

Comparison of Efficiency Data Sorting Algorithms Based on Execution Time

Achmad Fitro

Department of Computer Technology, NSC Polytechnic, Surabaya, Indonesia

ARTICLE INFO

Article History:

Accepted: 15 Feb 2023

Published: 05 March 2023

Publication Issue

Volume 10, Issue 2

March-April-2023

Page Number

15-21

ABSTRACT

In today's era, the development of information technology is increasingly rapid. This is because human life is currently very dependent on the needs of information technology. This can be proven by the number of human interactions with various gadgets, such as laptops, cellphones, computers, and so on. The development of information technology has made IT activists such as companies and programmers compete in making good applications. One of the most basic things that are mastered in making an application is making algorithms. Currently, there are many types of algorithms. One of them is the data sorting algorithm. In this study, we will try to examine 3 data sorting algorithms, namely Insertion Sort, Quick Sort, and Merge Sort. These three algorithms will be used to sort random data ranging from 1000 to 20,000 data. The three algorithms will be compared in terms of execution time. The results show that the Insertion Sort algorithm is a data sorting algorithm that has the fastest execution time compared to other algorithms, while the Merge Sort algorithm is the most time consuming algorithm compared to other algorithms.

Keywords: Data Sorting Algorithm, MATLAB Programming Language, Efficiency, Execution Time

I. INTRODUCTION

Nowadays, human life cannot be separated from information technology. This information technology can be in the form of computers, cellphones and so on. Through this information technology, humans can fulfill household needs, help work in the office, communicate, and so on [1].

The importance of information technology has finally encouraged companies in the field of information technology or programmers to develop applications on these gadgets. The application development that is

carried out certainly cannot be separated from the related knowledge of programming languages. Knowledge of programming languages is very important to make it easier in making an application [2].

There are so many programming languages that are developing today. In web programming, the programming languages that are often used are PHP, Javascript, HTML, ASP, XML, WML, and so on. [3]–[6]. In addition, android programming, the programming languages that are often used are Java, Kotlin, Dart, C#, and so on. Then in application

programming languages, the programming languages that are often used are Visual Basic, Delphi, and Matlab.[7]

Making a program using a programming language must go through the correct procedure. Programming usually always begins with making algorithms. The algorithm is a sequence of steps in solving a problem through a particular program [8], [9]. The word algorithm itself comes from the Arabic "algorism" which means "to calculate with numbers".

In the past, people who were referred to as "algorists" were people who were good at calculating using Arabic numbers. Armed with this term, linguists worked hard to find this word and finally found this word in an Arabic book written by a famous writer, namely Abu Abdullah Muhammad Ibn Musa Al-Khwarizmi. From here the westerners call this word "algorithm" or algorithm.

At first, a programmer must formulate steps to solve the problem to be worked on in the form of an algorithm. Furthermore, the algorithm will be summarized in a simpler form, namely a flowchart. This flowchart is what a programmer will pour into a programming language.

The algorithm itself has various types, one of them is a data sorting algorithm [10]. This algorithm is used to process random data into a sequence using certain steps. In sorting data, this algorithm has 2 types of settings.

These settings include sorting data in ascending (ascending order) and sorting data in descending (descending order) [11]. Sorting data in ascending order means sorting data from the smallest data to the largest data. While sorting the data in descending order means sorting data from the largest data to the smallest data.

There are various types of data sorting algorithms today. Each of these data sorting algorithms has advantages and disadvantages [12]. Each of these advantages and disadvantages is different for each data sorting algorithm. This study tries to compare the 3 most frequently used data sorting algorithms

which consist of Insertion Sort, Merge sort, and Quick Sort. [13],[14]

These three data sorting algorithms will be compared by examining the time taken to complete the data sorting. From this study, it is expected to be able to produce the fastest data sorting algorithm.

II. METHODS AND MATERIAL

As previously described, this study will compare 3 data sorting algorithms, namely Insertion Sort, Merge Sort, and Quick Sort. The initial step is to create a program to generate random data ranging from 1,000 to 20,000 data automatically. Making this program is done using the MATLAB programming language written on the MATLAB 2014 a software as the code editor. The steps used to generate random data can be seen in the flowchart in Figure 1 below.

