論文 / 著書情報
Article / Book Information

Title	Cognitive Correlates of Performance in Algorithms in a Computer Science Course for High School
Author	Aimee Theresa S Avancena, AKINORI NISHIHARA
Journal/Book name	AACE Society for Information Technology and Teacher Education 2012, , ,
Issue date	2012, 3
Copyright URL	http://www.aace.org/conf/copyright.htm
Copyright	Copyright (c) 2012 Association for the Advancement of Computing in Education(AACE)

Cognitive Correlates of Performance in Algorithms in a Computer Science Course for High School

Aimee Theresa S. Avanceña
Tokyo Institute of Technology
aimeetheresa@gmail.com

Akinori Nishihara
Tokyo Institute of Technology
aki@cradle.titech.ac.jp

Abstract: Computer science for high school faces many challenging issues. One of these is whether the students possess the appropriate cognitive ability for learning the fundamentals of computer science. High school students who are mostly novice programmers need to learn not only programming but also algorithmic thinking. Thus, there is a need to determine the necessary cognitive factors that can aid in devising strategies for effective learning of computing fundamentals. This ongoing research aims to verify among students of a science and technology high school in Japan the correlations between certain known cognitive factors and actual performance in learning algorithms.

Introduction

Computer science is now being considered as an essential subject along with the traditional STEM (science, technology, engineering and mathematics) (CSTA, 2005). High schools in many countries now offer computer science courses either as a core subject or as an elective. These schools and other academic organizations are also finding ways to improve the computer science curriculum. This is to address the many questions and challenges that computer science education for high school faces. A study concluded that studying computer science early on is fitting for those whose future career is related to this field and is a way to encourage students to pursue computer science related degrees (Sabin, et.al., 2005). This can have repercussions on the current high attrition problem among computer science majors and the poor performance of students in programming (Rafieymehr, 2008).

Related studies in the past include those that tried to determine success factors in computer science and to measure programming aptitude. Such studies aimed to provide a selection process that can assess the survival chances of the students in the field of computer science, basically in the introductory programming class (Avanceña & Nishihara, 2010). Most computer science curricula offer introductory programming at the beginning of the course with the assumption that it usually determines whether a student has enough capacity to succeed in later computing classes. According to the ACM Computing Curricula 2001 Final Report, programming is a prerequisite for many advanced courses in Computer Science so it is necessary that it be mastered early. Performance in introductory CS is also a measure of how the student will fare in the advanced courses (ACM, 2001).

This research focuses on the introductory programming course given to students in a science and technology high school in Japan. Assessment of aptitude in computing in secondary school or K-12 has become an extensive topic in many current researches and projects. Studies of the similar kind but focused on tertiary level computer science were done in the past in order to improve the selection process of incoming Computer Science students, to enhance the Computer Science curriculum, or to better select employees mainly programmers. Such studies which mostly aimed at identifying variables for measuring and predicting programming aptitude have been carried out as early as the 1960s and increased noticeably in the 1980s as programming courses became more popular. Variables such as aptitude and background in Math, gender, programming experience, previous academic performance, behavioral and cognitive aspects, types of personality and learning were considered in those studies (Rountree & Rountree, 2004).

Objectives

The main objective of the study is to determine among novice programmers the correlations between certain known cognitive factors and actual class performance on the topic of algorithms. The goal is to determine which among previously researched cognitive factors can be used to assess the aptitude of students taking up introductory programming and algorithm course in high school.

The participants of this research are students who belong to the Information Systems Course or Class II-B of TokyoTech High School of Science and Technology, a special secondary school of Tokyo Institute of Technology. TokyoTech High follows a science and technology-based curriculum where students choose a specific track at the end of their first grade. The course choices include: Applied Chemistry, Information Systems, Mechanical Systems Engineering, Electrical and Electronics, Architectural Design. This is done to help the students decide and prepare for their major in the university later on.

The right selection of a course either in the secondary or tertiary level is a crucial factor in being successful in it. Even performance in introductory programming and algorithm classes can be addressed properly if there is an assessment process which determines the students aptitude them. As some studies have shown, aptitude in programming has a consequential effect on the performance in a computing course. Though success and retention in the course can only be predicted, an aptitude test can be given to the students to determine if they have an innate ability to easily learn programming and fundamental algorithmic concepts. This research hopes to be an assessment aid for students of TokyoTech High School in choosing a course track and predicting their success in it.

