

CS와 CS Unplugged의 비교연구

전석주* · 조윤주

서울교육대학교* · 서울신내초등학교

요약

소프트웨어 교육에 대한 관심이 늘어감에 따라 컴퓨터과학에 대한 관심도 높아지고 있다. 컴퓨터과학은 컴퓨터와 컴퓨터시스템의 이론과 원리를 학습하는 학문으로서 이러한 컴퓨터과학의 지식과 개념은 소프트웨어 교육의 근본이라고 할 수 있다. 최근에 컴퓨터를 사용하지 않고 컴퓨터 과학의 핵심 내용을 전달하는 방법으로 컴퓨터과학 언플러그드라는 학습 방법이 전 세계적으로 수업에 많이 활용되고 있다. CS 언플러그드는 다양한 놀이 활동을 중심으로 컴퓨터과학의 원리를 배우도록 하는 학습 방법이다. 그러나 문제는 이러한 CS 언플러그드 활동이 실제 컴퓨터과학에서 배우는 핵심교과와 얼마나 연관성이 있는지에 대한 분석 없이 단순한 놀이 수업으로만 이루어지는 것이다. 따라서 본 연구에서는 컴퓨터과학의 핵심 내용과 CS 언플러그드의 다양한 활동에 들어 있는 컴퓨터과학 핵심 내용을 분석하여 비교 연구를 하고자 한다.

키워드 : 컴퓨터과학, CS 언플러그드, 소프트웨어교육

Comparison Study between Computer Science and CS Unplugged

Seok-Ju Chun* · Yunju Jo

Seoul National University of Education* · Seoul Sinne Elementary School

ABSTRACT

Recent interest in software education has naturally led to interest in computer science. Computer science is the study of theory and principle of computer and computational systems. This knowledge and concept of computer science is fundamental to software education. Recently, "Computer Science (CS) Unplugged", which is an educational method for introducing students to concepts of computer science without using a computer, has been widely used in classes around the world. CS Unplugged is a method for learning the basic principles of computer science through diverse playing activities. The problem, however, is that these CS unplugged activities take place in class without an analysis of how closely they relate to the core subjects learned in real computer science. Therefore, in this study, we will study on the comparison between the core contents of computer science and the core contents of computer science contained in the various activities of CS Unplugged.

Keywords : Computer Science, CS Unplugged, Software Education

2019년도 서울교육대학교 교내연구비에 의하여 연구되었음.

교신저자 : 전석주(서울교육대학교 컴퓨터교육학과)

논문투고 : 2019-10-21

논문심사 : 2019-11-29

심사완료 : 2019-12-20

1. 서론

컴퓨터 과학(Computer Science)이 4차 산업 혁명 사회를 이끌어가는 핵심 학문으로 자리 잡으면서 컴퓨터 과학 전공자뿐만 아니라 타전공자들의 관심도 매우 커지고 있다. 대학 교육과정에서 컴퓨터 과학 분야가 중요해짐에 따라 미국에서는 컴퓨터 과학 교육과정 내용에 대해 지속적인 연구를 해오고 있다. Computing Curricula 2001을 시작으로 2008년과 2013 개편 과정을 거쳐 CS2013 교육과정을 완성하며 지식 영역의 비중, 새로운 지식 영역 생성, 기존 지식 영역의 통합 등을 통해 보다 효과적인 교육이 이루어지도록 지속적으로 교육과정을 수정·운영해오고 있다[7]. 국내에서도 컴퓨터 과학교육에 어떤 지식 요소가 들어가야 하는지에 대해 연구가 진행되고 대학 교육에서는 컴퓨터 과학의 지식 영역을 체계화·구조화시켜 전공자를 위한 커리큘럼을 제공하고 있다. 최근에는 소프트웨어 교육에 대한 사회적·국가적 요구로 인해 대학 커리큘럼뿐만 아니라 초·중등 교육과정까지 컴퓨터과학의 지식 요소들이 흡수되었다. 초·중등 교육과정의 경우에는 컴퓨터과학 핵심 요소를 학생의 발달 단계에 맞게 CS 언플러그드 활동으로 제공하고 있다.

