Information Engineering Express
International Institute of Applied Informatics
2022, Vol. 8, No. 1, IEE658

Quantitative Measurement and Analysis to Computational Thinking for Elementary Schools in Japan

Rina Yano *, Hiroki Tanioka †, Kenji Matsuura †, Masahiko Sano †, Tetsushi Ueta †

Abstract

In Japan, programming education has been made compulsory in elementary schools since 2020. The Programming Education Guide (GPE) explains the purpose of programming education and the abilities that can be fostered through programming education. In addition, the "Portal Site for Programming Education Focusing on Elementary Schools" introduces various examples of programming education. However, there is little information measuring whether programming classes are effective in improving OTWP (Objective Thinking as a Way of Programming) abilities based on CT (Computational Thinking), except for reports of improvement after simple statistical analysis. Therefore, we prepared 30 CT questions, 12 basic and 18 applied, for the CT test considering four key techniques, decomposition, pattern recognition, abstraction, and algorithms, of which 14 questions were pre-test and seven questions were assessment test. In the experiment, 18 elementary school students from grades 1st to 6th were given a short workshop only once, and the analysis of the effect was done statistically, considering their habituation to the problems. The results of the experiment showed that there was no effect of the one-time workshop, unlike other reports of improvement that used simple statistical methods. It became clear that the CT ability was not improved by the short education. On the other hand, a new finding is that females may be inferior to males in three techniques: decomposition, algorithm, and abstraction.

Keywords: Programming education, Computational thinking, Learning analytics

1 Introduction

1.1 Background

In Japan, programming education at elementary schools has been made compulsory since FY2020. The Ministry of Education, Culture, Sports, Science, and Technology (MEXT) has issued a Guide to Programming Education (GPE) [1]. GPE describes the purposes of programming education and abilities to be nurtured by programming education. MEXT also organized the

* Division of Science and Technology, Graduate Schools of Sciences and Technology for Innovation, Tokushima University, Tokushima, Japan
† Center for Administration of Information Technology in Tokushima University, Tokushima, Japan

Future Learning Consortium with the Ministry of Internal Affairs and Communications and the Ministry of Economy, Trade, and Industry and published a website, Programming Education Portal Centered on Elementary Schools (Education Portal) [2]. This website introduces 56 cases, including trial classes and local events for programming education conducted at elementary schools in Japan (as of February 2020). Of the 56 cases, 23 described in the GPE did not consider programming ability and 31 cases show the evaluation of growth in ability based on subjective surveys of students and mentors. In the other two cases, teachers evaluated the children's programming skills and growth of self-competence and planning. According to the GPE, *"Thinking, decision-making, and expressive skills are not something that can be acquired in a short time or developed rapidly. And Objective Thinking as a Way of Programming (OTWP) is not something that can be fostered solely through programming activities."* [Translated from Japanese.] Hence, none of the 56 cases described in the GPE considered OTWP. However, it is important to be able to measure OTWP in every lesson.

1.2 Thinking as a way of programming

In GPE, OTWP is stated as the logical thinking ability to realize the action intended by the students themselves. For example, logical thinking focuses on what types of action combinations are needed to realize a series of intended behavior, how to compose symbols corresponding to each action, and how to optimize the combination of symbols to become closer to the intended behavior. OTWP was first defined at an expert meeting on programming education at MEXT in 2016. In the summary of the expert meeting [3], OTWP is a concept proposed based on the concept of Computational Thinking (CT) while organizing the relationship between programming and logical thinking, and is described as a definition of thinking as a way of programming. Therefore, CT and OTWP are closely related. CT is a thinking method adopted in the computer science curriculum in the United Kingdom and other countries globally, widely known by Jeannette M. Wing's column [4]. In the UK, according to a BBC article [5], it is classified into four key techniques, decomposition, pattern recognition, abstraction, and algorithms.

1.3 Previous Research

Various approaches have tried measuring the programming and logical thinking abilities of elementary school students. For example, questionnaires, tests, and deliverables in the programming workshops were subject to evaluation.

1.3.1 Evaluation using questionnaire

The questionnaire is a widely used method for measuring students' abilities, developed by programming education in Japan today. Wada [6] conducted a workshop using robot kits and Viscuit [7] for elementary school students. He also measured students' critical thinking using questionnaires. This questionnaire was developed by Kusumi et al. [8]. From the results, Wada concludes that the workshop enhanced students' thinking ability. In addition, Kalelioğlu [9] conducted a study using the Reflective Thinking Skill Scale Towards Problem Solving (RTSSTPS) to quantitatively evaluate the experiment with code.org [10]. In this study, RTSSTPS is developed by Kizilkaya and Askar ［11］ was used for finding out primary education school students' reflective thinking skills towards problem-solving. Mannila et al. [12] have created a tool called PESS (Programming Experience, Self-efficacy and Skills) that focuses on three areas: experience, self-efficacy, and skills. The PESS consists of a 20-question questionnaire about experience, a 22-question questionnaire about self-efficacy, and a 4-question test about skills. The skills test was adapted from or inspired by the CT Test by Román-González [13]. However, when evaluating using a questionnaire, we can use only subjective information as a basis for evaluating students' ability. To perform an accurate evaluation, we also require a method based on objective information.

1.3.2 Evaluation using deliverables in the programming workshop

A method exists to measure students' programming ability based on the deliverables created by the students. Ota et al. [14] quantitatively analyzed programs created by elementary school students to clarify the stages of acquiring programming ability. The analysis target was programs created by 4th–6th grade students in Japanese elementary schools using Scratch [15], one of the visual programming languages. For the evaluation, they used the evaluation criteria developed by their concept of CT. The results of the analysis showed the difference in students' programming ability between the 4th and 6th graders and between the 5th and 6th graders. In addition, Price and Price-Mohr [16] developed an animation engine called "Story-Writing-Coding" and analyzed the code created by children between the ages 7–11 using the engine. Based on the number of lines of code they created and their behavior in trying to fix errors, they examined the way children think when programming. We can use the measurement of students' abilities based on deliverables for objective and quantitative evaluation. However, if the artifacts are based on those created by others, or created by multiple students, it is difficult to clearly identify the contribution of the students. In addition, evaluation criteria for specific programming languages, such as Scratch, are difficult to use commonly for other languages, such as Viscuit, Python, and JavaScript. In Japan, the tools used in programming education are not regulated. Thus, general-purpose evaluation criteria are required, regardless of language or tool.

