

알고리즘과 프로그래밍 교육을 위한 정보과 교육내용체계에 대한 적절성 분석

정영식* · 신수범** · 성영훈***

진주교육대학교 컴퓨터교육과* · 공주교육대학교 컴퓨터교육과** ·

진주교육대학교 컴퓨터교육과***

요 약

알고리즘과 프로그래밍 영역에 대한 초중고등학교 교육과정을 개발하기 위해 국내의 교육과정을 비교·분석하였다. 그 결과 2015 개정 실과 교육과정과 정보과 교육과정에 포함된 내용이 영국이나 미국, 인도와 비교할 때 내용의 범위와 수준이 미흡한 것으로 분석되었다. 따라서 본 연구에서는 이를 개선하기 위해 한국정보교육학회에서 개발한 소프트웨어 교육과정 표준 모델을 참조하여 알고리즘과 프로그래밍 영역을 구분하여 전문가들을 대상으로 설문을 통해 내용의 중요도와 교육 시기의 적절성을 조사하였다. 그 결과, 연구자가 제시한 영역별 내용 중에서 알고리즘의 평가 이외에는 모든 영역에서 중요하다고 응답하였으나 영역별 교육 시기는 전문가마다 의견이 달랐다.

키워드 : 알고리즘, 프로그래밍, 정보과, 교육과정, 적절성

Analysis of Appropriateness in Information Curriculum for Algorithm and Programming Education

Young-Sik Jeong* · Soo-Bum Shin** · Young-Hoon Sung***

Jeonju National Univ. of Education* · Gongju National Univ. of Education** ·

Jinju National Univ. of Education***

ABSTRACT

We compared domestic computer science curriculum with foreign curriculums in order to develop content for algorithm and programming education in elementary and secondary schools. The results show that the levels and ranges of information included in the practical arts and information subjects of the 2015 Revised Curriculum are insufficient when compared with those of England, India, and America. In this paper, we surveyed experts about the importance of content and the age-appropriate time to begin teaching algorithm and programming in schools. The surveys were conducted using questionnaires used in KAIE's software standard model. Except for algorithm evaluation, the experts largely believe all content areas of algorithm education are important. However, they had differing opinions about the best time to begin teaching this subject matter.

Keywords : Algorithm, Programming, Information Education, Curriculum, Appropriateness

교신저자 : 성영훈(진주교육대학교 컴퓨터교육과)

논문투고 : 2016-11-13

논문심사 : 2016-11-15

심사완료 : 2016-11-28

1. 연구의 필요성 및 목적

소프트웨어와 지능형 로봇이 부가가치를 창출하는 4차 산업혁명이 도래하고 있다. 지식의 생산이 폭주하고, 지능형 로봇과 소프트웨어가 사회와 경제 분야에 폭넓게 활용되고 있으며, 교육 분야에서도 서서히 도입되고 있다. 따라서 학교에서는 더 이상 지식을 암기하는 교육보다는 다양한 소프트웨어를 활용하여 지식을 탐구하는 능력을 가르쳐야 하며, 더 나아가 간단한 소프트웨어를 직접 만들어서 문제 해결하려는 능력이 필요하다[23].

영국은 2014년부터 학생들에게 프로그램 사용법에서 벗어나 만드는 방법을 배울 수 있도록 ICT(Information and Communication Technology) 교과 대신에 ‘컴퓨팅(Computing)’ 과목을 신설하여 소프트웨어의 작동 원리를 이해하고, 프로그램을 직접 개발할 수 있는 역량을 키워나가고 있다. 더욱이 5세부터 14세의 모든 학생들에게 컴퓨터 프로그래밍을 배울 수 있도록 결정하였다[1].

이에 발맞춰 우리 정부도 2014년에 ‘소프트웨어 발전 종합 전략’을 제시하며, 미래 세대에 대한 소프트웨어 교육을 강조하였고[14], 교육부 역시 2015년에 발표된 소프트웨어교육 운영 지침과 2015 개정 교육과정을 발표하면서 기존의 정보통신기술 교육에서 탈피하여 소프트웨어 교육으로 전환하겠다고 강조하였다[12][13].

그동안 초중등교육과정에서는 만들어진 소프트웨어를 활용하는 방법을 중심으로 가르쳐왔다[22]. 그러나 최근에는 정보과 교육이 약화되면서 청소년들의 스마트 미디어 중독이나 게임 중독에 대한 문제가 대두되고 있다. 특히 인성과 사고력, 자아 정체성 등이 형성되는 초등학교 시기는 게임 중독으로 인한 폐해는 매우 크다[2][16][19]. 이러한 문제를 해결하는 방안 중 하나는 게임을 직접 제작함으로써 해결할 수 있다. 즉, 게임 프로그래밍은 게임 중독을 예방할 뿐만 아니라[11], 게임 중독을 치료할 수 있으며, 나아가 공부에 대한 자신감을 높이고, 진로 교육에도 도움이 된다[24].