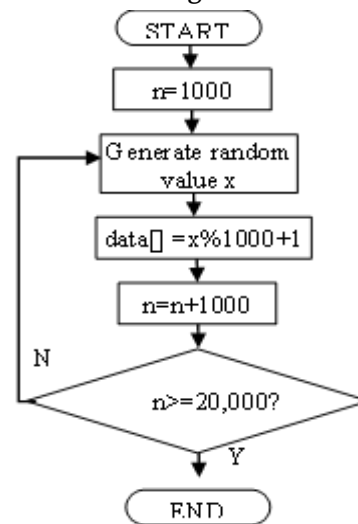


Fig 1. Flow diagram

Or in the form of an algorithm can be described as follows:

- Repeat from n, starting from 1000 to 20,000 in increments of 1000 per step.
- Generate random values between the numbers 0 to n.
- Divide the random value by n and then add 1.
- Store it in a data array variable.
- Do this process n times

The next step is to program the data sorting algorithm using the MATLAB programming language. The program also calculates the execution time needed to sort each data set that has been created. The following 3 types of algorithms will be used in this study.

1) *Insertion Sort Algorithm*

Insertion Sort is one of the data sorting algorithms by comparing all data elements to be sorted. Based on this nature, this algorithm is classified as a comparison based sort algorithm. The way this algorithm works is to find the right location for each data element using a sequential search technique [15], [16].

Next, this process is continued by putting an array element into its proper place. This process is repeated N-1 times.

In summary, the process of sorting data using the Insertion Sort algorithm is to compare the j-th data with the next data, where the value of j starts from the second data to the last data. When the checked data is smaller, then the data will be placed in front where it matches the position it should be [17].

From this explanation, the Insertion Sort data sorting algorithm is obtained as follows:

- The process starts by making the 2nd data element from the data array that has been given a key element. Key elements are data elements that will be used for comparison data in the next process.
- The next process is the key element compared to the previous element where this element is the element in the 1st data.
- If the first element is bigger than the key element, the key element will be inserted before the first element.
- Vice versa, if the first element is smaller than the key element, the key element will be inserted after the first element.
- The next process is to make the third element into a key element. Just like the previous process, this key element will be compared with the data element on the left and will be inserted into the position on the right.

The same checks as the previous process were carried out. This step is carried out until all the data is sorted.

2) *Merge Sort Algorithm*

Merge sort is one type of data sorting algorithm that is created to sort data with a very large amount and does not allow it to be accommodated in computer memory because it is too much. This algorithm was first discovered by John Neumann in 1945.

The basic principle of how the Merge Sort algorithm works is to follow a divide and conquer system. The divide and conquer principle is a method of solving problems that are too big, by dividing the problem into smaller ones, then solving each of the small problems. After that, combining all the problems that have been solved, which in the end becomes the solution of the whole problem [18], [19].

If this principle is applied in the problem of sorting data, then first, the data will be divided into 2 equal parts. For even data, the data will be split into half and half minus 1 for odd data. This data will be broken down again until it consists of only 1 data per block [20].

Then the data is combined again by comparing whether the first data is bigger than the (middle+1) data. If it is true, then the data to-(middle+1) will be inserted into the first data, until the middle is shifted to the 2nd data, up to (middle+1). This is done repeatedly until it becomes a complete data block as in the initial data.

In summary, the Merge Sort algorithm can be described as follows:

- The first stage is divide, dividing all data elements into 2 equal parts.
- The second stage is conquer, which sorts data on each data element that has been divided in the previous stage by calling the Merge Sort procedure recursively
- Combination stage, which combines data elements that are ordered recursively to get a whole series of sequential data

3) Quick Sort Algorithm

Quick Sort is a type of data sorting algorithm which uses a method of splitting the data series into several partitions. Because of this data sorting technique, Quick Sort is also known as partition exchange sort [21].