CS 언플러그드는 뉴질랜드의 팀 벨(Tim Bell) 교수가 제안한 프로젝트로서 컴퓨터 없이 컴퓨터 과학(Computer Science)를 학습할 수 있는 학습 방법이다. CS 언플러그드는 소프트웨어 교육의 한 가지 방법으로서 초·중등 교실에서 광범위하게 활용되고 있다. 학생들은 컴퓨터 과학의 원리를 재미있고 어렵지 않게 배울 수 있으며 가르치는 교사에게는 컴퓨터과학의 논리적인 측면을 컴퓨터 사용의 기능적인 측면과 복잡한 프로그래밍 학습 없이 가르칠 수 있기 때문에 다양하게 발전되어오고 있다. CS 언플러그드는 팀 벨 교수가 개발한 학습 활동 중심으로 컴퓨터과학의 핵심 아이디어를 학습하는 것을 중심으로 한다.

CS(컴퓨터과학)과 CS 언플러그드에 대한 관심이 증가하고 있지만 국내의 컴퓨터과학 교육에서 다루는 핵심 요소와 CS 언플러그드에서 다루는 CS 내용에 대한 연구가 부족하며 CS와 CS 언플러그드 내용 체계에 대한 비교·분석연구도 부족한 실정이다. 따라서 본 연구에서는 국내의 CS 교육과정을 분석하고 CS 언플러그드에 포함된 CS 학습 요소를 추출하여 CS 교육과정과 CS 언플러그드의 공통점과 차이점을 비교 분석하고자 한다.

2. 국내의 CS 교육과정 분석

국내의 컴퓨터과학 교육과정을 분석하기 위해서 대표적으로 서울대학교 컴퓨터공학부와 KAIST 전산학부의 교육과정을 살펴봤다. 서울대학교 컴퓨터공학부는 공과대학에 속한 학부로 이학인 KAIST 전산학부에 비해 기본적으로 디지털논리 및 디지털 논리 실험 등의 공학 설계 등의 과목이 추가적으로 포함되어 있다. 우리나라의 대표적인 컴퓨터 과학 분야의 학부교육과정에서 핵심교육인 전공필수 과목과 선택과목을 살펴보고 두 학부의 공통적인 부분에서 컴퓨터과학의 핵심 교육요소를 정리하였다.

2.1 서울대학교 컴퓨터공학부 교육과정

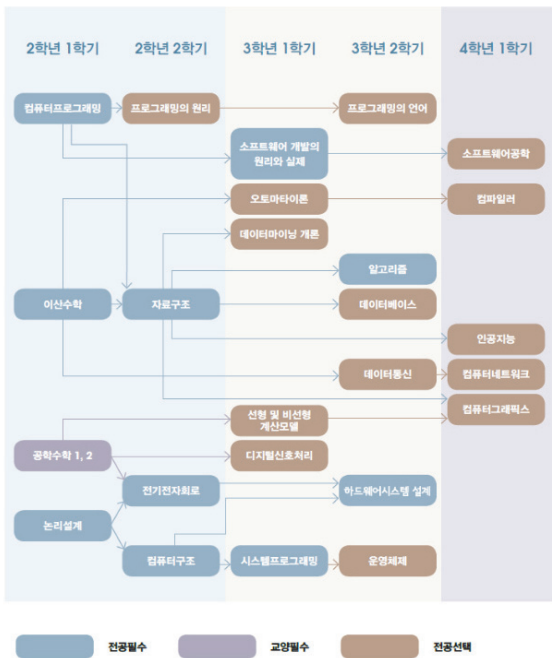
서울대학교 컴퓨터공학부의 경우 컴퓨터공학부 단일 전공학생과 컴퓨터공학 부전공 학생의 교육과정이 서로 다르게 편성되어 있다.