미래에 살아갈 초중등학생들에게 인터넷과 스마트 미디어는 현재뿐만 아니라 미래에도 없어서는 안 될 필수 도구이고, 정보과 교육은 현대인의 필수적인 지식으로 요구되고 있으며, 기초적인 과학의 목록 속에 빠져서는 안 될 시대가 되었다[5]. 정보과교육은 소프트웨어를 활용하는 소양 교육 수준이 아니라 정보화 사회에 능동적

으로 적응하고, 생활 속 문제를 정보기기를 이용하여 해결할 수 있도록 알고리즘적 사고를 길러줘야 한다[15]. 알고리즘은 학생들이 일상생활에서 직면할 수 있는 문제의 본질을 이해하면서 문제를 해결할 수 있고, 이를 통해 체계적인 정보과 교육이 가능하기 때문이다[9]. 또한, 학습에 논리력과 창의력에 대한 요구가 늘어나면서, 알고리즘을 컴퓨터 언어로 구현한 프로그래밍 교육을 통해 분석력, 논리력, 창의력을 함양하려는 다양한 시도가 이루어지고 있다[8].

그러나 2015 개정 실과 교육과정과 2015 소프트웨어 교육 운영 지침에 포함된 알고리즘과 프로그래밍 교육은 그 내용과 시수 측면에서 영국이나 미국, 인도의 교육과정에 비해 매우 빈약하다[21].

첫째, 초등학교와 중학교에서는 정보과가 독립 교과로 편성된 것이 아니라 실과나 기술·가정 등 다른 교과에 통합되어 있어 시수가 부족하다. 더욱이 초등학교의 경우 5~6학년군에만 편성된 실과에 포함되어 있고, 그나마 알고리즘과 프로그래밍 교육과 관련된 시수는 17시간으로 제한되어 있으며, 정보를 별도의 교육 분야로 편성하지 않아 중고등학교의 ‘정보’교과의 연계도 사실상 어렵게 되었다[7][9].

둘째, 정규 교과 이외에서 이루어지는 창의적 체험활동이나 방과후학교 프로그램, 학원에서 이루어지는 알고리즘과 프로그래밍 교육 내용을 살펴보면, 학생들의 알고리즘적 사고를 촉진하고, 문제해결력을 키우는 교육이 아니라 스스로 사고하는 것을 포기하도록 조장하거나 문제해결 의욕을 떨어뜨리고 있다[17].

셋째, 초등학교에서 가르치는 알고리즘과 프로그래밍 교육 내용은 중학교 정보과에 중복되는 부분이 많으며, 내용 수준도 매우 낮아 초등학교에서 현재 실시되고 있는 알고리즘과 프로그래밍 교육은 이미 2015 개정 실과 교육과정에서 제시한 수준 이상을 가르치고 있다[25].

넷째, 2015 소프트웨어 운영 지침에서는 초등학교 3학년년부터 고등학교까지 알고리즘과 프로그래밍 교육을 위한 내용체계를 일부 제시하였으나, 사실상 시수가 확보되지 않아 유명무실해지고 있다[20].

따라서 2015 개정 교육과정에서 알고리즘과 프로그래밍 교육이 체계적으로 이루어지기 위해서는 수시 개정을 통해 충분한 시수를 확보하고, 그에 따른 내용체계가 미리 마련되어야 한다. 그러므로 본 연구에서는 초중등

학교 정보과 교육과정에서 가르쳐야 할 알고리즘과 프로그래밍에 교육에 대한 시수를 확보 내용 체계표를 만들기 위해 국내외 초·중등학교에서 이루어지고 있는 알고리즘과 프로그래밍 내용 요소를 비교·분석하였다. 또한, 알고리즘과 프로그래밍 교육 내용에 대한 타당성과 교육 시기의 적절성을 확보하기 위해 전국 교육대학교 컴퓨터교육과 교수를 대상으로 설문 조사를 실시하였다.

2. 국내외 교육과정 비교 분석

2.1 국내외 교육과정

2000년과 2005년에 발표된 정보통신기술교육 운영 지침의 후속으로 개정된 소프트웨어 교육 운영지침은 2015년 2월에 발표되었고, 2015년 12월에는 실과 교육과정과 정보과 교육과정에서 알고리즘과 프로그래밍 교육을 강화시켰다. 특히 소프트웨어 교육 운영 지침은 초·중등학교에서 2015 개정 교육과정이 적용되기 전까지 소프트웨어 교육과정을 운영하고자 하는 학교에서 활용할 수 있도록 소프트웨어 교육을 하는 데 필요한 교육 목표와 내용, 방법, 평가 등을 안내하고 있다[12].

2.1.1 2015 소프트웨어 교육 운영 지침

소프트웨어 교육 운영 지침은 기존의 ICT 활용 중심의 교육에서 벗어나 소프트웨어를 직접 개발할 수 있도록 다음과 같이 알고리즘과 프로그래밍 교육을 강화하였다.

