

알고리즘 구현을 위한 수학적 기반이론에 대한 사전 설명의 효과에 관한 연구

이덕우

계명대학교 공과대학 컴퓨터공학부

A Study on the Effect of Preliminary Explanation on Mathematical based Theory in Algorithm Implementation

Deokwoo Lee

Division of Information Communication, Hankook University Department of Computer Engineering,
Keimyung University

요약 본 논문에서는 소프트웨어, 컴퓨터과학 또는 컴퓨터공학 분야 전공 과목을 수강하는 학생들이 알고리즘 구현을 위해 필요한 수학적 기반 이론에 대한 설명 위주의 수업의 효과에 대해 살펴본다. 본 연구에서 수행한 설문조사로부터 학생들이 구현해야 하는 알고리즘에 필요한 수학적 기반 이론을 구두로 설명하는 것이 문서로 된 강의자료보다 더 효과가 있다는 것을 알 수 있었다. 본 연구에서는 컴퓨터 분야 전공수업을 듣는 학생들이 프로그래밍 언어를 사용해서 알고리즘을 구현하기 전에 요구되는 수학적 기반 이론에 대한 의견을 알아보기 위해 학부생들에게 몇가지 질문으로 구성된 설문조사를 수행하였다. 학생들은 모두 세 그룹으로 나뉘어졌으며, 첫번째 그룹은 문서로 된 강의자료와 보충자료들만을 활용할 경우 알고리즘 구현의 가능성에 대한 질문을 받았다. 두번째 그룹은 기반이 되는 수학적 이론에 대한 구두설명을 제공받을 경우 알고리즘 구현 가능성에 대한 질문을 받았다. 세번째 그룹은 문서로 된 자료와 구두설명 모두 제공받았을 경우 알고리즘 구현의 가능성에 대한 질문을 받았다. 모든 학생들의 답변을 확인한 결과 학생들은 문서로 된 자료보다 구두설명을 제공받을 경우 알고리즘 구현에 더 효과적이라는 것을 알 수 있었다. 본 연구에서 내린 또 하나의 결론은 학생들은 문서와 구두설명 모두 제공받은 경우와 구두설명만을 제공받은 경우에 대해 알고리즘 구현에 있어 비슷한 효과를 얻는다는 것이었다. 그러므로 본 논문에서는 소프트웨어 알고리즘을 구현하기 전에 해당되는 개발 내용에 대한 지식의 기반이 되는 수학적 이론은 구두설명으로 충분히 전달될 필요가 있다는 결론을 내릴 수 있다.

Abstract This paper deals with the effects of preliminary explanations of mathematical-based theory on algorithm implementations, especially for students who major in software or computer science engineering. In this work, the results of a survey conclude that verbal explanations of the underlying mathematical theory of an algorithm to be implemented using programming languages is more effective than letterpress materials provided a priori that describe the mathematical theories, structure, or flow of the algorithm, etc. When surveying undergraduates in computer engineering with questions to acquire opinions about the effects of explanations of mathematical theory on the feasibility of implementing algorithms using programming languages, those students were segmented into three groups. The first group was provided with lecture notes and detailed supplementary materials, and were then asked to implement the algorithms. The second group was provided with a verbal explanation about the underlying mathematical theories of the algorithms. The third group was provided with both verbal explanations and letterpress materials. The conclusion is that the effect of a verbal explanation of a mathematical theory is almost the same as the effect from provision of both of letterpress materials and verbal explanations.

Keywords : Computer Education, Software Education, Software Implementation, Mathematical Theory, Algorithm

본 연구는 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구 (No. 2019R1G1A1100172)임.