컴퓨터공학부 단일 전공자의 경우 졸업을 위한 학점은 130학점(1996학년부터)이상을 이수해야 졸업할 수 있다. 2015~2018학번 전공자의 경우 전필37학점과 내규 4학점을 포함하여 전공 63학점을 이수해야 한다. 아래의 표에서 교육과정 속에 포함된 컴퓨터과학의 핵심 교과목을 확인할 수 있다.

<Table 1> Computer science and engineering courses of Seoul National University (2015~2018)

Credit	Computer engineering major requires 63 units including essential subjects (37 units) and bylaws subjects (4 units)
Essential Subjects	Discrete Mathematics Logic Design Computer Programming Electrical and electronic circuits Essential Data Structures Computer Architecture Principles and Practice of Software Development System Programming Hardware System Design Algorithm
Bylaws Subject	Computer Science Seminar or Leadership Seminar-IT (choose one) Creative integration design 1 or 2

컴퓨터공학 부전공은 컴퓨터공학부 전공필수를 포함하여 21학점 이상 이수하여야 하는데 전공필수 교과목은 자료구조, 시스템프로그래밍, 컴퓨터구조이며 전공 선택을 위한 교과목은 컴퓨터공학부 전공교과목 중에서 선택할 수 있도록 했다. 전기·정보공학부와 산업공학과에서 개설된 교과목과 유사한 과목이 있으나 2004학년부터 유사 과목 목록에 따라 중복 수강을 금지하고 있다. 또한 컴퓨터공학부에 개설된 과목을 타과의 유사 과목으로 대체를 금지하고 있으며 다만, 타과 과목이 컴퓨터공학부 과목보다 심화된 과목의 경우에만 예외를 인정하고 있다. 컴퓨터공학부 교과목은 교과 내용에 따라 선수학습체계를 가지고 있다. 서울대학교 컴퓨터공학부 선수 교과목 연계도는 그림 1과 같다.



<Fig. 1> Prerequisite diagram of computer science and engineering (SNU)

<Fig. 1>에서 보여 지는 선수 교과목 그래프는 서울대학교 컴퓨터공학부 학부과정에서 어떤 전공과목을 수강하기에 앞서 자신이 과연 수강하려 하는 전공과목에 대해 얼마나 많은 배경지식을 가지고 있는지를 확인해 볼 수 있게 해 주는 그래프이다. 따라서 서울대학교 컴퓨터공학

부에서는 그래프에서 보여 지는 것처럼 어떤 과목을 수강하기 위해서는 그 과목으로 들어오는 화살표가 있는 과목들을 모두 수강하는 것이 배경지식을 습득하고 교과목 내용을 이해하기에 유리하다. 컴퓨터과학의 내용을 난이도 및 연관성에 비추어 체계화 해 놓은 것으로 필수적으로 배워야 하는 지식 요소를 파악하기에 적합하다.

아래의 표는 컴퓨터공학부 학사과정 전공교과목을 1~4학년까지 1, 2학기호 분류해 놓은 것이다.

교과목 중 전공 필수로는 Discrete Mathematics, Electrical and electronic circuits, Computer Architecture, Computer Programming, Logic Design, Data Structures,

<Table 2> Course information of computer science and engineering (SNU)