1.3.3 Evaluation using tests

A method exists using scores of tests that are performed before and after the classes or workshops and measuring the ability developed. Saito et al. [17] conducted a programming course for 4th–6th graders and performed questionnaires, rubrics, and tests before and after (pre-test / assessment test). He found differences in the results and considered that students' ability of programming was improved. Ota et al. [18] introduced that in the UK, CAS (Computing At School), which was commissioned by the Ministry of Education, has published QuickStart Computing [19]. In it, the criteria for evaluating the performance of a new subject, Computer, are clearly stated. One of the evaluation criteria proposed is to have students solve Beaver Challenge [20] problems as a test. However, even if there is a significant difference between the pre-test and assessment test results, it is unclear whether the cause is an improvement in student ability or familiarity with the test. Therefore, students attending the workshop should be divided into experimental and control groups for comparison.

1.3.4 Evaluation using tests before and after workshops

Yano et al. [21] proposed a new evaluation method using tests while ensuring the fairness of education. In this method, subjects are first randomly divided into two groups, Group A and Group B. Group A students take a test before the workshop, and Group B students take the same test after the workshop. The scores of groups A and B are compared for measuring their ability growth. The effect of the workshop was not seen, and the presence or absence of experience was affected. However, bias might occur when grouping; therefore, if Group

A students have a higher ability in CT than Group B, Group A students could obtain higher scores in the test than Group B, regardless of the effectiveness of the workshop. Thus, students are divided into groups without bias about their abilities.

2 Methods

2.1 Proposed Model

Figure 1 shows an experimental model of the proposed method. A pre-test to classify the groups was added to the previously proposed method. The content of the added pre-test is a test to quantify CT, which is closely related to OTWP and should prevent any bias in the average ability of the subjects between groups A and B and accurately quantify the effects of the workshop.

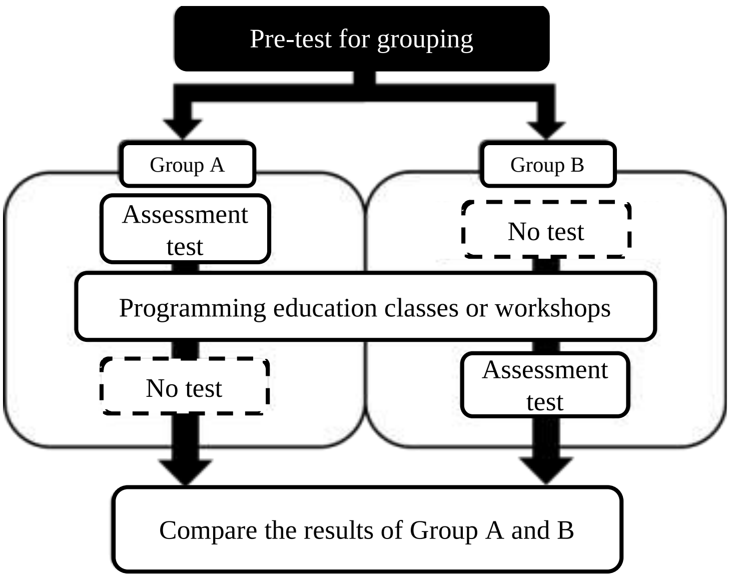

Figure 1: Proposed method to assess improvement in student ability

2.2 Computational Thinking Test

First, we prepared 30 CT questions, including 12 basic questions and 18 applied questions for the CT test considering four key techniques with reference to Kuno [22]. We conducted an online preliminary test using these 30 questions to 83 university students or above (programming experienced: 56, inexperienced: 27), see Tanioka et al. [23] for details. The results of the preliminary test showed three applied questions with significant differences ($p < 0.05$), and seven applied questions with significant differences ($p < 0.1$) using χ^2 tests in the mean between two groups with or without programming experience. To classify the abilities of elementary school students as well as university students and above, two of the seven applied questions are included for the pre-test and all seven for the assessment test.

2.2.1 Pre-test

The pre-test is conducted to group students in a well-balanced manner in CT ability. In the pre-test, we adopted 14 questions consisting of 12 basic questions (Questions 1–12) and two applied questions (Questions 13–14). These questions measure CT. Table 1 shows the correspondence between the 14 questions and four CT techniques. Then, Figure 2–7 show 14 questions of the pre-test.

Table 1: 14 questions of pre-test and four techniques of computational thinking

	Decomposition	Pattern recognition	Algorithms	Abstraction
Question 1–3	✓			
Question 4–6		✓		
Question 7–9			✓	
Question 10–12				✓
Question 13	✓		✓	✓
Question 14		✓	✓	✓

(1) Question 1–3: shown in Figure 2 which require the decomposition technique. These questions involve problems of a combination of five blocks to represent a movement that changes as shown in the presented picture. Conversely, they also include the problem of choosing what kind of picture change can be obtained from the presented combination of blocks. To solve these problems, we must break down the movements in the figure into simple movements and combine them.

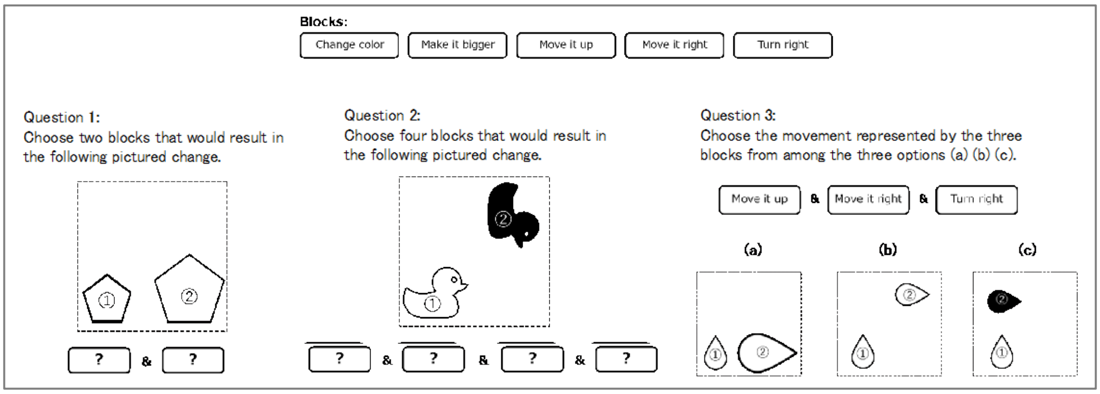

Figure 2: Questions of movement programming using blocks

(2) Question 4–6: shown in Figure 3 which require the pattern recognition technique. These questions involve problems understanding the patterns that the example robot will paint and applying those patterns to other tasks, and will also require an understanding of two-dimensional shapes, logical operations, and mapping concepts. To solve these problems, we must recognize the process as a pattern and apply that pattern to other processes.