*Corresponding Author : Deokwoo Lee(Keimyung Univ.)

email: dwoolee@kmu.ac.kr

Received June 22, 2021

Revised July 23, 2021

Accepted October 1, 2021

Published October 31, 2021

1. 서론

인공지능 기술은 과학·기술을 포함하고 있는 분야뿐 아니라 일상생활에 많이 적용되고 있으며 사회 전반의 변화와 혁신을 이끌어 가고 있다[1]. 이것은 인공지능 기술을 구성하고 있는 여러 가지 학문 분야의 융합이 실질적으로 이루어지고 있기에 가능하다. 기술적인 측면에서 본다면 인공지능 기술을 구현하기 위해 필요한 소프트웨어 기술의 발전, 소프트웨어 기술 발전을 이끌어 가고 있는 개발자들, 그리고 구현된 소프트웨어 기술이 실제로 상용화되는 것을 뒷받침해 주는 하드웨어의 발전이 있기에 가능한 것이다. 1990년대부터 개인용 컴퓨터(Personal Computer, PC) 보급이 늘어남에 따라 컴퓨터 활용에 대한 교육이 증가하기 시작하였고, 초·중·고등학교에서도 컴퓨터에 대한 지식(컴퓨터 구조 등)에 대한 전문적인 교육이 교육과정에 포함되기 시작하였다[2]. 하드웨어와 제조업 위주의 산업발전은 지난 몇 년간 소프트웨어 기술 중심의 발전으로 그 무게중심이 옮겨지고 있다 [3]. 인공지능 기술에 대한 관심은 소프트웨어 개발 인력에 대한 수요를 더욱더 증가시키고 있으며, 현재는 소프트웨어 개발 분야와 관련된 전공자들뿐 아니라 비전공자들도 교과과정 및 비교과과정을 통하여 소프트웨어 개발에 필요한 지식을 습득하고 있다[4,5].

소프트웨어 개발 업무를 구성하고 있는 것은 소프트웨어에 대한 요구분석, 기본설계, 상세설계, 구현, 시험, 테스트, 운영 및 유지보수이다 [6]. 이 중 가장 중요하면서도 소프트웨어 개발의 완성도에 가장 큰 영향을 끼치는 것은 구현 분야라 할 수 있다. 구현은 프로그래밍 언어를 사용하여 개발하고자 하는 대상을 프로그램화하는 것이라 할 수 있다. 소프트웨어 개발에서 구현 분야는 매우 전문적인 프로그래밍(코딩) 기술을 가진 개발자를 필요로 하고 있고, 개발자의 역량에 따라 개발 결과의 품질과 완성도의 편차가 매우 크다. 개발자가 완벽한 구현을 하기 위해서는 개발하고자 하는 내용에 대한 지식이 있어야 한다. 소프트웨어 개발자가 가진 도메인 지식(domain knowledge)은 한정되어 있는 반면 개발해야 하는 프로그램의 범위는 무한대에 가까울 만큼 매우 넓다. 사람의 일상생활에서 소프트웨어 기술이 접목되지 않은 분야가 거의 없다는 것을 생각해 볼 때, 개발자가 소프트웨어가 적용되는 분야에 대한 모든 지식을 확보한다는 것은 불가능하다고 할 수 있다. 소프트웨어 개발자는 개발 경험을 쌓으면서 여러 분야의 도메인 지식을 경험적으로 축적하면서 자신의 전문 응용 분야를 확립해

나가는 경우가 많다. 소프트웨어 개발에서 프로그래밍 언어를 활용한 구현에 있어서 사용되는 프로그래밍 언어의 종류도 매우 다양하다. 상용화된 제품에 탑재되는 소프트웨어 구현을 위해 사용되는 프로그래밍 언어는 최근 까지 주로 C 또는 C플러스플러스(C++) 였지만, 최근 들어서는 다양한 언어들이 사용되고 있다[7]. 프로그래밍 언어의 종류는 바뀌는 반면에 구현을 위한 설계에서 필요한 수학적 논리는 거의 바뀌지 않기 때문에 소프트웨어 개발자에게 수학적 지식 및 수학적 논리에 대한 이해는 매우 중요하다. 수학적 지식, 수학 논리에 대한 지식을 학습하기 위해 대부분의 대학교의 컴퓨터 관련 전공에서는 이산수학 과목을 개설하고 있고, 선형대수 등의 과목을 개설하는 경우도 있다[8,9].