	1st semester	2nd semester
1st grade		* Programming Practice
2nd grade	* Discrete Mathematics	* Electrical and electronic circuits
	* Computer Programming	* Computer Architecture
3rd grade	* Logic Design	* Logic Design
		* Data Structures
		* Computer Science Seminar
4th grade		* Principles of Programming
	* Automata theory	* Operating system
	* Linear and non-linear calculation model	* Hardware System Design
	* Digital Signal Processing	* Programming Languages
	* Principles and Practice of Software Development	* IT Venture Start-up views
4th grade	* System Programming	* Algorithm
	* Introduction to Data Mining	* Creative integration design 1
		* Database
		* Data Communications
		* Software Applications
		* Mobile Computing and Applications
		* Computer Modeling
	* Embedded Systems and Applications	* Multi-Core Computing
	* Software Engineering	* Computer Security
	* AI	* Web Information Systems
* Compiler	* Computer Fusion Applications	
* Computer Graphics	* Human-Computer Interaction	
* Computer network	* Introduction to Machine Learning	
* Leadership Seminar-IT	* Image Processing	
* Social Network Analysis	* Computer Vision	
* 2 designing creative consolidation	* Computer Technology Topics	
* Topics in Computer Systems	* Internet Security	
	* Based quantum computing and information	

Algorithm, System Programming, Principles and Practice of Software Development 으로서 필수 교과목이 컴퓨터과학에서 가장 중요한 지식 내용이라고 볼 수 있다.

2.2 KAIST 전산학부 교육과정

KAIST 학부생은 전공 선택 없이 무학과로 입학하고 있다. 무학과 입학 후 2학기 때 전산을 포함한 전공을 선택하고 있다. 졸업 학점 이수요건으로는 130학점 이상이며 전공 43학점, 부전공 21학점, 복수 40학점 이상을 이수 요건으로 안내하고 있다. 카이스트 전산학부는 전산학을 비롯하여 연관된 융합학문 전 분야에서 다양한 학사, 석사 및 박사 학위 프로그램을 제공하고 있다. 융합학문 전 분야에 대한 프로그램을 제공하는 이유는 오늘날의 컴퓨팅은 문제 해결뿐만 아니라 다양한 방법으로 사람들의 삶의 질을 향상시키는데 기여하고 있기 때문이다. 컴퓨팅의 가장 토대가 되는 알고리즘, 프로그래밍 언어, 시스템 및 운영체제 등의 연구분야와 더불어 비주얼 컴퓨팅, 소셜 컴퓨팅, 인터랙티브 컴퓨팅 등의 응용 분야가 우리 삶에 더욱 밀접한 분야로 활발히 연구되고 있다.

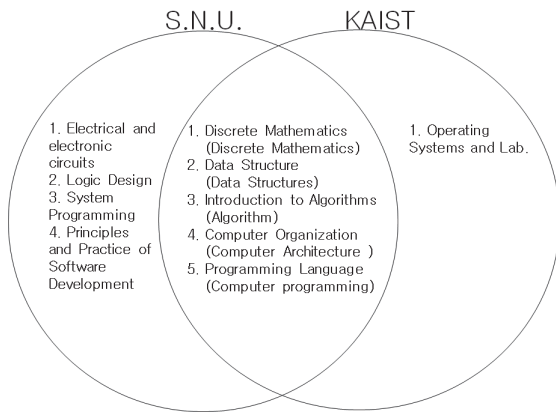
전산학부에서는 기존의 학사, 석사, 박사 학위 과정을 제공하는 전산학부 프로그램(기존의 전산학과 과정)과 더불어 다양한 분야에서 학생들의 재능을 개발하는 프로그램이 운영되고 있다. 소프트웨어심화과정에서는 임베디드 시스템, IT 서비스, UX와 비주얼 컴퓨팅 그리고 인텔리전트 SW로 세분화하여 전산학부의 기본적인 전공과목과 더불어 분야별로 특화된 커리큘럼을 볼 수 있다. 특화된 대학원 프로그램으로서 정보보호대학원은 정보보호 분야에서 석사 및 박사 과정이 있으며, 웹사이트대학원은 특히 웹(Web)을 중심으로 연구하는 커리큘럼으로 석사 및 박사 과정이 있다. 또한, 소프트웨어 대학원 프로그램은 소프트웨어 산업에서의 경영과 소통 능력을 갖춘 국제적으로 선도하는 소프트웨어 엔지니어로 교육하는 프로그램으로 석사 과정이 있다.