The first stage of using this algorithm in sorting data is the selection of pivots. The pivot is selected before the start of the data sorting iteration. The selection of this pivot is very important because it will determine the performance of this algorithm whether it is good or bad [14].

This pivot selection method is described as follows:

- The pivot is selected on the first data element, last data element, or middle data element. This pivot selection will be correct if the initial data sequence is random. On the other hand, if the pivot selection is done when the initial data sequence is good, the pivot selection will be less precise.
- The pivot can be chosen randomly from one of the data elements to be sorted. This technique is good but requires a random number generation procedure to select the pivot. So it requires more memory and does not reduce the complexity of the algorithm.
- The pivot can be selected on the middle element of the data. This middle element is achieved by dividing 2 data elements. This technique provides minimum complexity. In fact, finding the middle value on a random data element is not an easy problem.
- The steps for sorting data using the Quick Sort algorithm are as follows.
- The first stage is pivot selection by selecting the first data element, middle data element, last data element.
- After selecting the pivot, split the data into several partitions. In this case, the data element which is smaller than the pivot, will be to the left of the pivot element. Vice versa, if the data

element is bigger than the pivot, then the element will be to the right of the pivot element.

This process will continue until the pivot element finds its last ordering position.

III. RESULTS AND DISCUSSION

As previously explained, in this study, we will compare 3 data sorting algorithms, namely Insertion Sort, Merge Sort, and Quick Sort. Comparisons will be made by testing the three algorithms by solving random data ranging from 1,000 data to 20,000 data with increments per 1000 data. The time required for each algorithm to complete the data sorting process will be recorded for each amount of data. From here we get the results that each algorithm consists of 20 pieces of data.

The results of the execution time testing performed by the Insertion Sort algorithm for various amounts of data, can be seen in Table 1 below

Table 1. Execution time of insertion sort

No	Number of Data (pieces)	Execution Time(s)
1	1,000	0.016
2	2,000	0.050
3	3,000	0.113
4	4,000	0.203
5	5,000	0.309
6	6,000	0.453
7	7,000	0.622
8	8,000	0.807
9	9,000	1.011
10	10,000	1,237
11	11,000	1.497
12	12,000	1970
13	13,000	2,194
14	14,000	2.450
15	15,000	2,800
16	16,000	3.543

17	17,000	4.326
18	18.000	4.333
19	19,000	5.543
20	20,000	6.362

Meanwhile, the execution time testing performed by the Merge Sort algorithm for various amounts of data can be seen in table 2 below.

Table 2. Execution time of merge sort

No	Number of Data (pieces)	Execution Time(s)
1	1,000	0.008
2	2,000	0.004
3	3,000	0.005
4	4,000	0.008
5	5,000	0.010
6	6,000	0.011
7	7,000	0.011
8	8,000	0.017
9	9,000	0.024
10	10,000	0.023
11	11,000	0.023
12	12,000	0.024
13	13,000	0.025
14	14,000	0.025
15	15,000	0.027
16	16,000	0.039
17	17,000	0.049
18	18,000	0.046
19	19,000	0.053
20	20,000	0.056

Table 3. Execution time of quick sort

No	Number of Data (pieces)	Execution Time(s)
1	1,000	0.020
2	2,000	0.030
3	3,000	0.054
4	4,000	0.074
5	5,000	0.097

6	6,000	0.117
7	7,000	0.140
8	8,000	0.164
9	9,000	0.181
10	10,000	0.204
11	11,000	0.222
12	12,000	0.284
13	13,000	0.295
14	14,000	0.326
15	15,000	0.326
16	16,000	0.390
17	17,000	0.441
18	18,000	0.421
19	19,000	0.489
20	20,000	0.599

The execution time test, which is carried out by the Quick Sort algorithm for various amounts of data, can be seen in table 3 above.

After getting the execution time data from each algorithm, the next step is converting the data in a graph. This is used to see which algorithm is more efficient in terms of speed of execution time during the data sorting process. If the three data in tables 1, 2, and 3 above are made in graphical form, it will look like in Figure 1 below.