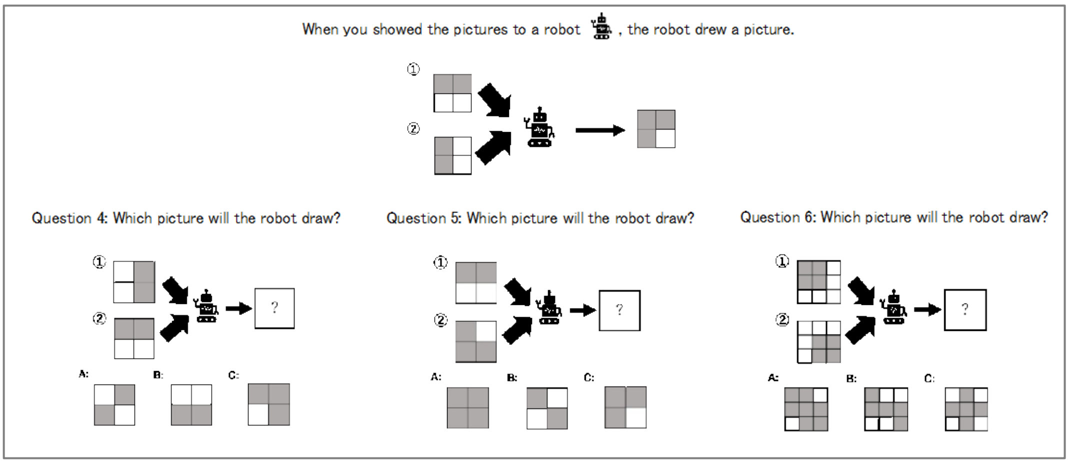

Figure 3: Questions of drawing pattern by robot

(3) Question 7–9: shown in Figure 4 which require the algorithms technique. These questions involve problems controlling the movement of an agent with two types of blocks when the map on which the agent is placed is viewed from above. Spatial cognitive ability is essential for these problems. In addition to this, we need to understand how to implement sequential, conditional, and iterative processes in a flowchart. To solve these problems, we must understand the block rules and build the process.

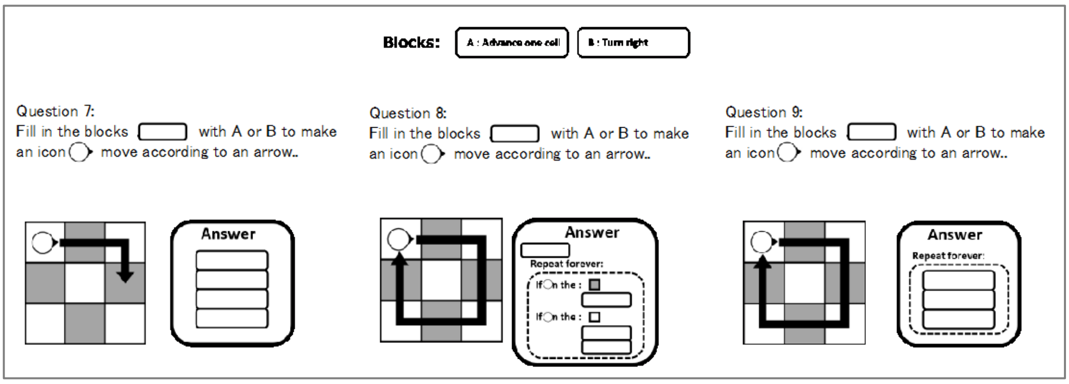

Figure 4: Questions of flowcharting with blocks

(4) Question 10–12: shown in Figure 5 which require the abstraction technique. These questions are inference problems where we must solve the problem based on what the robot says. There are three robots, and their statements are expressed in sentences, so reading comprehension is also required. Each robot has an ability to recognize certain objects and describe them in terms of color, shape, and taste, respectively. To solve these problems, we must find and combine the required conditions among the given conditions.

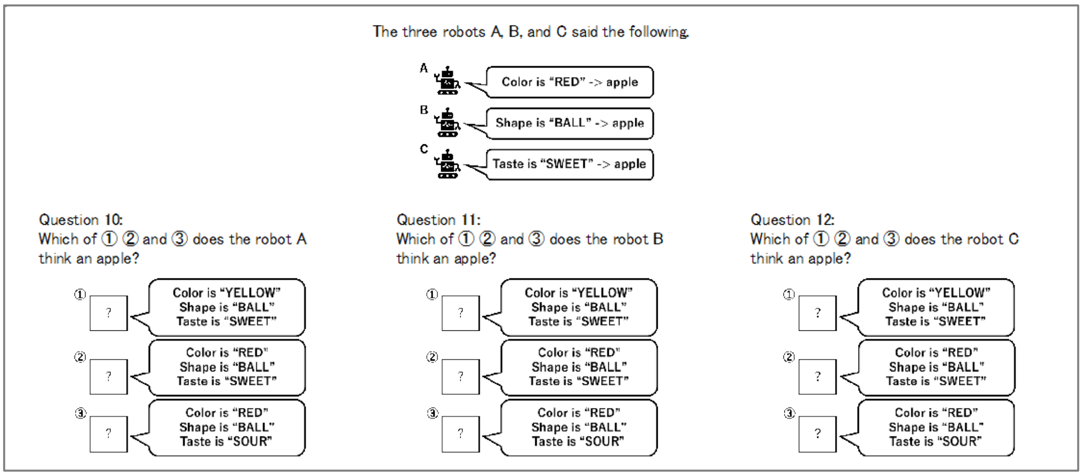

Figure 5: Questions of selecting the sentence matches the decisions by robots

(5) Question 13: shown in Figure 6 which requires multiple CT abilities, decomposition, abstraction, and algorithms. This question is a path finding problem where the path has a transit time condition when searching from the starting point to the end point. While shortest path problems such as the traveling salesman problem are well known as path finding problems, this problem deals with the longest path problem for undirected cyclic graphs. Therefore, it is expected to take more time to find the optimal solution than the directed acyclic graph search problem. To solve this problem, we must find all routes, choose the time-consuming paths, and combine the paths step-by-step. This applied question made a significant difference with or without programming experience in the preliminary test.

