ISSN 2409-2665
Journal of Logistics, Informatics and Service Science
Vol. 7 (2020) No. 2, pp. 116-126
DOI:10.33168/JLISS.2020.0208

Using Python to Analyze the Execution Time of Education Program: A Perspective of Search Algorithm

Bong-Chul Kim[*], Gi-Pyo Kong, Jong-Hoon Kim

Department of Computer Education, Jeju University, Jeju-do, South Korea

pankun@naver.com[*]

Abstract. In this study, an education program focusing on the effect of search algorithm on execution time using Python was developed and applied to improve the logical thinking of elementary school students. To enhance the effectiveness of the education program, it was developed by ADDIE model. The program was developed based on 133 elementary school students' needs. The education program was conducted for 42 hours for 6 days to verify the effectiveness of the program. The effectiveness of the education program was analyzed by a pre-post examination using the GALT test paper. The results demonstrated that the program improves elementary school students' logical thinking.

Keywords: SW education, Search algorithm, Execution time, Logical thinking, Python.

1. Introduction

Recently, the global economic and social environments have been rapidly changing because of the integration of software in these aspects. Software-oriented society may refer to a society where software is at the center of innovation, growth, and value creation; thereby determining the competitiveness of individuals, companies, and nations (Ministry of Science, 2014). As a reflection of this trend, the direction of education has also been changing. The mainstream of information education has changed from the conventional computer use-oriented education to the software (SW) education. The software education has a high educational value as it can enhance national competitiveness, problem-solving ability, and logical thinking, besides facilitating the understanding of computer information-processing.

Furthermore, considering that the communication method of the 21st century is becoming digitalized, the programming ability has become a basic quality that all social members, including students, should possess (Jung, 2015).

The countries with advanced education systems, such as the UK, USA, and Finland, have already paid attention to the software education, designating it as a compulsory subject and enforcing relevant education (Kim et al. 2015). In South Korea, the software education has also become compulsory in schools since 2018, so that information processing capability can be increased among other capabilities defined in the "2015 Revised Curriculum." In addition, the curriculum was revised for students of grades 5 and 6 in elementary schools to make the complete software education mandatory from 2019 (Ministry of Education, 2015).

Various kinds of educational programming languages are available for the software education. However, it is appropriate to start with languages that are easy to learn and can be applied to various forms of application programs while letting students have a sustained interest (Kim & Kim, 2014). If a text language is learned, as the transferability to a different text programming language is high, it is easy to accumulate basic programming knowledge and structure and use and apply that knowledge in program development. Although it takes a long time to become familiar with a text programming language, according to a recent study (SEO, 2010), it is favorable for complex programming and helpful for improvement of computational thinking as compared to block-type programming language. However, most elementary education sites use educational programming languages for block-based coding, such as Entry and Scratch, because they facilitate easy initial entry stage and have many elements that can be approached with interest.

Therefore, the present study used Python, a text programming language, to improve the logical thinking of elementary school students. Furthermore, after performing preliminary requirement analysis, the textbooks and programs were developed by focusing on the comparison and analysis of the execution time for search algorithms using Python. Among elementary school students of grades 4, 5, and 6, a volunteer sample of 25 students was targeted. Logical thinking was classified by using the following six logical thinking ability factors of the Group Assessment of Logical Thinking (GALT) questionnaire: preservation, ratio, variable control, probability, correlation, and combination.

2. Theoretical Background

A drawback of all text programming languages is that the syntax has to be accurately used in a method of inputting text for commands required in computer programming, and it takes a long time to learn the language. However, as the program-writing ability improves, programming time decreases and various programs can be written with relatively short codes. Furthermore, through debugging, analytical and logical thinking can be improved (Yoo, 2008).

What must be considered first in a learning program, using a text programming language, is a problem of resolving learners' cognitive burden. While learning, it is more important for learners to focus on the process and method of solving a problem rather than on learning the grammar of programming language (Park, E. Y. 2018).