아래의 표는 KAIST 전산학부의 교육과정을 정리한 내용으로 필수 교과목으로 프로그래밍 기초, 이산구조, 데이터구조, 알고리즘 개론, 전산기조직, 프로그래밍 언어, 운영체제 및 시험, 전산학 프로젝트가 있다.

서울대학교와 KAIST의 전공필수 교과목을 비교 분석하자면 아래의 그림과 같다.

<Table 3> Course information of computer science (KAIST)

Subject	
Mandatory Basic	Introduction to Programming
Elective Basic	Programming Practice
	Discrete Mathematics
	Data Structure
Mandatory Major	Introduction to Algorithms
	Computer Organization
	Programming Language
	Operating Systems and Lab.
	Problem Solving
	Digital System and Lab.
	Programming Principles
	System Programming
	Creative design of intelligent robots
	Embedded Computer System
	Formal Languages and Automata
	Introduction to Computer Network
	Introduction to Software Engineering
	Introduction to Database
	Symbolic Programming
	Natural Language Processing with Python
	Introduction to Human-Computer Interaction
	Machine Learning
	Introduction to Computer Graphics
	Introduction to Logic for Computer Science
	Computer Science Project
	Introduction to VLSI Design
	Compiler Design
Elective Major	Computation Theory
	Probabilistic Programming
	Mobile Computing and Application
	Distributed Algorithms and Systems
	Introduction to Information Security
	IT Service Engineering
	Business Process Engineering and Management
	Automated Software Testing
	Artificial Intelligence Based Software Engineering
	Software Project
	Requirements Engineering for Smart Environments
	Introduction to Services Computing
	Introduction to Artificial Intelligence
	Introduction to Social Computing
	Text Mining
	Collective Intelligence in Biomedical Applications
	Interactive Computer Graphics
	Introduction to Computer Vision
	Computer Ethic & Social Issues
	Special Topics in Computer Science
	Special Topics in Computer Science I
	Special Topics in Computer Science II



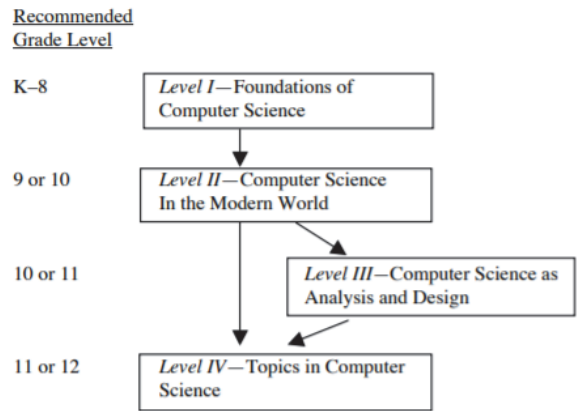
<Fig. 2> Courses comparison of SNU and KAIST

Operating Systems and Lab.을 제외하고는 5가지의 전공 필수교과목이 교과목명만 다를 뿐 내용이 같다. 즉, 컴퓨터공학 전공자로서 필수적으로 배워야 할 내용은 Discrete Mathematics, Data Structure, Introduction to Algorithms, Computer Organization, Programming Language 의 5가지 내용으로 볼 수 있다.

3. CS 언플러그드 분석

CS 언플러그드는 컴퓨터를 사용할 필요가 없는 환경에서 컴퓨터 과학의 지식과 개념을 재미있게 학습할 수 있도록 놀이기반으로 개발한 학습 방법이다. 컴퓨터과학에서 중요하게 다루는 내요 요소를 컴퓨터과학 입문자들이 이해하기 쉽도록 놀이 중심으로 개발한 활동으로 현재도 CS 언플러그드 팀 뿐만 아니라 많은 연구자들이 컴퓨터 과학 학습 주제를 지속적으로 개발하고 있으며 그 수업 효과에 대한 연구도 지속적으로 진행하고 있다[8].