컴퓨터는 단어에서 보여주는 그대로 계산(compute)을 하는 기계이다. 컴퓨터는 매우 큰 양의 숫자 연산도 사람보다 더 빠르고 정확하게 할 수 있다. 이러한 의미에서 일상생활에서 사용하는 계산기도 컴퓨터의 한 종류라고 할 수 있겠다. 다시 말해서, 컴퓨터 프로그램은 수학 및 논리연산으로 이루어져 있다. 기본적인 수학 및 논리 연산이 완성된 후 시각적인 효과를 보여주기 위한 그래픽, 영상처리 등이 추가되는 것이다. 컴퓨터는 빠른 연산을 할 수 있지만, 어떤 종류의 연산을 해야 하는지 결정하지 못한다. 프로그램 구현에 필요한 연산, 수학적 모델을 결정하는 것은 개발자가 할 수 있으며, 개발자가 효율적이고 정확한 수학적 모델을 설계해야 구현 역시 효율적이고 정확하게 할 수 있다고 볼 수 있다.

인공지능 기술은 대부분 소프트웨어 기술에 의존한다고 할 수 있다. 심층신경망 모델을 사용하기 위해 대용량의 연산을 고속으로 하기위해 고성능의 하드웨어가 필요하지만, 기본적인 구현을 위해 가장 먼저 필요하고 가장 중요한 것으로 간주되는 것은 이러한 모델에 대한 설계 및 수학적 모델의 정립이다. 이것이 정립되어야 프로그래밍 언어로 구현을 하고 컴퓨터가 연산을 실행에 옮기는 것이다. 하드웨어의 부담을 줄여주기 위해 심층신경망 모델의 경량화 연구도 활발히 진행되고 있기 때문에 수학적 모델을 어떻게 효율적으로 설계하느냐에 따라 개발자는 필요한 하드웨어의 성능을 결정할 수 있는 주체적인 입장이 될 수 있다고 하겠다. 인공지능, 사물인터넷, 빅데이터 등의 기술이 사회 전반의 변화를 이끌어 가고 있지만, 소프트웨어 개발자들은 이러한 변화에 가장 능동적이고 주체적으로 대처할 수 있는 전문성을 가지고 있어야 함과 동시에, 대학 및 기타 교육기관은 이러한 전문인력 양성을 위한 노력을 해야 한다.

대학교에서 컴퓨터공학전공 학생들을 위해 해당 전공(또는 학과) 자체에서 개설한 수학과목에 대한 교육은 한 두 학기 정도에 그치고 있다[10]. 필요한 수학적 지식을 학습시키기 위해 컴퓨터공학전공 학과에 별도의 수학 교과목을 개설하는 것은 현실적으로 부담이 크고, 수학과에서 개설하는 과목들과의 중복성 때문에 효율성 또한 매우 낮을 것이다. 그러므로 현재의 상태에서 컴퓨터공학전공 개설과목(전공과목)의 교육을 진행할 때 기반이 되는 수학적 지식을 함께 교육하는 것이 현실적으로 가장 가능성이 높고 효율적인 방법이라고 할 수 있다[11]. “인공지능을 위한 기초수학” 같은 과목들이 신규 개설되기도 하지만, 이러한 과목은 집필진에 따라 내용의 초점이 매우 다르기 때문에 일률적으로 모든 컴퓨터공학 전공 학생들을 위해 활용하는 것 역시 쉽지 않다[12]. 그리고 전공과목에서 수행될 프로그래밍 실습의 기반이 되는 수학과목을 담은 학습자료(문서 또는 책)를 학생들이 스스로 학습하게 한 후 개발 실습(또는 개발 업무)을 하도록 하는 것은 시간적인 효율을 확보할 수 있을 것이다. 그러나, 단순히 자료만을 활용한 사전학습을 통해 수학적 기반 지식을 적절하게 습득하는 것은 어려울 수 있다. 그러므로 전공과목을 강의하는 교수자는 학생들에게 해당 기술 또는 지식에 필요한 수학적 논리, 지식, 모델, 흐름을 적절하게 구두로 설명하는 것이 필요하다. 본 연구에서는 컴퓨터공학 전공 과목(실습이 포함된 과목)에서, 프로그래밍 구현을 위해 수학적 지식에 대한 직접적인 구두설명의 중요성에 대해 살펴보려고 한다. 학생들을 총 세 개의 그룹으로 나누어 다음과 같이 서로 다른 방법의 교육을 수행하였다.