Python. Python is an object-oriented programming language developed by Guido van Rossum in Netherlands in 1991. It is a well-known open-source software in South Korea. Abroad, this programming language is widely used for both practical and learning purposes. For instance, search programs used by Google or Infoseek, as well as Internet service programs of Yahoo, NASA, and YouTube have been developed using Python. Furthermore, as Python supports most operating systems, including Windows, Linux, and Mac, it has good portability and scalability (Park, E. Y. 2018). In addition, the Python programming language is suitable to learn for beginners who are encountering programming for the first time because it is relatively easy to learn and its graphic processing function is simple. Moreover, as it is useful for developing apps or Web applications. Its advantage is that it can be widely used as a programming language for convergent education (Python Software Foundation, 2017).

Logical Thinking. Logical thinking ability has been variably defined by scholars, and there is no consensus among them on defining this concept (So, 2000). Yet, in general, logical thinking is related to inductive/deductive reasoning and can be seen as a broader critical thinking (Fisher, 1987; Kang, 2018). In a narrow sense, logical thinking is a foundation of traditional logic and focuses more on formal logic with respect to argumentation. On the other hand, while based on a formal aspect of thinking, logical thinking in a broad sense refers to a mental ability that also deals with a critical aspect of considering the validity of contents of thought (Gwak, 1985).

In the present study, the concept of logical thinking is defined as a reasoning ability of analyzing and comparing subject matters logically, seeing critically, and thinking based on validity of relationship between the given subject matters.

Search Algorithm. A search algorithm searches the data that satisfy certain criteria or property in the data stored in memory space or a given input data set. Furthermore, it is the most fundamental algorithm in school subjects of data structure and algorithm. While search algorithm is widely used in real programming, it is also considered an easy algorithm for education, because its basic concept is simple and intuitive (Jeoung, 2002; Kim & Kim, 2013).

In the present study, we compare mainly linear search and binary search algorithms among search algorithms to analyze the execution time: the linear search method finds sequentially desired data in a given data set from the beginning, and the binary search method searches a sorted data set by dividing it into half.

Algorithm Execution Time. Algorithm execution time refers to the time taken by the algorithm to solve a certain problem. It is an important criterion for evaluating an algorithm's performance as the efficiency of a computer is determined based on the time it takes even when the same result is derived while solving the same problem. Algorithm execution time is also called time complexity and can be analyzed by measuring the execution frequency of a command statement. In the present study, we focus on algorithm education, through which students ponder on ways to solve a problem in minimal time, and its importance—beyond simply solving the problem (Computer Language Dictionary Compilation Committee, 2005).

Previous Study Analysis. Kim and Yang (2010) conducted several algorithm learnings in real-life cases for elementary school students. The results of their study demonstrated a significant difference in terms of logical thinking ability. Therefore, in the present study, we analyzed the effect of algorithm education on logical thinking ability in the software education using computers.

Park, D. R. (2018) developed and applied a Python learning program using a robot for grade 6 elementary school students. The results showed that the Python learning program using a robot had a positive impact on the computational thinking of students, and that the learning program using a text programming language was significant for elementary school students.

Furthermore, Moon (2018) developed a method of educating with Python to overcome the limitations of block-type programming learning. The author concluded that Python would be a learning tool that could realize ideas and thoughts of students while providing them with a positive motivation. However, a limitation of this study was that the proposed method was not actually applied to students.

Finally, Yoo (2018) claimed that the software education by applying Python could bring diversity into the conventional code block-oriented software education, and that its significance was determined by its future usability. However, elementary school students faced difficulties because of the limitations in the coding method itself and a lack of understanding of the meaning of English words. Therefore, the study concluded with a recommendation to find a remedy measure.

3. Study Method

Study Participants. To investigate the effect of a program developed in this study, an education contribution program was conducted at Jeju University. Twenty-five students were selected for the volunteer sample based on program volunteering sampling. The grade and gender of study participants are summarized in Table 1.

Education Program. We, then, developed an education program according to the ADDIE model, which is one of the models of educational program development (see Table 2). The five stages of the ADDIE model—analysis, design, development, implementation, and evaluation —were performed in a sequence. In accordance with the learners' requirement analysis result, the education program was designed, developed, and implemented in classes; finally, the results were evaluated using the logical thinking test.