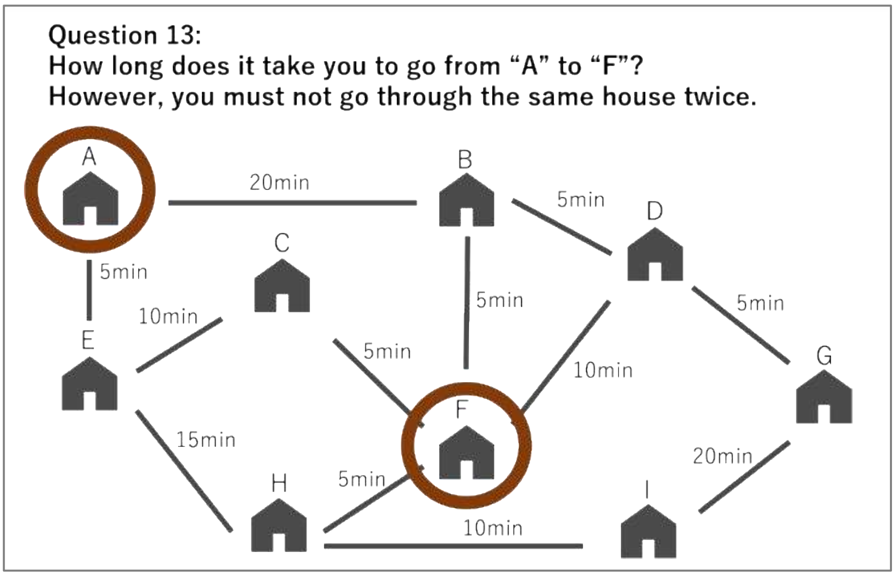

Figure 6: Questions of pathfinding problem

(6) Question 14: shown in Figure 7 which requires multiple CT abilities, algorithms, pattern recognition, and abstraction. This question is a problem based on a wildfire model occurring in a forest. From the state transitions of one block in the example, we are asked to predict the state transitions of nine blocks represented in a two-dimensional diagram. Wood block will become Fire block in the next turn if there is a Fire block next to it, and Fire block will become Burn mark block in the next turn. Burn mark block will not change in the next turn. This problem requires the ability to imagine and predict the future using visual information and patterned wildfire models. To solve this problem, we must understand the process, recognize the patterns, and simulate forest fire while focusing on the important symbols. This applied question made a significant difference with or without programming experience in the preliminary test.

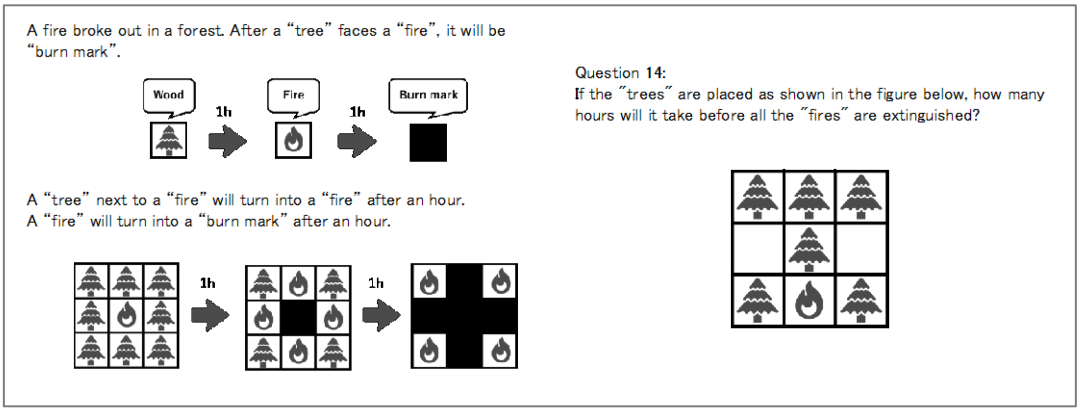

Figure 7: Questions of state transition on the map

2.2.2 *Assessment test*

For an assessment test, seven applied questions are adopted that are significantly different with or without programming experience in the preliminary test. Table 2 shows the correspondence between seven questions and abilities of CT. Then, Figure 8–12 show the seven questions for CT in the assessment test.

Table 2: Seven questions of assessment test and four techniques of computational thinking

	Decomposition	Pattern recognition	Algorithms	Abstraction
Question 1–2	✓			✓
Question 3	✓		✓	✓
Question 4	✓		✓	
Question 5–6		✓	✓	✓
Question 7	✓		✓	✓

(1) Question 1–2: shown in Figure 8 which require decomposition and abstraction techniques. These questions are problems about combinations of orders in a restaurant represented by a tree structure and choosing the best price for an actual order or request. In addition to reading comprehension, students are required to choose the correct answer from a combination of conditions. In real life, in various shopping situations, whether you are in the position of receiving an order or vice versa, this is a necessary ability to avoid unnecessary buying and selling. To solve these problems, we must extract the requirements from the terms of the customer's order, focus on required information, and summarize prices.

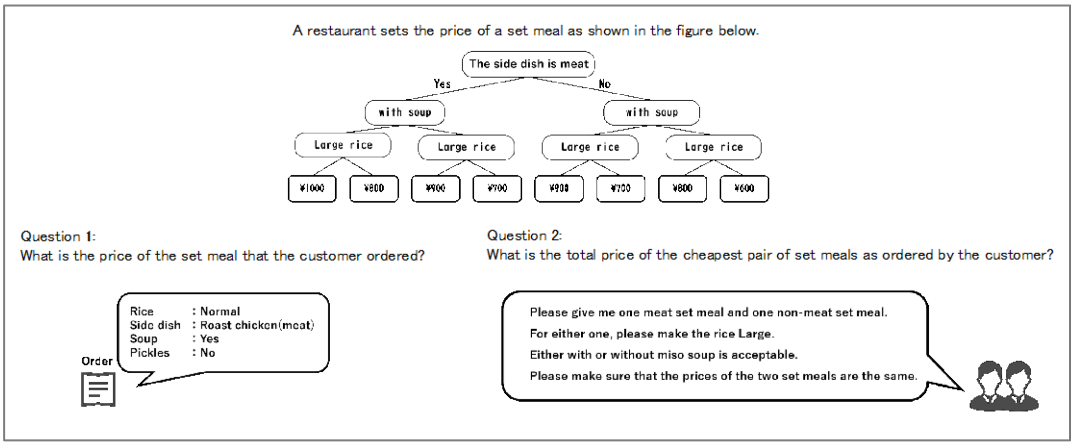

Figure 8: Questions of combinatorial optimization problems in restaurants

(2) Question 3: shown in Figure 9 which is similar to Question 13 of the pre-test to find the route for the most detour. This question requires multiple CT techniques, decomposition, abstraction, and algorithms. This question is a pathfinding problem in which the path has a transit time condition when searching from the starting point to the end point. To solve this problem, we must find all routes, choose the time-consuming paths, and combine the paths step-by-step.