첫째, 학교급별 목표를 살펴보면, 초등학교에서는 알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 컴퓨팅 사고로 이해하도록 하고, 중학교에서는 간단한 알고리즘을 설계하고 프로그램을 개발하여 문제를 해결할 수 있도록 하였다. 고등학교에서는 알고리즘을 효율적으로 설계하고, 프로그램을 개발하여 창의적으로 문제를 해결할 수 있도록 목표를 설정하였다.

둘째, 알고리즘과 관련된 구체적인 내용을 살펴보면, 초등학교에서는 알고리즘의 개념과 함께 순차, 선택, 반복 구조를 이용하여 문제 해결 절차를 표현하도록 하였

으며, 간단한 정렬, 탐색 알고리즘을 체험활동을 통하여 이해할 수 있도록 하였다. 중학교에서는 알고리즘의 개념을 이해하고, 알고리즘을 자연어, 순서도, 의사코드로 나타내고 제어 구조를 이용하여 알고리즘을 설계하거나, 정렬과 탐색을 활용하여 여러 가지 문제를 해결하도록 하였다. 고등학교에서는 선형구조와 비선형 구조의 개념을 이해하고, 정렬과 탐색 이외에도 압축하는 다양한 방법을 사용하고, 여러 알고리즘을 비교평가할 수 있도록 하였다.

셋째, 프로그래밍과 관련된 내용을 살펴보면, 초등학교에서는 프로그래밍 언어의 기본 요소를 이해하고, 주어진 프로그램을 동일하게 만들거나 수정하여 간단한 프로그램을 만들도록 하였다. 중학교에서는 프로그래밍 언어의 개념, 종류, 개발 환경을 이해하고, 자료의 입출력문, 제어구조, 비교·논리 연산자를 다루고, 프로그램에서 오류를 확인하여 수정할 수 있도록 하였다. 고등학교에서는 변수와 상수, 다중 선택·반복 구조, 배열, 함수, 변수의 참조 범위 등을 이용하여 프로그램을 작성하도록 하고 있다.

2.1.2 2015 개정 교육과정

알고리즘과 프로그래밍 교육과 관련하여 2015 개정 교육과정은 2017년에 중학교 정보과 교육과정이 적용되고, 초등학교는 2018년부터 실과 5~6학년에서 적용된다.

첫째, 초등학교 실과에 포함된 알고리즘과 프로그래밍 교육 내용은 중·고등학교의 정보과 교육과정과 내용 영역 면에서 일관성이 부족하다. 실과는 ‘가정 생활’과 ‘기술의 세계’로 구분하고, 알고리즘과 프로그래밍 교육을 별도의 영역 구분 없이 기술의 세계의 하위 영역인 기술 시스템 속에 포함하고 있다. 그러나 중·고등학교 정보과 교육과정은 문제 해결과 프로그램 영역에 알고리즘과 프로그래밍 영역을 별도로 구분하여 제시하고 있다.

둘째, 알고리즘과 관련된 내용을 살펴보면, 초등학교에서는 알고리즘이라는 용어를 사용하지 않고, 절차적 사고라는 표현을 사용하고 있으며, 중학교는 알고리즘의 의미, 중요성을 이해하고, 문제 해결 과정을 알고리즘으로 구상하고 표현하도록 하고 있다. 고등학교에서는 순차, 선택, 반복 구조를 활용하여 알고리즘을 설계하도록 하고 있고, 수행 시간의 관점에서 알고리즘을 분석하고

비교하도록 하고 있다.

셋째, 프로그래밍과 관련된 내용을 살펴보면, 초등학교에서는 프로그래밍 도구를 사용하여 기초적인 프로그래밍 과정을 체험하고, 자료를 입출력하는 단순한 프로그램을 설계하고, 순차, 선택, 반복 구조를 이해하도록 하고 있다. 중학교에서는 입력과 출력, 변수와 연산자, 제어구조를 활용한 프로그래밍, 소프트웨어의 비교·분석을 포함한다. 고등학교에서는 텍스트기반 프로그래밍 언어를 활용하고, 변수와 자료형, 다양한 연산자, 표준입출력과 파일 입출력, 중첩 제어구조, 배열, 함수를 포함하며, 알고리즘을 프로그램으로 구현하여 효율성을 비교 분석하도록 하고 있다.

2.2. 국외의 교육과정

국외의 교육과정 중에서 알고리즘과 프로그래밍 교육을 초·중등학교에서부터 도입하고 있는 영국과 미국, 인도를 중심으로 살펴보았다.

2.2.1 영국의 교육과정

영국은 프로그래밍 교육을 강화시키기 위해 기존의 ICT 과목을 Computing 과목으로 변경하고, 2014년부터 적용하고 있다[6].