Table 1: Grade and gender of the participants

Class	Male	Female	Total
Grade 4	5	0	5
Grade 5	6	2	8
Grade 6	7	5	12
Total	18	7	25

Table 2: Educational program development plan

Analysis	**Analysis of the learners' preliminary requirement (survey targeting upper-grade students)**
Design	Goal set up and tool selection -Effect of education on execution time of search algorithm using Python Evaluation tool design -Logical thinking test (GALT)
Development	Teaching/learning plan (42-hour lessons) Student activity sheet (42-hour lessons)
Implementation	Implementation of software education
Evaluation	Post-event logical thinking test of learners (GALT)

Needs assessment. We used Rossett's Needs Assessment model according to the procedure of the ADDIE model. The Rossett's model, an education needs assessment model used in corporate training, provides easy-to-apply information to people performing the needs assessment. The needs assessment was conducted with 133 grade-6 elementary school students (72 males and 61 females)

Table 3: SW education tools students experienced (N=82)

Tools	Students
EPL (e.g., Scratch, Entry)	52 (63.4%)
Physical computing	19 (23.2%)
Unplugged coding	6 (7.3%)
Computer language	4 (4.9%)

Table 3 shows the results of investigating class tools mainly used in the software education by students with some software education experience (N=82). In most cases, the education was provided using an educational programming language (EPL), such as Scratch or Entry, which was followed by physical computing. On the other hand, the use of computer language had the lowest proportion.

Table 4: Necessity of algorithm execution time training

Necessity	**Students**
Very likely	14 (10.5%)
Somewhat likely	24 (18%)
Neutral	78 (58.6%)
Somewhat unlikely	8 (6%)
Very unlikely	9 (6.8%)

Table 4 shows the necessity of education for algorithm execution time. Most respondents answered “Neutral” (don’t know), and it was assumed that the concept of “algorithm execution time” was somewhat too difficult or unfamiliar to the students. The positive replies (28.5%) had a higher proportion than negative responses (12.8%).

Design and Development of Program. To design a software education program using Python, the learning level of education targets and difficulty level of programming were considered. In particular, by selecting linear search, which is the most basic method among the search algorithms, and binary search, which is easy to understand for even elementary school students, the textbook was composed to facilitate comparative analysis of execution time. The learning themes are presented in Table 5 below.

Table 5. The theme of education program

Hour	**Learning Theme**
1-7	•Orientation •Learning Python basic functions •Input of pre-test
9-14	•Practice with print and input codes •Practice with random code •Practice with arithmetic and condition codes
15-21	•Utilization of the code that has been learned •Practice with list code
22-28	•Understanding linear search •Understanding binary search
29-35	•Creating a project plan for individual projects •Creating individual project objects
36-42	•Presenting individual project work •Input of post-test

Given that many students encountered a text programming language like Python for the first time, the curriculum was composed to intensively learn the basic grammar of Python on the first day. On the second day, random codes, arithmetic codes, and relationship codes that can be used in many different ways were learned. On the third day, the learnt codes were used and practiced. Furthermore, the list that is based on a search algorithm was learned. On the fourth day, among the search algorithms, the linear search and the binary search were compared, and the students examined how these searches operated. The students actually performed coding, analyzed the difference in two search methods, and observed the difference in their execution time. On the fifth day, using an individual project learning method, the learners solved various problems on their own and completed an assignment. On the sixth day, the students made presentations and completed the post-test, thereby finishing the education program.

In particular, even when learning basic grammar, conditional statements, and loop on days 1–3 by focusing on the algorithm execution time analysis, guidance was given to the students to think about the way of coding to reduce the execution time. Furthermore, guidance was given to the students to go through a process of comparing the codes that had different execution times, even though the same result was produced, and to think about the possibility of having a better way. To learn two search programs (linear search and binary search) conducted on the 4th and 5th days, the students were guided to look for the answer on their own regarding which method was more effective in which situation.

Study Design and Treatment. Targeting 25 elementary school students of grades 4--6, we conducted 42 lessons in six days, including orientation, pre-/post-test of logical thinking, and project presentation.

To prove the effect of an education program, pre-test and post-test were conducted by using the GALT, a logical thinking test.

4. Study Results

Verification of Education Program. In the next step, the results were analyzed regarding the effect that Python’s search algorithm-based execution time software (SW) education had on students’ logical thinking.

