital dictionary is also made to make it easier for people to find computer vocabulary that they don't understand.
- This Digital Dictionary can be easily used by users who want to find computer vocabulary.

#### References

- [1] Abdulloh, Rohi. AgusSyahputra (Ed). 2015. *Super Dahsyat Responsive Web Design Dengan Foundation 5*. Cirebon: CV. ASFA SOLUTION.
- [2] A.S, Rossa & Shalahuddin, M. 2013. *Rekayasa Perangkat Lunak*. Bandung: INFORMATIKA
- [3] Binarso, YusiArdi. Eko Adi Sarwoko dan Nurdin Bahtiar. 2012. *Pembangunan SistemInformasi Alumni berbasis Web pada Program Studi Teknik Informatika Universitas Diponegoro*. UniversitasDiponegoro.
- [4] Chadha, Ankit R. Rishikesh Misal and Tanaya Mokashi. 2014, Modified Binary Search Algorithm, International Journal of Applied Information System(IJAIS), 7, 38
- [5] Harison dan Ahmad Syarif. 2016. *Sistem Informasi Geografis Sarana pada Kabupaten Pasaman Barat*. InstitutTeknologi Padang.
- [6] Knuth, Donald E. *The Art Of Comuter Programming vol. 1*. Stanford University: Addison Wesley Longman, Inc.
- [7] Kumar, Parveen. 2013. Quadratic Search: A New and Fast Searching Algoritm (An extension of classical Binary Search Strategy). International Journal of Computer Applications, 65, 43-44
- [8] Munir, Rinaldi. Leony Lidya. 2016. *Algoritma dan Pemograman Dalam Bahasa Pascal, C, dan C++*. Bandung: Informatika.
- [9] Mutiawani,Viska. Juwita& Irvanizam. *Aplikasi Kamus Dwibahasa Aceh-Indonesia Berbasis Java untuk Telepon Genggam*.Universitas Syiah Kuala.
- [10] Pushpa, Suri and Prasad Vinod. 2007, Binary Search Tree Balancing Methods: A Critical Study, IJCSNS International Journal of Computer Science and Network Security, 7, 237
- [11] Roshadi, Aditya Yanuar. 2014. *Perancangan Istilah Teknologi Informasi Berbasis Web MenggunakanMetode Web-QEM*. Universitas NegeriYogyakarta.