CS 언플러그드에서 다루는 컴퓨터과학 영역은 ACM에서 개발한 K-12의 컴퓨터과학 표준 교육과정에 대한 연구와 함께 이루어지는 것이 좋다. 컴퓨터과학 표준 교육과정에 따른 컴퓨터과학의 구조는 아래의 그림과 같다[9].



<Fig. 3> K-12 CS-Unplugged Curriculum

컴퓨터과학 표준 교육과정에서 다루는 컴퓨터교육 커리큘럼과 더불어 팀 벨 교수의 CS 언플러그드 활동이 어떤 CS의 개념을 어떻게 포함하고 있는지, 학생들에게 어떤 활동으로 제공하고 있는지에 대해 분석할 필요가 있다.

3.1 CS 언플러그드 교육과정

다음의 표는 CS 언플러그드 콘텐츠의 모든 주제에 대한 교육과정을 요약한 것을 보여주는 표이다.

<Table 3> CS unplugged curriculum integrations

Topic	Activity	Curriculum Areas
Binary numbers	Binary Candles or Normal Candles on your Cake	Literacy: Writing
	Binary Candles or Normal Candles on your Cake	Literacy: Writing
	Whose cake is it?	Literacy: Writing
	Binary Name Necklaces	Art
	Binary Patterns	Art
	Binary Tunes	Performing Arts: Music
	Biographies and binary number system history	Literacy: Reading Literacy: Writing
	Binary Art	Art
	Quick card flip magic	Mathematics: Numeracy
	What if?	Literacy: Writing
Error detection and correction	Biographies and error control history	Literacy: Writing
	Instructional writing	Literacy: Writing
	Word study activity	Literacy: Word Study

Kidbots	Storytelling	Literacy: Speaking
	Move to a number	Mathematics: Numeracy
	Finding shapes	Mathematics: Geometry
Searching algorithms	Moving in a shape	Mathematics: Geometry
	Where is my hat?	Literacy: Writing
	Biography	Literacy: Reading Literacy: Speaking Literacy: Writing
	Drama or video activity	Art Performing Arts: Drama
Sorting networks	Music activity	Performing Arts: Music
	Retelling a story	Literacy: Speaking
	Growing into a butterfly	Science: Biology

<Table 4> CS Unplugged topics

Topic	Ages	Lessons	Curriculum integrations	Programming challenges
Binary numbers	5-10	6	7	24
Error detection and correction	5-10	3	5	24
Kidbots	5-10	4	4	50
Searching algorithms	5-10	6	6	NA
Sorting networks	5-14	4	2	NA

3.2 CS 언플러그드 교육 사례분석: Colorado School of Mines(U.S.A.)

미국의 마인즈 중학교에서는 초등학생들 위주로 만들어진 CS 언플러그드 활동을 중학교 학생들에 맞도록 강의계획을 수립하고 3단계의 레슨 플랜을 만들었다. 마인즈중학교에서 CS 언플러그드 교육과정을 개발하는데 있어서 학생 및 교사들에 대한 목표를 설정하였다. <Table 5>에서는 학생과 교사 각각에 대해 어떤 결과물을 기대하는지 나타내고 있다.

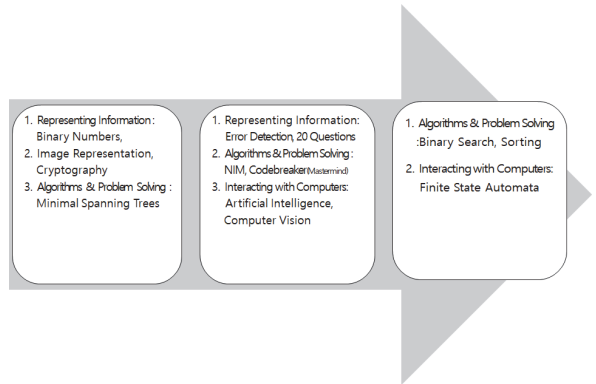
<Table 5> Desired outcomes for students and teachers

Outcome for Students	Outcome for Teachers
<ul style="list-style-type: none"> Show an interest in computing. Gain confidence that they understand computing topics. Understand the fundamental 	<ul style="list-style-type: none"> Use CS Unplugged in their classrooms. Understand the concepts being taught.