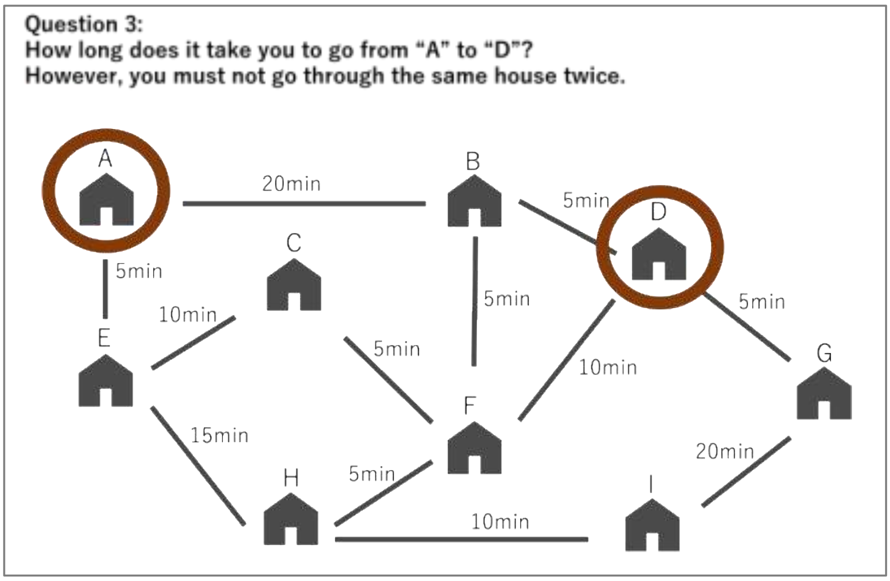

Figure 9: Questions of state transition on the map

(3) Question 4: shown in Figure 10 which requires decomposition and algorithm techniques. This question is a problem which needs to be examined in terms of efficient procedures for baking bread using multiple toasters. The bread needs to be baked front and back, but since it can be swapped in the middle of the baking process, the efficiency can be increased with ingenuity. The ability to solve such problems is also important in real life when considering the efficiency of work. In programming, the ability to use memory efficiently and concurrent processing are also closely related. To solve this problem, we must understand the processes and resources, break down the process into several conditions, and combine them step-by-step to simulate cooking.

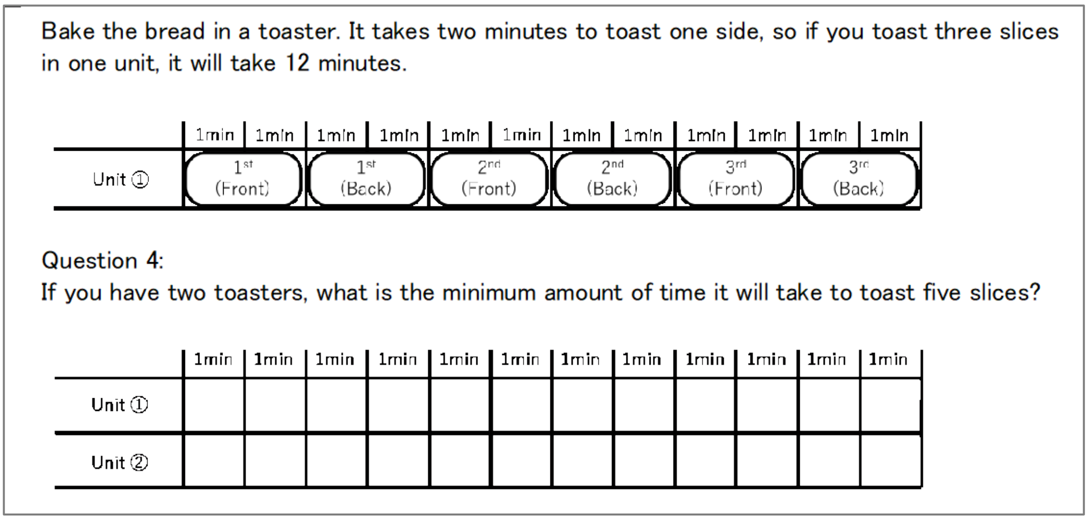

Figure 10: Questions of scheduling algorithm for baking bread

(4) Question 5–6: shown in Figure 11 which are similar to Question 14 of the pre-test. These questions require multiple CT techniques, algorithms, pattern recognition, and abstraction. These problems are based on a wildfire model occurring in a forest. From the state transitions of one block in the example, we are asked to predict the state transitions of nine blocks represented in a two-dimensional diagram. Wood block will become Fire block in the next turn if there is Fire block next to it, and Fire block will become Burn mark block in the next turn. Burn mark block will not change in the next turn. This problem requires the ability to imagine and predict the future using visual information and patterned wildfire models. To solve these problems, we must understand the process, recognize the patterns, and simulate forest fire while focusing on the important symbols.

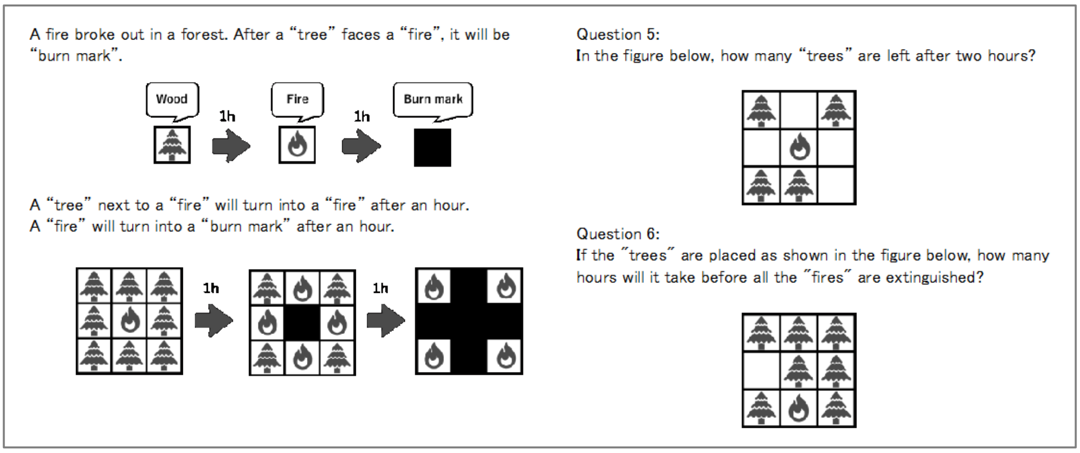

Figure 11: Questions of scheduling algorithm for baking bread

(5) Question 7: shown in Figure 12 which requires algorithms, decomposition, and abstraction. This problem considers a top view of a map of a grid of roads on which cars run from start to finish. When the car makes three different movements at the intersection, turning right, going straight, and turning left, the red, yellow, and blue lights light up, respectively. Spatial cognitive skills are essential for this problem. In addition, since the diagram does not show any information that would indicate the path of the car, we need to infer the car's movement from the history of the lamps. To solve this problem, we must understand the rules and positions of the start and goal, break down the process, and combine them step-by-step to simulate a car driving.