Normality Test of Logical Thinking Test. Given that the sample size did not meet the criteria of including at least 30 persons, the analysis was performed prior to the verification to check whether the logical thinking test results of experimental group satisfied the normality by performing a normality test. For the normality test, Shapiro-Wilks test, a commonly used test, was performed (see the results in Table 6).

In the results of the normality test, the significance probability (p) of “Preservation,” “Variable control,” “Correlation,” and “Combination” amounted to 0.001, 0.000, 0.000, and 0.001, respectively. Therefore, we had to reject the null hypothesis and the assumption of the data not satisfying the normality test. On the other hand, the significance probability (p) of “Ratio” and “Probability” was 0.167 and 0.056, respectively. Since these values were larger than the significance level, 0.05, the null hypothesis was accepted, and normal distributions were confirmed.

Table 6: Normality test

Subscale	**Descriptive Statistics (N=25)**				***p***
	M	**SD**	**Max**	**Min**	
Preservation	.3600	.8602	3	-1	.001**
Ratio	.5200	1.228	3	-2	.167
Variable control	.1600	.6879	2	-1	.000**
Probability	.0400	1.059	2	-2	.056
Correlation	.1600	.5537	1	-1	.000**
Combination	.4000	.7071	2	-1	.001**

**p<.05 **p<.01*

Pre-test and Post-test of Logical Thinking. To investigate the change in logical thinking in the pre-test and post-test results, the paired t-test was performed for the factors that satisfied the normality test. The Wilcoxon signed-rank test, a non-parametric test, was used to establish the factors that did not satisfy the normality.

Table 7: Changes in logical thinking (paired t-test)

Subscale	**N**	**Pre-test**		**Post-test**		**t**	***p***
		M	**SD**	**M**	**SD**		
Ratio	25	2.56	1.895	3.08	2.178	-2.116	.045*
Probability	25	1.16	.943	1.20	.764	-.189	.852

**p<.05 **p<.01*

In the paired t-test results, the mean score of “Ratio” increased by 0.52 points, from 2.56 to 3.08, and the p-value was 0.045, showing a significant increase in the mean score. On the other hand, in case of “Probability,” the mean increased by 0.04 s, from 1.16 to 1.20, but the *p*-value was 0.854, showing an insignificant increase.

Table 8. Changes in logical thinking (Wilcoxon’s test)

Subscale	**N**	**Pre-test**		**Post-test**		Z	*p*
		M	**SD**	**M**	**SD**		
Preservation	25	2.88	1.054	3.24	.879	-2.000^b	.046*
Variable control	25	1.48	1.418	1.64	1.497	-1.155^b	.248
Correlation	25	.240	.436	.400	.577	-1.414^b	.157
Combination	25	1.20	.707	1.60	.645	-2.500^b	.012*

**p<.05 **p<.01, b. based on the rank of negative*

In the results of the signed-rank test for the factors that did not satisfy the normality in Table 7, the mean score of “Preservation” showed a 0.36-point increase, from 2.88 to 3.24, and the *p*-value of 0.046. The mean score of “Combination” showed a 0.40-point increase, from 1.20 to 1.60, and the *p*-value of 0.012. In other words, the mean score increased significantly for “Preservation” and “Combination,” respectively. On the other hand, the mean score of “Variable control” showed an increase by 0.16 points, from 1.48 to 1.64, while that of “Correlation” increased by 0.16 points, from 0.24 to 0.4; however, these changes did not attain statistical significance ($p \geq 0.05$).

Analysis of Study Results. The paired t-test was performed on the factors that satisfied the normality among the results of the GALT, and the Wilcoxon’s signed-rank test was performed for the logic factors that did not satisfy the normality.

In the results of the paired t-test for two logic factors that satisfied the normality, a significant improvement was found in “Ratio.” In the results of the signed-rank test for four logic factors that did not satisfy the normality, a significant improvement was observed in “Preservation” and “Combination.” In summary, through the search algorithm-based education program, the mean score improved in the cases of “Probability,” “Variable control,” and “Correlation” among the logic types of the GALT, but p-value did not attain a statistical significance. However, a significant improvement was shown in “Ratio,” “Preservation,” and “Combination.”

Taken together, our results demonstrate that the search algorithm-based execution time analysis of SW education using Python can have a positive impact on the improvement of elementary school students’ logical thinking, and the SW education using Python can be effectively used to teach this group of students.