첫째, 알고리즘과 관련된 내용을 살펴보면, 초등학교 1~2학년(KS1; Key Stage 1) 단계에서는 알고리즘이 무엇인지 이해하고, 디지털 기기에서 프로그램으로 구현되는 방법을 제시하고 있다. 3~6학년(KS2; Key Stage 2) 단계에서는 논리적인 추론을 이용하여 알고리즘의 작동 원리를 설명하고, 알고리즘의 오류를 찾아 수정하도록 하고 있다. 중학교 단계에서는 정렬과 검색과 같은 핵심 알고리즘을 이해하고, 동일한 문제를 해결하기 위한 알고리즘의 유용성을 비교하고 평가하는 것을 포함한다.

둘째, 프로그래밍과 관련된 내용을 살펴보면, 초등학교 1~2학년 단계에서는 프로그램의 개념을 이해시키고, 디버깅과 함께 프로그램의 결과를 예측하도록 하고 있다. 초등학교 3~6학년 단계에서는 물리적 시스템을 제어하고 시뮬레이션을 하는 것을 포함하여 프로그램을 설계하고, 작성하고, 디버깅하며, 순차, 선택, 반복을 사

용하고, 변수와 입출력을 활용한다. 중학교 단계에서는 텍스트 기반 프로그램 언어를 포함한 2개 이상의 언어를 사용하고, 배열, 리스트, 테이블과 같은 자료 구조를 활용하며, 프로시저나 함수를 이용하여 모듈 프로그램과 간단한 논리 연산(AND, OR, NOT)을 이해하도록 하고 있다.

2.2.2 미국의 교육과정

ACM(Association for Computing Machinery)과 CSTA(The Computer Science Teachers Association)에서 개발한 Computer Science 교육과정에 따라 많은 주에서 알고리즘과 프로그래밍 교육을 하고 있다[4].

첫째, 알고리즘과 관련된 내용을 살펴보면, 초등학교 1~3학년은 자신의 생각과 이야기를 순서대로 표현하고, 컴퓨터 없이 정보를 나열하거나, 소프트웨어의 동작 원리를 이해하도록 가르치고 있다. 초등학교 3~6학년에서는 컴퓨터 없이 탐색과 정렬과 같은 알고리즘을 이해하도록 하며, 중학교 단계에서는 순차적인 명령을 이용하여 알고리즘을 정의하고, 평가하도록 하고 있으며, 탐색과 정렬 알고리즘을 사용하도록 하고 있다. 고등학교 단계에서는 순차, 선택, 반복, 재귀, 빅데이터와 관련된 알고리즘을 이해하고, 알고리즘의 효과성, 정확성, 명확성을 평가하는 내용을 포함하고 있다.

둘째, 프로그래밍과 관련된 내용을 살펴보면, 초등학교 1~3학년은 간단한 작업을 수행할 수 있는 구문을 작성하고, 3~6학년은 순차문을 이용하여 프로그램 작성하거나, 블록 기반 프로그래밍 언어를 이용하여 문제를 해결하도록 하고 있으며, 중학교 수준에서는 반복, 조건, 논리, 연산, 변수, 함수를 이용하여 문제를 해결하도록 하고 있다. 고등학교 수준에서는 프로그램의 정확도를 높이기 위해 다양한 시험과 디버깅을 수행하고, 라이브러리를 이용하며, 자료형에 맞는 파일 형태를 선택하고, 추상화 툴을 이용하여 규모와 효과, 보안을 고려한 시스템을 설계하는 방법을 포함하고 있다.

2.2.3 인도의 교육과정

인도는 알고리즘과 프로그래밍 교육과 관련된 교과서를 2008년부터 만들기 시작하여 2012년에 1~8학년 컴

퓨터과학 교과서를 완성하였다. 주요 내용을 살펴보면 다음과 같다[18].

첫째, 알고리즘 교육과 관련하여 초등학교 3학년부터 컴퓨터 없이 단계적 활동을 하도록 하며, 4학년에서는 주어진 일의 순서를 적기, 5학년은 주어진 문제 해결을 위해 알고리즘 사고 활용하기, 6학년은 다양한 해결 방법 중에서 선택하기, 7학년은 작업을 체계적으로 나누어 해결하기, 8학년은 배운 것을 여러 분야에 적용하기 등을 포함하고 있다.

둘째, 프로그래밍 교육과 관련하여 3학년은 그래픽 기반의 프로그래밍 언어를 이용하여 간단한 프로그램을 작성하고 실행하기, 4학년은 조건문, 반복문, 이벤트 핸들링 활용하기, 5학년은 키보드 입력, 변수, 리스트 개념 적용하기, 6학년은 텍스트 기반 프로그래밍 언어 사용하여 순서도를 프로그램으로 변경하기, 7학년은 변수, 배열, 입출력 함수 사용하기, 8학년은 서브루틴과 문자열 함수, 라이브러리를 호출하기 등을 포함하고 있다.