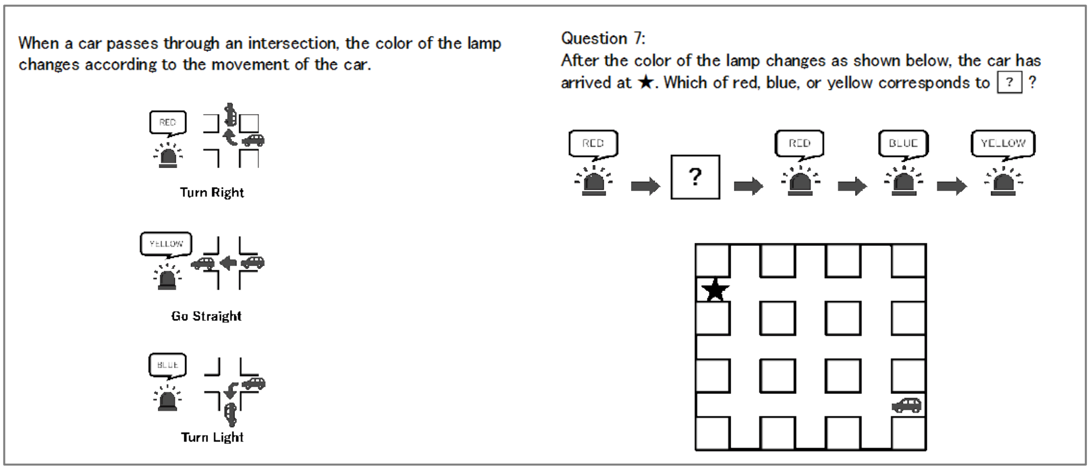

Figure 12: Questions of vehicle path tracking problem on a map

3 Experimental Results

This experiment was conducted with 18 students from the 1st–6th grade of elementary school who participated in the programming workshop [24]. The workshop was a one-and-a-half-hour Viscuit programming online workshop. Beforehand, we classified two groups, Groups A and B, based on the pre-test so that no difference exists in CT ability. The students of Group A took the assessment test before the workshop. The students of Group B took the assessment test immediately after the workshop. The average test time was 30 minutes or less for seven applied questions.

3.1 Pre-test results

A pre-test (maximum score: 14 points) was conducted for the workshop participants, which were divided into two groups so that the pre-test mean and standard deviation balance as much as possible. Table 3 in the appendix shows the assessment test scores of all participants. All the students answered Q1–Q7 and Q11 correctly and there was no difference, while there was variation in Q8, Q10, and Q12–Q14. Table 4 shows the key statistics of the pre-test results. If the mean points of the two groups to compare are the same and the significance level α is 0.05, all the differences are $p \geq 0.05$ using the Mann–Whitney U test, and the null hypothesis is accepted.

3.2 Assessment test results

An assessment test (maximum score: 7 points) was also conducted for the workshop participants. Group A took the assessment test before the workshop and Group B took the assessment test immediately after the workshop. Table 5 in the appendix shows the assessment test scores of all participants. Table 6 shows the key statistics of the pre-test results. For your reference, the mean and standard deviation of the test results of all participants were 4.83 and 1.62, respectively. Furthermore, if the mean points of the two groups to compare are the same and the significance level a is 0.05, the difference between Group, Experienced (Exp), and Grade is $p \geq 0.05$ using the

Table 3: The score list of 18 attendances of the pre-test

Group	Exp-erienced	Grade	Gender	Total Score	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14
A	✓	2	Female	3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
A	✓	2	Male	4	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	
A	✓	3	Male	7	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓
A	✓	4	Male	5	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	
A	✓	5	Male	7	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		
A	✓	5	Male	2	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓
A	✓	6	Female	5	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	
A	✓	6	Male	5	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		✓
A	✓	6	Male	6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
A		2	Male	5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
A		4	Male	4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
B	✓	1	Female	3	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		
B	✓	2	Female	4	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓		
B	✓	4	Female	3	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓		
B	✓	5	Male	6	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓
B	✓	6	Male	6	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		
B		1	Female	3	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓		✓
B		2	Male	6	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	

Table 4: Key statistics of participants scores in the pre-test

	Group		Experienced		Gender		Grade	
	A	B	Yes	No	Female	Male	1–3	4–6
n	11	7	14	4	6	12	8	10
M	11.82	11.14	11.5	11.75	11.33	11.67	11.75	11.4
SD	0.98	0.69	0.94	0.96	1.03	0.89	0.89	0.97
Diff	0.68	$p = 0.127$	–0.25	$p = 0.695$	–0.34	$p = 0.430$	0.35	$p = 0.512$

Mann–Whitney U test and the null hypothesis is accepted, except for Gender. A significant difference exists in the total score of gender in the assessment test, even though there was no significant difference in the pre-test. The female group is 1.75 points behind the male group. Calculated referring to Table 4, the female group had a low correct answer rate in

Table 5: The score list of 18 attendances of the assessment test

Group	Exp-erienced	Grade	Gender	Total Score	Q1	Q2	Q3	Q4	Q5	Q6	Q7
A	✓	2	Female	3	✓		✓		✓		
A	✓	2	Male	4	✓		✓		✓		✓
A	✓	3	Male	7	✓	✓	✓	✓	✓	✓	✓
A	✓	4	Male	5	✓	✓	✓	✓	✓		
A	✓	5	Male	7	✓	✓	✓	✓	✓	✓	✓
A	✓	5	Male	2	✓		✓				
A	✓	6	Female	5	✓	✓	✓		✓		✓
A	✓	6	Male	5	✓	✓	✓		✓		✓
A	✓	6	Male	6	✓	✓	✓	✓	✓	✓	
A		2	Male	5	✓	✓	✓		✓	✓	
A		4	Male	4	✓	✓	✓				✓
B	✓	1	Female	3	✓	✓					✓
B	✓	2	Female	4	✓	✓			✓	✓	
B	✓	4	Female	3	✓	✓			✓		
B	✓	5	Male	6	✓	✓	✓	✓	✓		✓
B	✓	6	Male	6	✓	✓	✓	✓		✓	✓
B		1	Female	3	✓	✓	✓				
B		2	Male	6	✓	✓	✓		✓	✓	✓