Methods

The study involves the implementation of known cognitive factors tests to determine what cognitive factors are necessary for learning algorithms. The goal is to find out correlations among the cognitive factors and algorithm tests and to check for possible predictors of success in the midterm exam on algorithms. This is to identify which cognitive factors determine the students' ability to learn fundamental algorithms. The research is still in its initial stage and the proposed initial framework is shown in Figure 1. Three known cognitive factors from the Kit of Factor-Referenced Cognitive Tests by the Educational Testing Services (ETS) are chosen for this research. These are Associative Memory, Induction, and Spatial Scanning. Written tests that correspond to these factors were converted to online tests and the instructions were translated to Japanese.

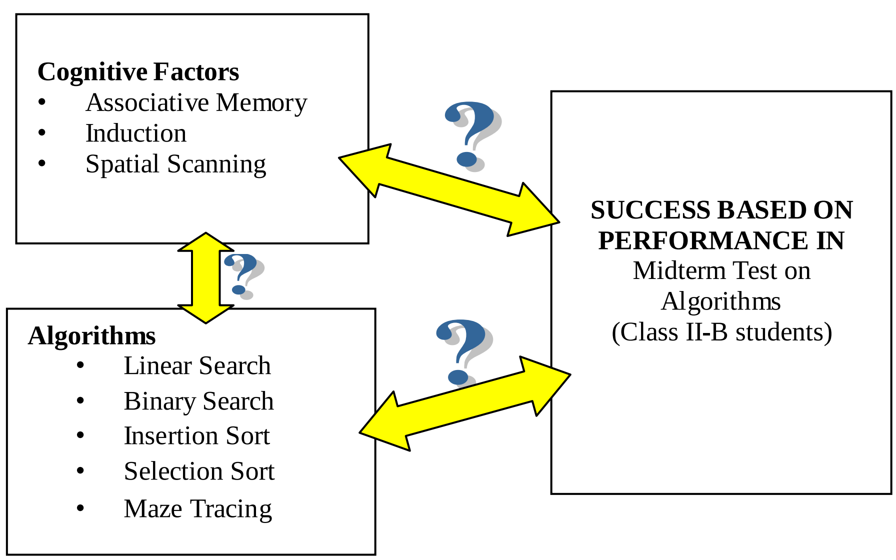

Figure 1. Initial Research Framework

The first cognitive factor considered in this research is Associative Memory and the corresponding test chosen for this factor is the Object-Number Test. This factor signifies the ability to recall one part of a previously learned pair of items even when they are not related by applying memory storage and retrieval. The object number test entails recalling an object and its corresponding number value. The second factor is Induction and the test for this is the Letter Sets Test which is similar to an odd-man-out test. Spatial Scanning is the last cognitive factor considered and the corresponding test for it is Map Planning. The test involves finding the shortest route between two points as quickly as possible. A city map is given which is composed of numbered squares as the buildings, dark lines as streets, and circles as road blocks that indicate impassable roads (Ekstrom, 1976). Screenshots of the tests are shown in Figure 2.

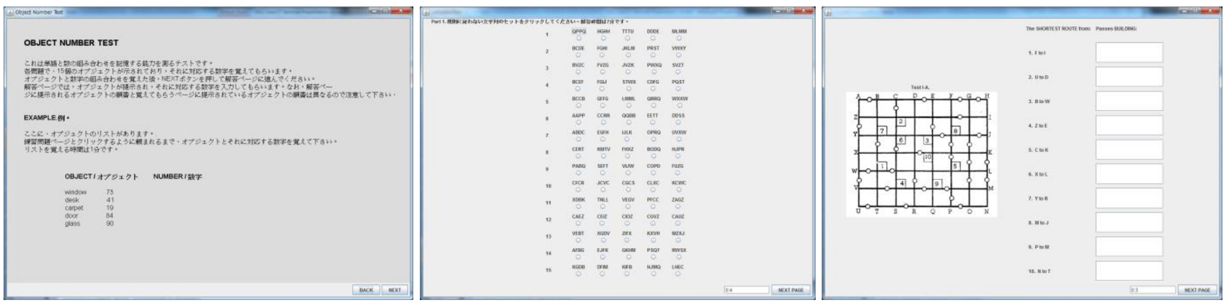

Figure 2. Screenshots of Object Number, Letter Sets and Map Planning Tests

Simulations of fundamental algorithms such as Linear Search, Binary Search, Insertion Sort, Selection Sort and Maze Tracing were also created in order to test the students' inherent ability to implement them even without prior knowledge or formal lesson. Corresponding games or tests were created for these algorithms where student performance such as the number of tries, time and movements were gathered. A scoring scheme based on the number of tries, wins/losses and time was used for the games. Figure 3 shows the screenshots for Linear Search, Insertion Sort and Selection Sort.