2.3 시사점

알고리즘과 프로그래밍 교육과 관련된 국내외 교육과정을 비교분석하면 얻은 시사점은 다음과 같다.

첫째, 소프트웨어 교육 운영지침에 제시된 내용 수준과 2015 개정 교육과정에서 제시된 내용 수준이 서로 일치하지 않는다. 예를 들면, 소프트웨어 교육 운영지침에서는 정렬과 탐색 알고리즘이 초등학교에 제시되고 있으나, 정보과 교육과정에서는 고등학교 수준에 제시되고 있다. 또한, 운영지침에서는 고등학교에 선형과 비선형 자료 구조, 압축 등을 제시하고 있으나, 정보과 교육과정에서는 이에 대한 언급이 없다.

둘째, 우리나라는 프로그래밍 언어를 배우는 시기가 다른 나라에 비해 너무 늦다. 영국과 인도는 초등학교 3학년부터 시작하고 있으나, 우리나라는 초등학교 5~6학년에서 배우고 있다. 텍스트 기반 언어의 경우는 더욱 차이가 크다. 우리나라는 고등학교 때에 가르치고 있으나, 영국과 인도는 중학교 단계에서부터 가르치고 있다.

셋째, 우리나라 초등학교에서 가르치는 알고리즘과 프로그래밍 구성 요소의 수준이 다른 나라에 비해 낮다. 특히 프로그램의 오류를 찾아 수정할 수 있는 디버깅에 대한 개념을 영국에서는 초등학교 저학년에서 도입하고

있으나, 우리나라는 중학교 수준에서 제시하고 있다. 또한, 인도와 영국에서는 순서도와 중첩 반복문, 이벤트, 함수 등의 개념이 초등학교에서부터 도입되고 있으나, 우리나라는 중학교와 고등학교에서 도입된 경우가 있다.

3. 알고리즘과 프로그래밍 교육을 위한 내용 체계

알고리즘과 프로그래밍 교육과 관련된 국내외 교육과정을 분석한 결과를 토대로 우리나라 교육과정의 개선 방안을 제시하기 위해 2014년부터 한국정보교육학회에서 개발한 소프트웨어 교육과정 표준 모델을 참조하여 [26], 알고리즘과 프로그래밍의 내용 영역을 구분하고, 전국 교육대학교 컴퓨터교육과 교수를 대상으로 온라인 설문조사를 실시한 후 내용 영역에 대한 적절성을 검토하였다. 응답자는 39명이었다.

3.1 알고리즘과 프로그래밍의 내용 영역

우선, 알고리즘의 내용 영역은 알고리즘의 개념(일의 순서, 알고리즘의 의미, 프로그램과의 관계), 알고리즘의 표현(문제해결 과정을 말, 글, 그림, 기호, 순서도, 의사코드 표현), 알고리즘의 분해(복잡한 알고리즘 분해, 알고리즘 간의 관계), 알고리즘의 구조(순차, 반복, 조건 구조, 중첩 반복, 복잡한 제어 구조), 알고리즘의 결과(현재 상태와 목표 상태, 결과 예측, 오류 수정, 알고리즘 개선), 알고리즘의 실제(탐색과 정렬, 협력하여 알고리즘 만들기), 알고리즘의 평가(장단점 비교, 수행시간과 복잡도, 성능 평가와 선택) 등으로 구분하였다.

다음으로 프로그래밍의 내용 영역은 프로그램의 개념(프로그램의 의미, 필요성, 종류, 장점, 알고리즘과의 관계, 컴퓨터와의 관계), 프로그램의 작성(간단한 프로그램 작성, 알고리즘을 프로그램으로 옮기기, 산술, 비교, 논리 연산자, 텍스트 프로그래밍), 제어 구조(이벤트, 순차, 반복, 조건, 중첩 반복), 자료형(변수, 변수의 범위, 배열, 리스트), 라이브러리(함수, 매개변수, 사용자정의함수, 표준입출력, 파일입출력, 라이브러리 호출), 프로그램의 평가(다양한 정렬과 탐색, 오류 수정, 성능 평가), 프로그램의 실제(통합개발환경, 디버깅, 문서 작성, 협력하여

프로그램 설계 및 개발) 등으로 구분하였다.

3.2 적절성 검토 방법

알고리즘과 프로그래밍의 내용 요소에 대한 적절성을 평가하기 위해 교육 내용의 중요도와 교육 시기의 적절성을 검토하도록 하였다.

첫째, 교육 내용의 중요도는 리커드 척도를 활용하여 매우 적절하다를 5점으로, 전혀 적절하지 않다는 1점으로 평가하였으며, 교육 시기의 적절성은 초등 1~2학년, 초등 3~4학년, 초등 5~6학년, 중학교, 고등학교 등 5단계 중에서 선택하도록 하였다.