- 방법 1 : 문서로 된 강의자료 및 보충자료만 제공 후 구현 가능성 조사
- 방법 2 : 강의시간에 구두설명(필기 포함)만 제공 후 구현 가능성 조사
- 방법 3 : 강의 시간에 문서로 된 자료와 구두설명(필기 포함) 모두 제공 후 구현 가능성 조사

이렇게 3가지 방법으로 교육을 진행한 후 학생들이 수행한 프로그래밍 구현실습에 어떤 방법의 수학적 지식의 전달이 효과적이었는지에 대해 알아보았으며, 본 연구에서 도출한 결론은 구두설명(필기포함)이 충분히 이해되었을 경우 문서자료의 추가 여부는 크게 중요하지 않았다는 것이다. 즉, 수학적 지식의 전달 방법은 여전히 강연자가 학생들에게 직접 설명하면서 필기를 수행할 때 효과적이었음을 알 수 있었고, 이 방법이 정보기술, 소프트웨어 개발 분야에도 가장 적절한 방법이라는 결론을

내릴 수 있었다.

본 논문에서는 소프트웨어 개발자들 또는 소프트웨어 전공 학생들에게 필요한 수학적 기반지식의 전달은 구두 설명이 반드시 필요하다는 내용을 다루고자 한다. 본 논문의 구성은 다음과 같다. 2장에서는 소프트웨어 개발에서 수학적 지식 및 논리의 중요성에 대해 설명하고, 본 연구에서 다루고 있는 강좌에서 필요한 수학적지식의 사례에 대해 설명한다. 3장에서는 강좌에서 다루는 프로그래밍 실습에 필요한 수학적 지식에 대한 자료(문서) 제공과 수학적 지식에 대한 구두강의에 대해 학생들이 느끼는 중요성에 대한 설문조사 결과 및 분석 내용을 제시하고 4장에서 본 논문을 끝맺는다.

2. 프로그래밍에 필요한 수학적 이론

소프트웨어 개발의 결과물은 실행 가능한 프로그램 또는 프로그램이 탑재된 하드웨어 제품이라 할 수 있다. 프로그램의 완성도를 결정하는 것에는 여러 가지 요소가 있지만 프로그래밍 언어를 활용한 구현 작업이 가장 큰 비중을 차지하고 있다. 그러나 단순 구현을 하는 것은 프로그래머(programmer) 또는 코더(coder)이지, 개발자라고 할 수는 없다. 개발자는 프로그램에 대한 필요성 조사, 개발의 실현가능성 검토, 개발내용에 대한 설계 및 기획을 모두 할 수 있어야 한다. 또한 프로그램 구현 후에 이루어질 시스템 통합, 각종 테스트, 유지, 보수에 대한 지식도 갖추고 있어야 한다. 구현을 제외한 요소 중 가장 중요한 것들 중 하나는 설계라 할 수 있다. 설계는 프로그램을 제작하는 여러 가지 방법 중 가장 최적의 방법을 찾는 과정이라 할 수 있으며, 설계를 어떻게 하느냐에 따라 구현에 필요한 시간, 인력, 장비 등의 자원 및 비용이 결정된다. 즉, 좋은 설계는 적은 비용과 자원으로 최고 품질의 소프트웨어를 만들 수 있는 것을 가능하게 해준다. 여러 가지 방법 중 최적의 방법을 찾는 과정이 설계이기 때문에 설계는 개발자의 주관적인 창의력이 반드시 뒷받침 되어야 한다. 문제를 해결하는 것을 프로그램 구현의 완성이라고 볼 때, 설계는 문제 해결의 방법을 찾는 과정이라 할 수 있고, 이 과정에서 창의성이 필요한 것이다. 문제 해결을 위한 창의적 방법을 찾아야 하는 점에서 소프트웨어 개발은 수학문제를 푸는 것과 비슷한 과정을 거친다고 할 수 있다. 다시 말해서 소프트웨어 개발을 위해 프로그래밍 언어를 사용한 구현작업을 할 때, 어떻게 프로그래밍을 할 것이냐에 대한 설계에는 수학적