<ul style="list-style-type: none"> computing concepts presented to them. Recognize that there is a range of career opportunities in computing 	<ul style="list-style-type: none"> Feel confident teaching CS Unplugged activities.
---	--

CS 언플러그드를 통해 학생들이 컴퓨팅에 대한 관심을 기르고 기본적인 컴퓨팅 개념을 이해하는데 중점을 두었으며 교사들이 CS 언플러그드 활동을 가르치는 것에 대해 자신감을 가지고 컴퓨팅 개념을 이해한 뒤 CS 언플러그드를 직접 교실에서 가르치도록 다양한 자료를 제시하고 있다.

또한, CS 언플러그드를 초급자, 중급자, 고급자 단계 별로 계열화하여 각 단계에 속해있는 학생들에게 적합한 컴퓨터과학 내용 요소(주제)를 제시하고 있다. 아래 그림의 왼쪽 단계부터 초급자, 가운데가 중급자, 오른쪽이 고급자를 위한 컴퓨터과학 내용 요소이다.



<Fig. 4> CS Unplugged concepts for low, intermediate, and high level students

<Fig. 4>에서 보여지는 마인즈 중학교의 3단계 CS 언플러그드 교육과정을 자세히 살펴보면 다음과 같이 분석할 수 있다.

- [1단계] CS Unplugged 처음 도입하는 단계
- 1) Representing Information: Binary Numbers, Image Representation, Cryptography
 - 2) Algorithms & Problem Solving: Minimal Spanning Trees

<분석> 1단계에서는 Representing Information 과 Algorithms & Problem Solving은 이전 섹션에 분석한 CS의 필수 과목에 해당한다고 할 수 있다. 그러나 Cryptography은 1단계에 적합하지 않고 또한 Minimal Spanning Trees는 필수 과목에 해당하지만 CS 전공에서 대학 3학년 수준에서 공부하는 내용이므로 2단계이상에서 교육하는 것이 적합하고 대신 1단계에서는 기본적인 자료 구조나 알고리즘에서 스택, 큐와 몇 가지 정렬 알고리즘을 다루는 것이 더 적합하다고 할 수 있다.

[2단계:] CS Unplugged 기본과정을 마친 후 단계

- 1) Representing Information:
Error Detection, 20 Questions
- 2) Algorithms & Problem Solving:
NIM, Codebreaker (Mastermind)
- 3) Interacting with Computers:
Artificial Intelligence, Computer Vision

<분석> 2단계의 내용에서는 Representing Information은 어느 정도 중학생 수준에서 가능하나 나머지 NIM, Codebreaker (Mastermind), Artificial Intelligence, Computer Vision 등은 실제로 CS의 필수 과목에 해당하지 않고 3단계이상에서 다루든지 아니면 CS의 필수과목에 해당하는 내용으로 수정이 필요하다. 단, Artificial Intelligence은 최근의 가장 큰 이슈이므로 어느 정도 다룰 수 있다고 할 수 있겠다.

[3단계:] 2단계 다음에 추가적인 activities 단계

- 1) Algorithms & Problem Solving:
Binary Search, Sorting
- 2) Interacting with Computers:
Finite State Automata

<분석> 3단계의 첫 번째 Binary Search와 Sorting 은 CS의 필수과목인 알고리즘의 핵심내용에 해당하므로 1단계나 2단계에서 다루는 것이 적합하고 Finite State Automata는 CS전공에서 컴파일러과목을 이수한 대학 3, 4학년에 배우는 수준의 내용으로 굳이 중학생 수준에서 배울 필요가 없고 다른 CS 전공의 필수 교육과정의 내용으로 대체하는 것이 필요하다.