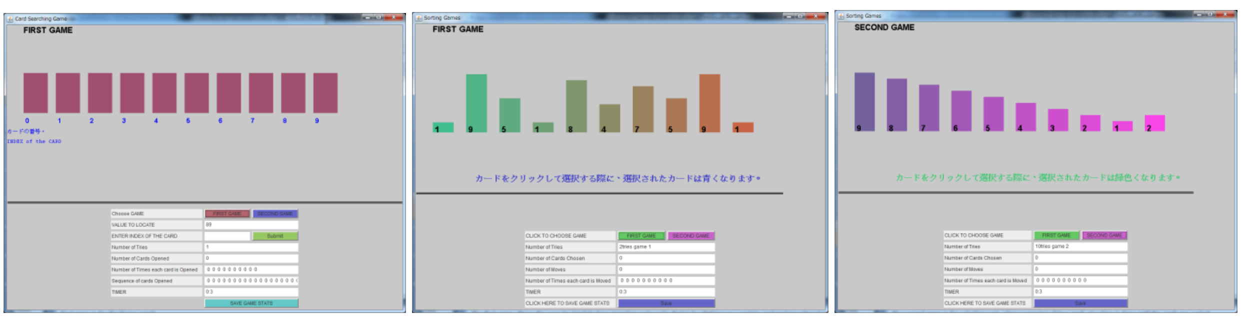

Figure 3. Screenshots of Linear Search, Insertion Sort and Selection Sort Games

The aim is to determine correlations between the chosen cognitive factors and these algorithms and the performance of the forty (40) students of Class II-B in the midterm exam on algorithms. The algorithms included in the test are Linear Search, Binary Search, Insertion Sort and Selection Sort. These topics are taught in the Programming Technology class in the second term starting in September and the midterm exam is usually given in October. The tests were administered in September 2011 before the lessons on algorithms. Another objective is to check for possible predictors of success in the said programming course from among the known cognitive factors and algorithm tests.

Being in its initial stage, other background information on the students were gathered for this study. Some of these information include the students' programming experience and comfort level. These may be included in the statistical analysis later on. A background questionnaire was given to the class in May 2011 which included questions about their programming experience and other pertinent questions. Table 1 shows the four mostly used programming languages by the students. The format of the table is the same as the one used in the questionnaire. C is the most commonly used programming language among the students as it is the required language tool for the course.

	Never used	Used once or twice	Used in one project	Used many times for one or two terms	Used many times for more than year	Formal instruction?	
						Yes	No
C	1	2	1	31	5	40	
C++	23	11	2	3	1	1	14
Java	31	4	2	2	1	1	8
PHP	37	1		2		8	3

Table 1. Programming Experience

The other data gathered for the background study includes the students' knowledge of English and musical ability. Among the forty (40) students, only six (6) have indicated that they can speak English and eighteen (18) can play musical instruments while twenty-four (24) can read musical notes. There is no published research yet on the correlation between musical talent and programming aptitude but there are informal comments made on the issue. The students were also asked how they find programming and they gave their answers using the scale from 1 to 5. Table 2 shows the descriptions for each comfort level and the summary of the students' response. Perhaps these background data can later be included in the analysis.

Comfort Level	Description	Responses
1	very difficult and cannot solve most programming problems	9
2	difficult but can solve some programming problems	18
3	just okay, neither comfortable nor difficult	9
4	comfortable but sometimes cannot solve some programming problems	4
5	very comfortable and can solve any programming problem	0

Table 2. Comfort Level in Programming

Preliminary Results

The cognitive tests as well as those on the algorithms were administered in September 2011 to forty (40) students of Class II-B. The scores on the tests were tallied and the summarized. A portion of the score summary sheet is shown in Table 3 below.

Student Number	Object Number Test	Letter Set Test	Map Planning Test	Linear Search	Binary Search	Insertion Sort	Selection Sort	Maze Tracing
Full score	30	30	40	50.00	50.00	50.00	50.00	87.47
1	25	24	37	10.101	2.676	4.484	-7.845	27.845
2	7	23	34	2.838	9.679	4.348	3.279	29.357
3	20	23	38	8.197	12.346	5.051	5.348	46.727
4	12	23	38	5.291	7.634	3.425	2.924	32.246

Table 3. Part of the Score Sheet Summary for the Cognitive and Algorithm Tests

Initial statistical analysis was done between the cognitive tests and algorithms. Using the Pearson R correlation, the following initial correlations were computed. Table 4 shows the results from SPSS. For each correlation, the Pearson Correlation value and the significance level are provided.