지식과 논리가 포함된다. 컴퓨터의 모든 연산은 산술연산과 논리연산으로 이루어져 있기 때문에 프로그램 구현을 위한 설계는 반드시 수학적 모델로 이루어져야 하고, 적절한 수학적 모델을 활용한 설계를 통해 효율적인 구현 작업으로 이루어진다고 하겠다. 현재 대학교의 컴퓨터공학 전공 학생들은 대부분 고등학교에서 자연계열 교육과정을 이수하였지만, 인공지능 기술의 보편화로 인해 점차 계열의 구분 없이 소프트웨어 개발에 대한 능력이 요구되고 있다[13].

본 연구에서는 컴퓨터공학전공에서 개설한 학부전공 과목에서 프로그래밍 실습에 필요한 수학적 기반지식에 대한 전달 방법을 아래와 같이 세 가지로 나누어 각각의 방법을 학생들에게 적용해 보았다.

- 방법 1 : 프로그래밍 대상이 되는 내용에 필요한 수학적 기반 지식을 담고 있는 강의자료, 보충자료 등을 문서의 형태로 학생들에게 제공하여 스스로 학습하게 한 후 프로그래밍 실습을 수행할 경우 완성도 있는 구현 가능성에 대한 의견을 조사함.
- 방법 2 : 프로그래밍 대상이 되는 내용에 필요한 수학적 기반 지식을 수업 당일에 구두 설명(필기를 포함)의 형태로 학생들에게 제공한 후 프로그래밍 실습을 수행할 경우 완성도 있는 구현 가능성에 대한 의견을 조사함.
- 방법 3 : 프로그래밍 구현의 대상이 되는 내용에 필요한 수학적 기반 지식을 담고 있는 강의자료, 보충자료 등을 문서의 형태로 학생들에게 제공하고 이 자료를 가지고 수업시간에 구두설명(필기포함)을 동시에 학생들에게 제공한 후 완성도 있는 프로그래밍 구현의 가능성에 대한 의견을 조사함.

위와 같은 방식을 학부생 3학년과 4학년 총 약 242명에게 적용하였고, 프로그래밍 구현의 가능성에 대한 의견은 설문조사를 통해 수집하였다. 기존의 일반적인 수학강의와 다르게 본 연구에서 다루는 수학지식에 대한 교육의 내용은 프로그래밍 구현에 필요한 수학 내용만 담고 있다. 그러므로 수학전공자의 관점에서 보았을 경우 해결해야 하는 문제의 복잡도나 난이도가 높지 않다. 본 연구를 적용한 컴퓨터공학 전공과목에서 필요한 수학적 기반지식을 요약하면 Table 1과 같다.

Table 1. Basic mathematical theories for major subjects.