5. Conclusion

To improve the logical thinking for elementary school students in Jeju Province, in the present study, we introduced a search algorithm-based education program using Python and developed and applied this program according to the development stages of the ADDIE model. To easily control external factors while proceeding with the education and increase the reliability of study, an intensive project education containing 42 lessons was conducted during six days by using a holiday period instead of conducting it at weekend classes during a school semester. Furthermore, the assignments related to the content that had been learned and that would be learned in the future were given every day, and the students were supervised to perform the assignments to help them achieve dense learning within a short period. The results of pre-test and post-test of logical thinking confirmed that the SW education using the education program developed in the present study can have a positive effect on the logical thinking of elementary school students.

However, there was a limitation in generalizing the results of the present study as the experimental group did not meet the criteria of including at least 30 participants, which needs to be satisfied for a correlation study. Furthermore, since we focused only on the experimental group (and there was no control group), we could not establish whether the positive effect on the logical thinking was the effect of the education program developed in the present study. Furthermore, because of time constraints, a comparative analysis of execution time was performed only for the linear search and the binary search in the search algorithm education. Additionally, there was a limitation for the students to implement them on their own. In the future research, it would be necessary to replicate these results using an experimental group and a control group—consisting of at least 30 participants or more—and performing a comparative analysis of correlation between factors.

References

Computer Language Dictionary Compilation Committee. (2005). Computer Internet IT Dictionary. Iljin Publishing Co.

Fisher, R.(ed.). (1987). Problem Solving in Primary School. Oxford: Basil Blackwell Ltd. 1-33

Gwak, B. S. (1985). A Study on the Development of CAI Program for Improving Logical Thinking. Seoul: Korea Educational Development Institute.

Jeoung, I. K. (2002). Development of S/W component for search algorithm education. *Journal of The Korean Association of Information Education*, 6(2), 179-186.

Jung, G. W. (2015). An application and analysis of the visual and textual programming language for the Micro Controller Education. *Seoul National University of Education*. Master’s Thesis.

Kang, J. Y. (2018). An analysis of the effect of elementary school software education on Logical Thinking. *Mokpo University of Education*. Master’s Thesis.

Kim, B. C. Kong, G. P. and Kim J. H. (2019). Effect of search algorithm execution time analysis education on logical thinking of elementary school student. *International Journal of Computer Sciences and Information Technologies for Education*, 4(2), 9-16.

Kim, J. D. and Yang, G. W. (2010). The effect of algorithm learning in real life case on logical thinking ability. *Journal of The Korean Association of Information Education*, 14(4), 555-560.

Kim, J. H. and Kim, J. J. (2013). An Introduction to Computers. Seoul: Hanbit publishing co.

Kim, J. H., Kim, Y. M. and Seo, Y. H. (2015). Flowchart & C Language. Seoul: Daol publishing co.

Kim, K. M. and Kim, H. S. (2014). A case study on necessity of computer programming for interdisciplinary education, *Journal of Digital Convergence*, 12(11), 339-348.

Ministry of Education. (2015). Software Education Guideline. http://www.moe.go.kr

Ministry of Science. (2014). ICT and Future Planning. http://www.msip.go.kr

Moon, M. Y. (2018). A study on the teaching methods of Python Programming Language for elementary school students. *The Journal of Korean Association of Computer Education*, 9(1), 33-41.

Park, D. R. (2018). Developing a robotics learning program for elementary school students. *Daegu National University of Education*. Master's Thesis.

Park, E. Y. (2018). Jump to Python 2nd edition. http://wikidocs.net/mybook/read/page?pageid=4

Python Software Foundation. (2017). Python about. https://www.python.org/about/

Seo, S. W. (2010). The effect of computational thinking ability using TPL and VPL with robot programming education. *Korea National University of Education*. Master's Thesis.

So H. R. (2000). Logic and Thought. Seoul: Ewha Womans University Publishing Co.

Yoo, I. H. (2018). The Design of SW education for elementary school using Python and Robots. *Korean Association of Information Education*, 149-155

Yoo, J. A. (2008). A Study on the education of programing language by using the Open Source Software Python. *Dankook University of Education*. Master's Thesis.