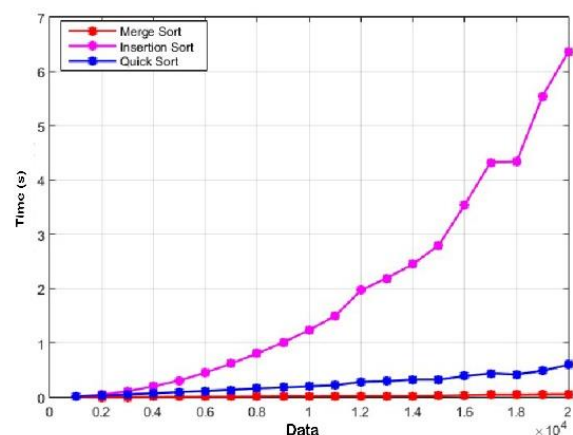


Fig 1. Execution time comparison graph

It is found that the Merge Sort algorithm is the fastest algorithm. On the other hand, the Insertion Sort

algorithm is the most time consuming algorithm compared to other algorithms

IV. CONCLUSION

This study has compared 3 data sorting algorithms; Merge Sort, Quick Sort, and Insertion Sort. These three data sorting algorithms were compared based on the time spent by each algorithm to sort random data from 1,000 to 20,000 data in increments of 1000 data. The results showed that the Insertion Sort algorithm is a data sorting algorithm that has the fastest execution time compared to other algorithms. On the other hand, the Merge Sort algorithm is the most time consuming algorithm compared to other algorithms.

V. REFERENCES

- [1]. Charlie and A. R. Yohanis, "Game Application For Learning Bubble Sort Algorithm," *Kalbiscientia J. Sains dan Teknol.*, vol. 3, no. 2, pp. 24–38, 2016.
- [2]. P. Palvia, J. Ghosh, T. Jacks, and A. Serenko, "Information technology issues and challenges of the globe: the world IT project," *Inf. Manag.*, vol. 58, no. 8, p. 103545, 2021, doi: <https://doi.org/10.1016/j.im.2021.103545>.
- [3]. S. GUPTA, A. KAUSHIK, C. THIEMAN, and M. PADOMEK, "SIGNIFICANT REDUCTION IN CARBAPENEM AND FLUOROQUINOLONE USE WITH IMPLEMENTATION OF MULTIFACETED ANTIBIOTIC STEWARDSHIP PROGRAM (ASP) INTERVENTIONS," *Chest*, vol. 160, no. 4, Supplement, p. A1070, 2021, doi: <https://doi.org/10.1016/j.chest.2021.07.990>.
- [4]. A. F. Rudianto and R. Fauzan, "Application for Calculating Psychological Pressure in the DASS (Depression, Anxiety, and Stress Scale) Scale Using the Certainty Factor Method," *Computer (Long. Beach. Calif.)*, vol. 1, p. 2.
- [5]. Y. Wang, K. S. Cheng, M. Song, and E. Tilevich, "A declarative enhancement of JavaScript programs by leveraging the Java metadata infrastructure," *Sci. Comput. Program.*, vol. 181, 2019, doi: [10.1016/j.scico.2019.05.005](https://doi.org/10.1016/j.scico.2019.05.005).
- [6]. D. Weragama and J. Reye, "Analysing student programs in the PHP intelligent tutoring system," *Int. J. Artif. Intell. Educ.*, vol. 24, no. 2, 2014, doi: [10.1007/s40593-014-0014-z](https://doi.org/10.1007/s40593-014-0014-z).
- [7]. A. Triyadin, I. S.Pd., M.PFis, and Z. Zulkarnain, "THE INFLUENCE OF THE MATLAB PROGRAM-BASED CONTEXTUAL TEACHING AND LEARNING (CTL) LEARNING MODEL ON STUDENTS' LEARNING OUTCOMES IN CLASS VIII MATERIAL OF SMPN 3 NARMADA 2020/2021," *ORBITA J. Kajian, Inov. dan Apl. Pendidik. Fis.*, vol. 6, no. 2, 2020, doi: [10.31764/orbita.v6i2.3364](https://doi.org/10.31764/orbita.v6i2.3364).
- [8]. G. G. Maulana, "LEARNING BASIC ALGORITHMS AND PROGRAMMING USING WEB-BASED EL-GORITHMS," *J. Tek. Mesin*, vol. 6, no. 2, 2017, doi: [10.22441/jtm.v6i2.1183](https://doi.org/10.22441/jtm.v6i2.1183).
- [9]. P. A. Rahayuningsih, "Comparative Analysis of the Complexity of the Sorting Algorithms (Shorting)," *J. Evolusi*, vol. 4, no. 2, 2016.
- [10]. A. A. Jabar and A. S. Anas, "Desktop Based Sorting Algorithm Interactive Learning Application," *JTIM J. Teknol. Inf. dan Multimed.*, vol. 1, no. 1, 2019, doi: [10.35746/jtim.v1i1.10](https://doi.org/10.35746/jtim.v1i1.10).
- [11]. I. Gunawan, S. Sumarno, and H. S. Tambunan, "Use of Sorting Bubble Sort Algorithm for Determining Student Achievement Values," *SISTEMASI*, vol. 8, no. 2, p. 296, May 2019, doi: [10.32520/stmsi.v8i2.493](https://doi.org/10.32520/stmsi.v8i2.493).
- [12]. E. Sunandar and I. Indrianto, "Implementation of the Bubble Sort Algorithm for 2 Variant Models of Data Sorting Using the Java Program Language," *PETIR*, vol. 13, no. 2, 2020, doi: [10.33322/petir.v13i2.1008](https://doi.org/10.33322/petir.v13i2.1008).