Mathematics	Subjects (to be implemented using programming languages)
Convolution	Filtering operation in image processing
Gaussian distribution	Filtering operation in image processing
Calculating average	Filtering operation in image processing
Extracting minimum, maximum and median numbers	Filtering operation in image processing
1st and 2nd derivatives	Filtering operation in image processing
Laplacian	Filtering operation in image processing
Exclusive OR operation	Cyclic redundancy codes algorithm in computer networks
Addition, Multiplication, Division of Binary numbers (0 and 1)	Checksum algorithm in computer networks

3. 교육 방법 적용 사례 및 결과

수학과 소프트웨어 프로그래밍 능력과의 관계에 대한 연구는 기존에 많이 수행되었고, 수학교육에 인공지능을 적용하거나 수학교육에 컴퓨터를 활용한 효과에 대한 연구는 많이 수행되었다[14-16]. 그러나, 컴퓨터 프로그래밍에 필요한 수학적 지식을 어떤 방법으로 전달하는 것이 효과적인가에 대한 연구는 비교적 적게 진행되었다. Table 1에서 설명한 수학적 기반 지식을 학부과정의 컴퓨터공학 전공 학생들(복수전공, 타전공으로부터 전과한 학생들 모두 포함)에게 학습자료 또는 구두강의 형태로 전달하였으며 전달 방법을 요약하면 Table 2와 같다.

Table 2. Categories of methods to deliver mathematical theories.

Method	Provisions
Method 1 (Group 1)	Letterpress materials (lecture notes and supplementary materials) only
Method 2 (Group 2)	Verbal lecture only with taking notes
Method 3 (Group 3)	Verbal lectures (with taking notes and lecture notes) and letterpress materials

Table 2에서 제시한 방법들의 효과를 파악하기 위해 학부 수업을 수강 중인 242명의 학생들에게 설문조사를 실시하였으며, 설문조사의 결과는 Table 3과 같다.

Table 3. Categories of methods to deliver mathematical theories.

Question	Answers	Ratio(%)
Is Method 1 expected to be effective for software implementation?	Strongly Yes	4.46
	Yes	19.64
	Neutral	31.08
	No	35.00
	Strongly No	9.82
Do you think Method 2 is more effective than Method 1 for software implementation?	Strongly Yes	12.50
	Yes	28.33
	Neutral	37.50
	No	17.50
Do you think Method 3 is more effective than Method 1 for software implementation?	Strongly Yes	4.17
	Yes	24.73
	Neutral	28.57
	No	33.52
Do you think Method 3 is significantly more effective than Method 2 for software implementation?	Strongly Yes	10.44
	Yes	2.75
	Neutral	19.37
	No	34.55
Which on will you choose among Method 1 and Method 2?	Method 1	35.60
	Method 2	6.81
	Neutral	3.66
	Strongly No	3.66
Which on will you choose among Method 2 and Method 3?	Method 1	13.61
	Method 2	86.39
	No difference between two	17.28
Which on will you choose among Method 2 and Method 3?	Method 2	64.40
	Method 3	18.32
	No difference between two	18.32

Table 3의 결과로부터 본 연구에서는 수학적 기반 지식의 설명을 필기, 강의자료, 보충자료와 함께 구두로 하는 것이 실제 프로그래밍 구현단계로 넘어갈 때 가장 효과가 있는 것이라는 결론을 내릴 수 있고, 문서로 된 자료들을 학생들에게 미리 제공하여 학습하게 하는 것은 어느 정도의 도움이 될 수는 있지만 (약 23.8%의 학생이 방법 1의 효과에 대해 긍정적으로 응답함) 구두설명을 포함하는 강의의 효과 (약 53.3%의 학생이 방법 1에 비해 방법 3을 선호한다고 응답함) 에 비하면 그 효과가 적다는 것을 알 수 있다. 구두설명만을 사용한 강의를 강의자료만 학습하는 방법보다는 더 효과가 높다는 설문조사 결과를 확인할 수 있고 (방법 2가 방법 1보다 효과가 있