Table 6: The key statistics of participants scores in the assessment test

	Group		Experienced		Gender		Grade	
	A	B	Yes	No	Female	Male	1–3	4–6
n	11	7	14	4	6	12	8	10
M	4.82	4.43	4.71	4.5	3.5	5.25	4.38	4.9
SD	1.54	1.51	1.59	1.29	0.84	1.42	1.51	1.52
Diff	0.39	$p = 0.612$	0.21	$p = 0.270$	–1.75	$p = 0.015$	–0.53	$p = 0.415$

Questions 3 and 4. Additionally, Group B had a low correct answer rate in Question 3. A correlation exists between Grade and the correct answer rate of Question 4. When breaking down the number of people for each attribute of the groups, Group A has a lower female rate of 18% (2/11) than Group B with 57% (4/7). Hence, the point difference between the groups in Question 3 could be because of the pro-portion of female students in the group.

4 Discussion

In this section, we discuss the relationship between the percentage of correct answers for each group or attribute and each element of the CT technique which consists of decomposition, pattern recognition, algorithm, and abstraction. More specifically, Question 3 requires three CT techniques, decomposition, algorithm, and abstraction (Table 2). Question 4 requires two CT techniques, decomposition, and algorithm. From Table 13, all 12 male students answered Question 3 correctly, and all six female students were incorrect for Question 4. Therefore, female students were not good at decomposition, algorithm, and abstraction.

The tendency is clear from Table 7. The null hypothesis is that if the mean scores of the two groups in Group, Exp, Gender, and Grade are the same and the significance level α is 0.05, and the Mann-Whitney U test is used to find the difference. For the mean scores of the two groups in Group, Exp, and Grade, the null hypothesis was not rejected. For the mean scores of the two groups of gender in the decomposition, algorithm, and abstraction, the null hypothesis is rejected. From the results of this experiment, the effect of the one-off workshop was not seen as well as in the previous studies. Therefore, the difference in ability by gender is highlighted.

Table 7: The comparison points for four techniques of computational thinking

	Decomposition		Pattern recognition	Algorithm		Abstraction	
Group	0.21 (p = 0.708)		0.18 (p = 0.627)	0.66 (p = 0.379)		0.31 (p = 0.544)	
Exp	–0.07 (p = 0.826)		–0.14 (p = 0.820)	–0.43 (p = 0.587)		–0.21 (p = 0.784)	
Gender	–1.33 (p = 0.015)	**	–0.41 (p = 0.268)	–1.75 (p = 0.008)	**	–0.5 (p = 0.026)	**
Grade	–0.78 (p = 0.118)		0.25 (p = 0.445)	–0.38 (p = 0.524)		–0.15 (p = 0.714)	

**: $p < 0.05$

Furthermore, Figure 13 is a violin chart comparing the differences between the points of the four techniques by Group, Experience, Gender, and Grade. From the top left graph, it can be said that Group B's points were about the same or slightly lower than Group A's points. From this fact, we can say that CT techniques did not improve at the one-off workshop. From the top right graph, programming experience did not make a significant difference for any of the CT techniques. From the bottom left graph, females' points are inferior to males' points in each technique. From the bottom right graph, 4th–6th grade students are slightly better than the 1st–3rd grade students at decomposition.

These results show that CT techniques were not improved by the single programming workshop that have been reported to be effective for improving students' ability [6][11]. The fact that slight differences in decomposition ability were found across grades is consistent with the results of existing studies [14]. Although other study [9] reported that the female group was equal to or better than the male group, it is noteworthy that the male group outperformed the female group in all CT techniques in the present experiment. Additionally, since Question 3 in assessment test is similar to Question 13 in pre-test, and Question 5–6 in assessment test is similar to Question 14 in pre-test, the comparison can be used to determine whether or not there is familiarity with the questions. In fact, the rate of correct answers improved by 0.5 for Question 3 in assessment

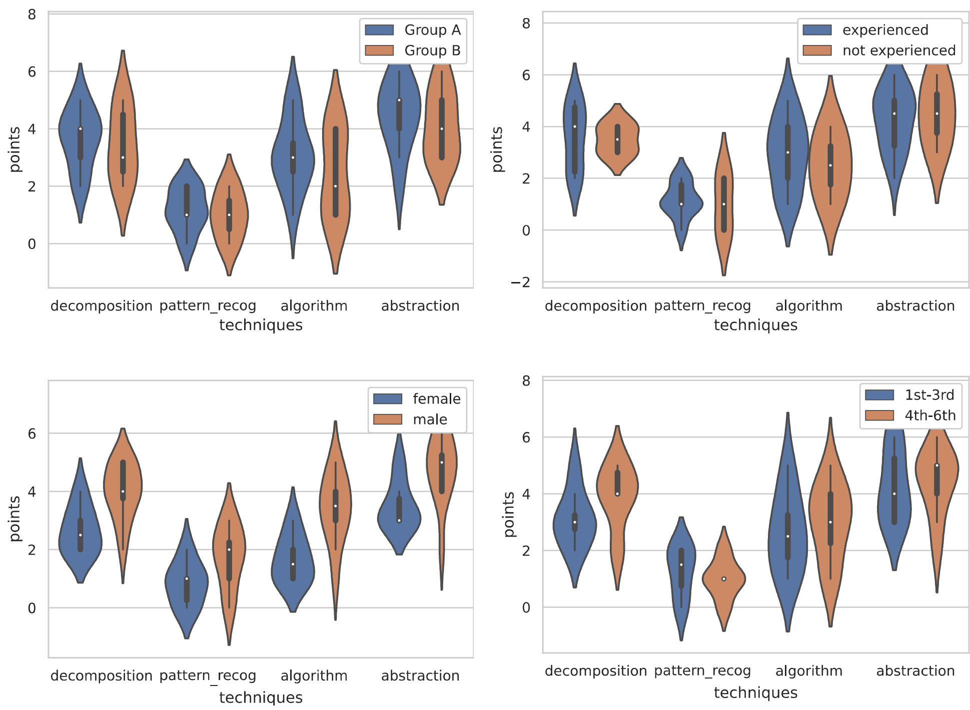

Figure 13: The differences between workshop Group, Experienced, Gender, and Grade for four techniques of CT

test, and the rate of correct answers improved by average 0.167 for Questions 5–6. This indicates the fact that students are accustomed to the same format questions, which was a concern about the experimental results from an experimental method of testing the same group before and after the classes or workshops.