	Associative Memory	Induction	Spatial Scanning	Linear Search	Binary Search	Insertion Sort	Selection Sort	Maze Tracing
Associative Memory	1	.252	-.072	.166	-.082	-.228	-.213	-.091
		.117	.659	.307	.613	.157	.205	.578
Induction	.252	1	.566(**)	.293	.314(*)	-.150	-.121	.055
	.117		.000	.067	.048	.357	.474	.734
Spatial Scanning	-.072	.566(**)	1	.093	.113	.065	-.058	.224
	.659	.000		.567	.487	.692	.732	.165
Linear Search	.166	.293	.093	1	.446(**)	.207	-.098	.283
	.307	.067	.567		.004	.199	.562	.077
Binary Search	-.082	.314(*)	.113	.446(**)	1	.372(*)	.514(**)	.288
	.613	.048	.487	.004		.018	.001	.072
Insertion Sort	-.228	-.150	.065	.207	.372(*)	1	.755(**)	.289
	.157	.357	.692	.199	.018		.000	.071
Selection Sort	-.213	-.121	-.058	-.098	.514(**)	.755(**)	1	.311
	.205	.474	.732	.562	.001	.000		.061
Maze Tracing	-.091	.055	.224	.283	.288	.289	.311	1
	.578	.734	.165	.077	.072	.071	.061	

* Correlation is 95 % significant (2-tailed).
** Correlation is 99% significant (2-tailed).

Table 4. Pearson correlation results among the cognitive factors and algorithms

Among the cognitive factors, Induction correlated with Spatial Scanning. Induction also correlated with Binary Search. These two correlations may be explained by the idea that induction means making generalizations from individual instances. Spatial scanning requires that manner of thinking because a general strategy of finding the shortest route between two points has to be gained from the distinct cases encountered in scanning the city map. The same applies to Binary search where the technique of dividing the search range by half for each iteration can be generalized based on the experience from each try. Insertion sort and Selection sort have a strong correlation perhaps due to the fact that both aim to achieve the same goal of sorting the cards in ascending order. Ability to understand implement one sorting algorithm may also indicate the same ability with other sorting algorithms. The same principle applies for the correlation between Linear Search and Binary Search. Binary Search also correlates with the two sorting algorithms perhaps because all three algorithms apply the same procedure of locating positions in an array in a non-sequential manner.

Future Work

As previously mentioned, this is an ongoing research. So far, only the results of the cognitive and algorithm tests and some background information are available. The results of the midterm exam are currently being analyzed together with the results of the online tests. Other scoring schemes for the algorithm tests are also being applied. This is to further verify the correlations between the known cognitive factors and algorithms and the actual performance of the students in their computer science class. The other background information gathered may also be incorporated in the analysis.

References

CSTA Curriculum Improvement Task Force. (2005). *The New Educational Imperative: Improving High School Computer Science Education*, *Using worldwide research and professional experience to improve U. S. Schools.* ACM, New York.

Sabin, M., Higgs, B., Riabov, V., & Moreira, A. (2005). *Designing a pre-college computing course.* Journal of Computing in Small Colleges.

Rafieymehr, A. (June 2008) *Kids In Computing (K.I.C.): Is There a Solution to Solve the Computer Science Enrollment Problem?* ACM SIGCSE Bulletin, Volume 40, Number 2,pages 107-111.

ACM. (December 2001). *ACM Computing Curricula 2001*, *Computer Science Volume* ,In Chapter 7 Introductory Courses, available at http://www.acm.org/sigcse/cc2001/cs-introductory-courses.html.

Rountree, N. and Rountree, J. and Robins, A. and Hannah, R. (2004). *Interacting Factors that Predict Success and Failure in a CS1 Course.* In ACM SIGCSE Bulletin, 36(4).

Ekstrom, R. B., French, J. W., Harman, H. H., and Dermen, D. (1976). *Manual for kit of factor-referenced cognitive tests 1976.* Educational Testing Service, Princeton, New Jersey.

Avanceña, A.T. & Nishihara, A. (2010). *Automated Tests for Measuring Cognition and Aptitude in Introductory Programming*. In Z. Abas et al. (Eds.), Proceedings of Global Learn Asia Pacific 2010 (pp. 3371-3379). AACE.