다는 긍정적 응답이 약 , 특히, 구두설명만을 사용한 강의와 구두설명(방법 2)과 강의자료를 함께 사용한 강의(방법 3) 사이에 큰 차이를 느끼지 못한다는 답변도 전체 응답자의 18.32%를 차지하였다 (방법 2를 선택한 학생의 수는 전체의 17.28%). 이 결과를 바탕으로 소프트웨어 개발자들을 위한 프로젝트에 필요한 수학적 기반이론 또는 전공자들에게 필요한 프로그래밍 구현을 위한 수학적 기반이론을 교육할 때에는 대면 방법이든 비대면 방법이든 반드시 강의자가 피교육자들에게 구두로 설명을 하면서 필기를 병행해 가는 방법을 반드시 포함하여 사용하는 것이 적절하다고 하겠다. 현대 사회에서의 교육은 다양한 시청각 자료와 영상그래픽이 포함된 시각적 효과를 극대화하여 집중력을 향상시키는 시도를 하고 있지만, 수단으로서의 수학적 기반이론을 학습하는 피교육자들은 대부분 수학을 주전공으로 하고 있지 않기 때문에 직접 설명을 하면서 이해도를 높일 수 있는 방법이 더 효과적이라 할 수 있다.

4. 결론

본 연구에서는 소프트웨어 개발을 위한 프로그래밍 구현 단계로 넘어가기 전에 설계 단계에서 필요한 수학적 모델링을 위한 수학적 기반지식을 어떻게 전달하는 것이 효과적인지에 대한 연구를 실제 강의에서 적용해 보면서 그 결과를 살펴보았다. 본 연구는 수학적전공자가 아닌 학생들이 컴퓨터공학 전공 과목에서 필요한 프로그래밍 실습을 하기 위해 필요한 수학적 기반지식을 별도의 수학 전공 수업을 통해 배우는 것이 물리적·시간적으로 한계가 있다는 문제점에 주안점을 두었다. 즉, 수학적지식을 목적이 아닌 수단으로 사용해야 하는 컴퓨터공학 전공자들, 특히 프로그래밍 언어를 사용해 개발을 해야 하는 사람들에게 짧은 시간 안에 필요한 도메인 지식과 연관된 수학을 어떻게 효과적으로 제공할 것인가에 대한 방법을 알아보고자 3개의 그룹으로 나누어 설문조사를 실시하였다. 그 결과 수단으로서의 수학적지식은 전공수업 시간에 구두로 필기와 함께 전달하는 것이 문서로 된 풍부한 자료 제공보다 더 효과적임을 확인할 수 있었다. 이러한 결과를 바탕으로 향후 프로그래밍 실습 또는 소프트웨어 개발자들의 직무교육에서 필요한 수학적지식은 해당되는 수학적이론에 대한 자료를 제공하는 것보다 구두로 강의를 수행하는 방법을 고려해 볼 수 있을 것이다.