둘째, 교육 내용의 중요도는 조사 결과의 타당도를 분석하기 위해 타당도 비율(CVR; Content Validity Ratio)을 다음과 같이 계산한 후 응답한 패널 수가 39명임을 감안하여 CVR 값이 0.33 이상인 값을 타당한 것으로 판단하였다[10].

$$CVR = \frac{N_e - \frac{N}{2}}{\frac{N}{2}}$$

N_e : 적절하다, 매우 적절하다고 응답한 수

N : 전체 응답자 수

셋째, 교육 시기의 적절성을 분석하기 위해 다음과 같이 합의도를 산출하였다. 일반적으로 합의도는 0.75 이상일 때 합의한 것으로 간주할 수 있다[10].

$$\text{합의도} = 1 - \frac{Q_3 - Q_1}{Mdn}$$

Q_3, Q_1 : 제1사분위 값과 제3사분위 값

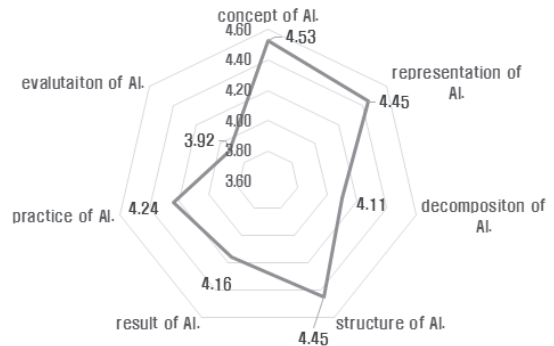
Mdn : 중앙값

3.3 적절성 검토 결과

교육 내용의 중요도와 교육 시기의 적절성에 대한 전문가들의 설문 결과를 정리하면 다음과 같다.

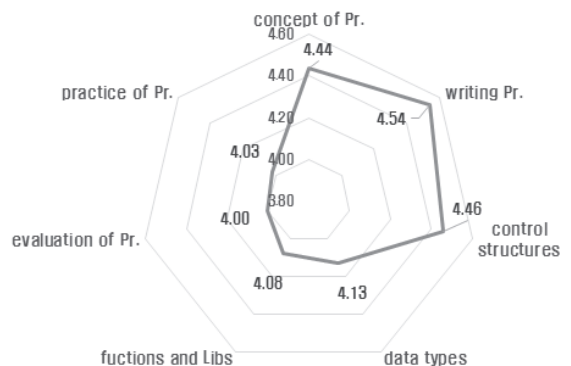
3.3.1 교육 내용의 중요도

알고리즘 교육의 세부 영역별로 내용의 중요도를 분석한 결과 (Fig. 1)과 같이 알고리즘의 평가가 3.92로 가장 낮았지만 여전히 보통 이상으로 나타났다. 알고리즘의 개념은 4.53으로 가장 높았고, 알고리즘의 표현과 제어 구조가 4.45로 높게 나타났다. 응답자의 신뢰도는 알고리즘의 평가만 0.50으로 낮게 나타났고, 나머지는 영역은 모두 0.75 이상으로 긍정적으로 나타났다. 알고리즘 평가 영역은 적절하지 않다는 의견이 응답자의 13.1%를 차지하였다.



(Fig. 1) Importance of algorithm

프로그래밍 교육의 세부 영역별로 중요도를 분석한 결과 (Fig. 2)와 같이 프로그램의 평가가 4.00으로 가장 낮게 나타났지만, 적절하다는 응답이었다. 또한, 프로그램 작성이 4.54로 가장 중요하다고 응답하였고, 다음으

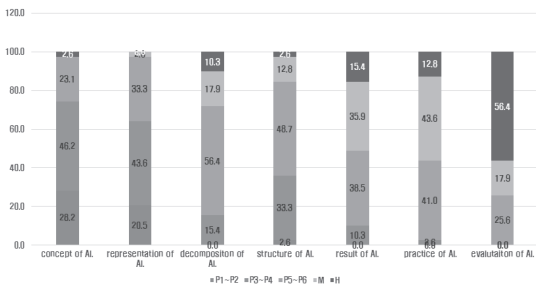


(Fig. 2) Importance of programming

로는 제어 구조가 4.46, 프로그램의 개념이 4.44 순으로 나타났다. 신뢰도는 프로그램의 평가와 프로그램 실체가 0.63으로 낮게 나타났고, 나머지 영역은 모두 0.75 이상을 나타냈다. 프로그램 평가와 프로그램 실체는 모두 7.7%가 적절하지 않다고 응답하였다.

3.3.2 교육 시기의 적절성

알고리즘 영역에 대한 교육 시기의 적절성은 (Fig. 3) 과 같이 나타났다. 알고리즘 결과와 알고리즘 실체는 합의도가 0.75로서 전문가들이 대체적으로 초등학교 고학년이나 중학교에서 가르치는 것이 적절하다고 응답하였다. 그 외의 영역에 대해서는 합의가 이루어지지 못했다.