- [13]. N. Rathi, "QSort– Dynamic Pivot in Original Quick Sort," *Int. J. Adv. Res. Dev.*, vol. 3, no. 7, 2018.
- [14]. W. T. Saputro, "Quick Sort Algorithm Complexity To Find Time And Memory Efficiency," *INTEK J. Inform. dan Teknol. Inf.*, vol. 1, no. 1, pp. 1–6, 2018.
- [15]. A. S. Mohammed, Ş. E. Amrahov, and F. V. Çelebi, "Bidirectional Conditional Insertion Sort algorithm; An efficient progress on the classical insertion sort," *Futur. Gener. Comput. Syst.*, vol. 71, 2017, doi: 10.1016/j.future.2017.01.034.
- [16]. E. Retnoningsih, "Data Sorting Algorithm with Insertion Sort and Selection Sort Methods," *Inf. Manag. Educ. Prof.*, vol. 3, no. 1, 2018.
- [17]. R. Maulana, "Comparative Analysis of the Complexity of the Sorting Algorithms," *Informatika*, vol. 3, no. September, 2016.
- [18]. J. Lobo and S. Kuwelkar, "Performance Analysis of Merge Sort Algorithms," 2020, doi: 10.1109/ICESC48915.2020.9155623.
- [19]. M. Yusman, A. Aristoteles, and A. R. Irawati, "PARALLEL AND SERIAL COMPUTATION ANALYSIS OF MERGE SORT ALGORITHM," *J. Sains MIPA*, vol. 18, no. 1, 2012.
- [20]. D. Anggreani, A. P. Wibawa, P. Purnawansyah, and H. Herman, "Comparison of Sorting Algorithm Efficiency in Bandwidth Usage," *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 96–103, 2020, doi: 10.33096/ilkom.v12i2.538.96-103.
- [21]. M. R. Maulana, "Comparison of Sorting Algorithms in the Java Programming Language," *IC-Tech*, vol. 12, no. 2, 2017.

Cite this article as :

Achmad Fitro, "Comparison of Efficiency Data Sorting Algorithms Based on Execution Time", *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)*, ISSN : 2456-3307, Volume 9, Issue 2, pp.15-21, March-April-2023. Available at doi : <https://doi.org/10.32628/CSEIT2390151>
Journal URL : <https://ijsrcseit.com/CSEIT2390151>