References

- [1] S. Baek, "Exploring Social Impact of AI", Informatization Policy, Vol.23, No.4, pp. 3-23, Dec. 2016
DOI : <https://doi.org/10.22693/NIAIP.2016.23.4.003>
- [2] E. Lee, "Perspectives and Challenges of Informatics Education: Suggestions for the Informatics Curriculum Revision", The Journal of Korean Association of Computer Education, Vol.12, No.2, pp. 1-10, Mar. 2018
DOI : <https://doi.org/10.32431/kace.2018.21.2.001>
- [3] S. Park, S. Ah, "An Exploratory Study of the Competence of Computational Thinking: for Software Developer", The Journal of Korean Association of Computer Education, Vol.19, No.5, pp.41-53, Sept. 2016.
DOI : <https://doi.org/10.32431/kace.2016.19.5.004>
- [4] E. Jang, J. Kim, "Contents Analysis of Basic Software Education of Non-majors Students for Problem Solving Ability Improvement - Focus on SW-oriented University in Korea", Journal of Internet Computing and Services, Vol.20, No.4, pp. 81-90, Aug. 2019,
DOI : <https://doi.org/10.7472/iksii.2019.20.4.81>
- [5] H. Kim, K. Kim, "Effect Analysis of the SW Curriculum Reorganization for Non-CS Students on the Class Satisfaction", Proceedings of the Korean Association of Computer Education, pp. 45-48, Aug. 2018.
ISSN : 2287-4097
- [6] E. Choi, "Software Engineering", Jeongiksa, 2019,
ISBN : 978-89-353-0522-3
- [7] J. Lee, "Spyder(Scientific PYthon Development EnviRonment)", The Korean Institute of Electrical Engineers, Vol65, No.5, pp. 41-48, May. 2016.
ISSN : 1598-4613
- [8] S-O. Kim, K-M. Ahn, J-W. Lee, "A study for the development of curriculum and courses of mathematics for engineering majors", Communications of Mathematical Education, Vol.23, No.4, pp.961-976.
UCI : G704-002113.2009.23.4.001
- [9] S. Chun, "A Study on Learning Program of Discrete Mathematics for Computer Software", Journal of The Korea Society of Computer and Information, Vol.16, No.2, pp. 235-242.
UCI : G704-001619.2011.16.2.005
- [10] B. Suh, "Sharing Educational Values of Mathematics as Fundamental Liberal Arts", Higher Education, Vol. 189, pp. 59-66, June. 2015.
- [11] S. Lee, "An Analysis of the Middle School Mathematics Curriculum in France: Focusing on 'Algorithms and Programming'", School Mathematics, Vol.22, No.1, pp. 125-159, Mar. 2020.
DOI : <https://doi.org/10.29275/sm.2020.03.22.1.125>
- [12] S-G. Lee, J. Lee, Y. Ham, "Artificial Intelligence and College Mathematics Education", Communications of Mathematical Education, Vol.34, No.1, pp.1-15, Feb. 2020.
DOI : <https://doi.org/10.7468/iksmee.2020.34.1.1>
- [13] S. Lee, J. Kim, "A Study of SW Education for Non-Majors in Sungkyun SW Education iNstitution(SSEN)", Proceedings of The KACE, pp. 107-109, Aug. 2017.
- [14] J. Ee, N. Huh, "A study on the relationship between artificial intelligence and change in mathematics education", Communications of Mathematical Education, Vol.32, No.1, pp. 23-36, Feb. 2018
DOI : <https://doi.org/10.7468/iksmee.2018.32.1.23>
- [15] S. Lee, S-S. Ko, "The Effects of the Mathematical Program, DM[®] Based on Coding Instruction Using Python", The Journal of Educational Research in Mathematics, Vol.28, No.4, pp. 479-499, Nov. 2018
DOI : <https://doi.org/10.29275/ierm.2018.11.28.4.479>
- [16] G. Lee, "Theoretical statistics education using mathematical softwares", The Korean Journal of applied Statistics, Vol32, No.4, pp. 485-502, June. 2019.
DOI : <https://doi.org/10.5351/KJAS.2019.32.4.485>

이 덕 우(Deokwoo Lee)

[정회원]



- 2007년 2월 : 경북대학교 전자전 기컴퓨터학부 졸업(공학사)
- 2012년 12월 : North Carolina State University, USA, 전기컴퓨터공학부 석사, 박사
- 2013년 1월 ~ 2013년 4월 : Washington Univ. in St. Louis, USA, 박사후연구원

- 2013년 9월 ~ 2016년 8월 : 삼성전자 무선사업부 책임 연구원
- 2016년 9월 ~ 2018년 2월 : 영산대학교 가상현실콘텐츠 전공 조교수
- 2018년 3월 ~ 현재 : 계명대학교 공과대학 컴퓨터공학전공 조교수

<관심분야>

신호처리, 영상처리, 컴퓨터비전