5 Conclusion

This study proposed a method and tests to measure the educational effects of programming education objectively and quantitatively. We used four techniques based on CT related on the OTWP described in GPE and conducted two tests (pre-test and assessment test). We experimented with 18 elementary students from grades 1st to 6th. From the experimental results, the workshop had no effect, it was not enough to conduct a single short workshop except for familiarity with the question format. Hence, in order to develop CT techniques, programming education still requires a long-term perspective. Additionally, although no significant effect occurred from the workshop, programming experience, and grade, the female group was inferior to the male group in three techniques, decomposition, algorithm, abstraction. However, the scale of the experiment is too small to clarify whether this result is a trend that occurred only in this experiment or not. Thus, we will conduct a long-term and large-scale survey, and investigate efforts to clarify and standardize CT competency descriptions.

Acknowledgement

We would like to thank staffs and participants of Tokupon AI School for cooperating experiments. This work was supported by JSPS KAKENHI Grant Number JP18K11572 and JP18H03344. The authors would like to thank Enago (www.enago.jp) for the English language review.

References

[1] "Guide to programming education." MEXT (in Japanese). https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1403162.htm (accessed Nov. 30, 2021).

[2] "Programming Education Portal Centered on Elementary Schools." Future Learning Consortium (in Japanese). https://miraino-manabi.jp (accessed Nov. 30, 2021).

[3] "Meeting of experts on programming education and development of logical thinking ability, creativity, problem-solving ability, etc. at the elementary school stage." MEXT (in Japanese). https://www.mext.go.jp/b_menu/shingi/chousa/shotou/122/index.htm (accessed Nov. 30, 2021).

[4] J. M. Wing, "Computational Thinking," *Commun. of the ACM*, vol. 49, no. 3, pp. 33–35, 2006.

[5] "Introduction to computational thinking." BBC. https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1 (accessed Nov. 30, 2021).

[6] S. Wada, "Development and practice of programming educational materials toward inquiry-based learning -Fostering deep learning-," (in Japanese), *JSSE Res. Rep.*, vol. 33, no. 4, pp. 33–38, 2019.

[7] Digital Pocket LCC. "Viscuit." https://www.viscuit.com/ (accessed Feb. 8, 2021).

[8] T. Kusumi, M. Murase, and A. Takeda, "Measurement of critical thinking attitudes of upper grades and junior high school students," (in Japanese), *JSET J.*, vol. 33, no. 1, pp. 33–44, 2016.

[9] F. Kalelioğlu, "A new way of teaching programming skills to K-12 students: Code. org," *Comput. in Human Behav.*, vol. 52, pp. 200–210, 2015.

[10] "Learn today, build a brighter tomorrow." Code.org. https://code.org/ (accessed Nov. 15, 2021).

[11] G. Kizilkaya and P. Askar, "The development of a reflective thinking skill scale towards problem solving," *Educ. and Sci.*, vol. 34, no. 154, pp. 82–92, 2009.

[12] L. Mannila, F. Heintz, S. Kjällander, and A. Åkerfeldt, "Programming in primary education: towards a research based assessment framework," in *Proc. WiPSCE'20: Workshop in Primary and Secondary Comput. Educ.*, ACM, 2020, pp. 1–10.

[13] M. R.-González, “Computational thinking test: Design guidelines and content validation,” in *Proc. the 7th Int. Conf. on Educ. and New Learn. Technologies (EDULEARN15)*, 2015, pp. 2436–2444.

[14] G. Ota, H. Kato, and Y. Morimoto, “Quantitative Analysis for Acquisition of Children’s Programming Skills: Scratch Programming of Grade 4-6,” (in Japanese), *IPSJ Trans. on Comput. and Educ. (TCE),* vol. 5, no. 3, pp. 35–43, Oct. 2019.

[15] MIT Media Laboratory. “Scratch.” https://scratch.mit.edu/ (accessed Mar. 16, 2021).

[16] C. B. Price and R. M. Price-Mohr, “An evaluation of primary school children coding using a text-based language (Java)," *Comput. in the Schools*, vol. 35, no. 4, pp. 284–301, 2018.

[17] D. Saito, H. Washizaki, Y. Fukazawa, T. Yoshida, I. Kaneko, and H. Kamo, “Learning Effects in Programming Learning Using Python and Raspberry Pi: Case Study with Elementary School Students,” in *Proc. 2019 IEEE Int. Conf. on Eng., Technol. and Educ. (TALE)*, IEEE, 2019, pp. 1–8.

[18] T. Ohta, Y. Morimoto, and H. Kato, “A Survey of Information Education Curricula Including Programming Education in Other Countries: Focusing on the United Kingdom, Australia, and the United States,” (in Japanese), *Jpn. J. of Educational Technol.* vol. 40, no. 3, pp. 197–208, 2016.

[19] M. Dorling, “CAS Computational Thinking - A Guide for teachers, Computing At School.” Accessed: Nov. 15, 2021. [Online]. Available: https://community.computingatschool.org.uk/resources/2324/single

[20] “Beaver Computing Challenge - Mathematics Contests, CEMC.” University of Waterloo. https://cemc.uwaterloo.ca/contests/bcc.html (accessed: Mar. 16, 2021).

[21] R. Yano, H. Tanioka, K. Matsuura, M. Sano, and T. Ueta, “Quantitative Measurement and Analysis to Thinking as a Way of Programming for Elementary School in Japan,” in *Proc. IIAI-AAI2020*, Sep. 2020, pp. 163–168.

[22] Y. Kuno, “Examination Questions Construction Manual for Evaluating Thinking / Judgement / Expression Abilities,” (in Japanese), in *IPS Symp.*, Aug. 2018, no. 1, pp. 1–8.

[23] H. Tanioka and R. Yano, “Development and Evaluation of Quizzes Aimed at Quantifying Computational Thinking,” in *Proc. IIAI-AAI2021*, Jul. 2021, pp. 188–191.

[24] “Tokupon AI School.” Tokushima University (in Japanese). https://www.tokushima-u.ac.jp/ (accessed Nov. 15, 2021).