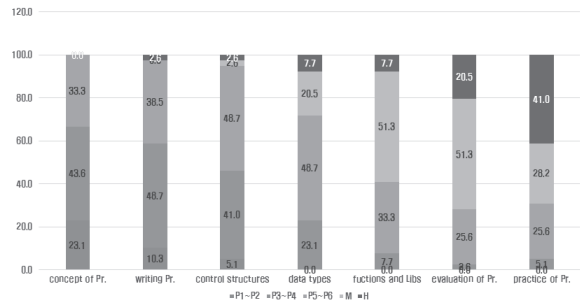


(Fig. 3) The age-appropriate time to begin teaching algorithm

알고리즘 개념과 표현은 초등학교 3~4학년이 가장 적절하다는 응답이 많았고, 알고리즘의 분해와 구조는 초등학교 5~6학년이 가장 적절하다는 응답이 많았다. 알고리즘의 평가는 고등학교가 가장 적절하다는 응답이 많았다.

프로그래밍 영역에 대한 교육 시기의 적절성은 (Fig. 4)와 같이 나타났다. 함수와 라이브러리, 프로그램의 평가는 합의도가 0.75 이상으로 전문가들의 합의가 이루어졌다. 우선 함수와 라이브러리는 중학교에 가르치는 것이 적절하다는 의견이 많았고, 프로그램의 평가는 고등학교가 적절하다는 응답이 많았다. 그 외의 영역은 전문가들의 합의는 이루어지지 못했다. 프로그램의 개념과 프로그램 제작은 초등학교 3~4학년이 적절하다는 응답이 많았다. 제어구조와 자료 유형은 초등학교 5~6학년이 적절하다는 응답이 많았다. 프로그램의 실체는 고등

학교가 적절하다는 응답이 많았다.



(Fig. 4) The age-appropriate time to begin teaching programming

4. 결론 및 제언

알고리즘과 프로그래밍 영역에 대한 국내의 사례를 분석한 결과, 2015 개정 실과 교육과정과 정보과 교육과정에 포함된 내용이 영국이나 인도와 비교할 때 내용의 범위와 수준이 미흡한 것으로 분석되었다.

따라서 본 연구에서는 이러한 문제를 해결하기 위해 한국정보교육학회에서 개발한 소프트웨어 교육과정 표준 모델을 참조하여 알고리즘과 프로그래밍 영역을 세부적으로 구분하고 각각의 세부 영역에 대한 중요도와 교육 시기의 적절성을 분석하였다. 알고리즘에 대한 평가 이외의 모든 영역에서 전문가들은 중요하다고 응답하였다.

교육 시기의 적절성에 있어서는 합의가 거의 이루어지지 않았으나, 알고리즘의 결과는 초등학교 고학년에, 알고리즘의 실체와 함수, 라이브러리, 프로그램의 평가는 중학교 수준에서 다루는 것이 적절하다는 의견을 주었다. 본 연구 결과를 토대로 다음과 같이 제언하고자 한다.

우선, 2015 개정 교육과정을 수정 고시하여 알고리즘과 프로그래밍 교육 내용을 초등학교부터 본격적으로 시작할 수 있도록 해야 한다. 초등학교에서 배우는 내용 수준은 간단한 프로그램을 체험하는 수준이다. 이는 다른 국가의 교육 과정 수준에 비해 매우 낮으므로 초등학교 3학년부턴 알고리즘과 프로그래밍 교육이 가능하

도록 해야 한다.

끝으로, 알고리즘과 프로그래밍의 세부 영역별 성취 기준을 마련해야 한다. 본 연구에서 제시한 영역의 범위가 너무 넓어 특정 학년에서만 가르치는 것에 대한 합의가 쉽지 않았다. 따라서 각 세부 영역별로 구체적인 내용 요소와 성취기준을 추출하고, 각각의 내용 요소별 성취기준에 대한 교육 시기의 적절성을 델파이 조사를 통해 분석한 후 적절한 교육 시기에 대한 합의가 필요하다.

참고문헌

- [1] Bell, T., Witten L. H., Fellows, M., Adams, R. McKenzie, J. (2005). Computer Science Unplugged: An enrichment and extension programme for primary-aged children. University of Canterbury. Computer Science and Software Engineering.
- [2] Changho Lee(2013). Awareness among Youth about the Causes and Solutions of Internet Game Addiction. Youth Policy Reports, 50. National Youth Policy Institute.
- [3] Chul Kim, Gapsu Kim, Hyunbae Kim, Youngsik Jeong, Ingee Jeong(2015). Development of software education curriculum model. Korean Association of Information Education.
- [4] CSTA(2011). CSTA K-12 Computer Science Standards, Revised 2011. TheCSTA Standards Task Force.
- [5] Dookwon Baek, Jongsun Hwang(1993). Computer Science. Hongreung Publishment.
- [6] England Department for Education(2013). Computing Programmes of study for Key Stages 1-4. <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>.
- [7] Eunhoo Park(2007), Development of concrete materials activities teaching-learning program for computer science in elementary school. Korean National University of Education.
- [8] Gyosik Moon(2007). On the Direction of the Computer Algorithm Education Based on Conceptual Algorithms, *Journal of The Korean as-sociation of information education*, 11(1), 29-38.
- [9] Hwakyong Lim, Jinah Kim, Jinho Bae(2004). A Instructional Learning of Sort Algorithm by Role-play for Computer Education in the Primary School. *Journal of the Korean Institute of Information Scientists and Engineers*. 4.
- [10] Injoong Ju, Dongyeol Park, Misug Jin(2010). The Study of Core Competency's Domains and Levels. Korea Research Institute for Vocational Education & Training.
- [11] Jaewon Kwon(2004). A Case Study on Treatment of Youth's Game Addiction using Game Type Substitution Therapy. *Korean Journal of Youth studies*, 11(3).
- [12] Ministry of Education(2015). Software education guideline.
- [13] Ministry of Education(2015). Practical arts(Technology & Home Economics)/Information Curriculum.
- [14] Ministry of Science, ICT and Future Planning(2014). The Action Planning in Software Centric Society.
- [15] Seokyoung Choi, Younghak Jin, Youngsik Kim(2011). Study on Anchored Instruction For Algorithm Learning of Elementary School Students. *Proceeding of the Korean association of computer education*, 15(2), 51-56.
- [16] SeonKwan Han, SooHwan Kim, JungBo Seo(2010). The Development of the Game Addiction Remedy Program based on Scratch Programming. *Journal of the Korean Association of Information Education*, 14(1), 61-68.
- [17] Soon-Jo Hong, Sun-Kwan Han(2004). Development of S/W Component for Search Algorithm Education, *Journal of The Korean as-sociation of information education*, 9(21), 179-187.
- [18] Sridhar, I., Farida, K., Sahana, M., Vijayalakshmi,

C., Malathy, B. and Usha, V. CMC: A Model Computer Science Curriculum for K-12 Schools. IIT Bombay. <https://www.cse.iitb.ac.in/internal/techreports/reports/TR-CSE-2013-52.pdf>.

[19] Sunghyun Baek(2009). The study on relief of elementary students' game addition through the program of game development education. Gyeongin National University of Education.

[20] Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C. and Verno, A. (2002). A Model Curriculum for K-12 Computer Science. Report of the ACM K-12 Education Task Force Computer Science Curriculum Committee.

[21] Youngsik Jeong(2014). A Study on the Content Framework of Algorithm Education in Primary and Middle Schools. *Journal of The Korean association of information education*, 18(2), 275-284.

[22] Youngsik Jeong, Chul Kim(2014). A Study on the Learning Objectives, Instructional Design, and Evaluation Methods in the Software Developing Education. *Journal of the Korean Association of Information Education*, 18(1), 185-193.

[23] Youngsik Jeong, Gapsu Kim, Ingee Jeong, Hyunbae Kim, Chul Kim, Jeongsu Yu, Jongwoo Kim(2015). Development of the software curriculum model of elementary students. *Journal of the Korean Association of Information Education*, 19(4), 457-480.

[24] Youngsik Jeong, Jeongsu Yu(2014). The Effects of Elementary Students' Perception of the Game Programming Using KODU. *Journal of the Korean Association of Information Education*, 18(3), 453-460.

[25] Youngsik Jeong, Soobum Shin, Younghoon Sung(2016). Analysis of Domestic and International Primary and Secondary Curriculum for Problem Solving, Algorithm and Programming.

저자소개



정영식

1996 춘천교육대학교 수학교육학과(교육학학사)
 2001 한국교원대학교 컴퓨터교육과(교육학석사)
 2004 한국교원대학교 컴퓨터교육과(교육학박사)
 2004~2011 한국교육개발원 연구위원
 2004~현재 전주교육대학교 컴퓨터교육과 교수
 관심분야: 컴퓨터교육, 프로그래밍, 이러닝
 E-Mail: nurunso@jnue.kr



신수범

1991 인천교육대학교 (교육학학사)
 1998 한국교원대학교 (교육학석사)
 2002 한국교원대학교 (교육학박사)
 2002~2005 KERIS 연구원
 2005~현재 공주교육대학교 컴퓨터교육과 교수
 관심분야: 컴퓨터교육
 E-Mail: ssb@gjue.ac.kr



성 영 훈

2000. 진주교육대학교(학사)

2002. 진주교육대학교 교육대학원
컴퓨터교육 전공(석사)

2010. 경상대학교 대학원 컴퓨터과
학(공학박사)

2011~2015. 한국교육학술정보원
연구원

2015~현재 진주교육대학교 컴퓨
터교육과 조교수

관심분야 : SW교육, 컴퓨팅융합교
육, 국가행정정보시스템

e-mail : yhsung@cue.ac.kr