따라서 미국의 마인즈 중학교에서 만든 CS 언플러그드 기반의 교육과정은 각 단계에서 CS의 필수 교육과정을 충실히 반영하지 못하고 있으며 특히, 1, 2단계에 있어야 할 기본적인 CS 내용이 3단계로 배치된 경우와 CS의 필수 내용이 아닌 내용이 CS의 핵심 내용이 배제된 상태에서 교육과정에 포함된 것은 문제라고 할 수 있다. 따라서 마인즈 중학교의 교사들이 CS의 교육과정을 정확히 이해하고 중학생 수준의 CS 언플러그드 교육과정을 설계하였다면 학생들에게 더 적합하고 수준에 맞는 좋은 교육과정을 제공했을 수 있다는 아쉬움이 있다.

3.3 CS 언플러그드 교육 분석

CS 언플러그드, Colorado School of Mines의 CS 언플러그드 교육 사례, K12 Computer Science에서 제시하는 CS의 핵심 개념 등을 분석하였을 때 CS 언플러그드에 포함된 컴퓨터과학 내용 요소는 다음과 같이 정리할 수 있다.

<Table 6> Computer Science Contents included in CS Unplugged

Topic	Activity	Computer Science concept
Data Representing Information	Binary Numbers	
	Image Representation	Discrete
	Text Compression	Mathematics
	Error Detection	Data Structure
Algorithms : Putting Computers to Work	Information Theory	
	Database	Database
	Searching Algorithms	
	Sorting Algorithms	Algorithm
Procedures: Telling Computers What to do	Sorting Networks	
	Minimal Spanning Trees	
	Routing and Deadlock	Operating Systems
	Network Protocols	Network
Intractability: Really Hard Problems	Divide and Conquer	Algorithm
	Finite State Automata	Automata
	Programming Languages	Programming Language
	Programming Languages - Harold the Robot	
Cryptographic: Sharing Secrets	Graph Colouring	Data Structure
	Information Hiding	
	Cryptographic Protocols	Security
	Public Key Encryption	

The Human Face of Computing:	Human Interface Design	A.I.
	The Turing Test	
Interacting with Computers	Artificial Intelligence	
	Phylogenetics	Network communication
Community Activities	Modems Unplugged	
	Divide and Conquer	Algorithm
	Databases	Database
	Scout Patrol(Encryption)	Security

컴퓨터과학의 필수 내용인 프로그래밍, 이산수학, 자료구조, 알고리즘, 컴퓨터 구조 5가지의 내용이 CS 언플러그드에 적절하게 배치되어 있으며 그 중 자료 구조와 알고리즘에 관한 CS 언플러그드 활동이 가장 많았다.

4. 결론 및 제언

컴퓨터과학에 대한 관심이 계속되면서 컴퓨터과학의 핵심 내용에 대한 관심도 증가하고 있다. 컴퓨터과학의 필수 내용 요소는 5가지로 프로그래밍, 이산수학, 자료구조, 알고리즘, 컴퓨터구조로 정리된다. 이러한 CS의 필수 내용 요소는 CS 언플러그드에 다양하게 제작되어 배분되어 있으며 특히, 자료 구조와 알고리즘에 관련된 언플러그드 활동이 많이 제시되어 있다.

그러나 이러한 언플러그드 활동들은 실제 CS전공에서 배우는 핵심 내용이나 단계와 수준을 전혀 고려하고 있지 않으므로 교사들이 이러한 내용을 초등이나 중등에서 교육과정을 만들 때 반드시 CS전공의 필수교육과정을 정확히 이해해서 만드는 노력이 필요할 것으로 생각된다.

참고문헌

[1] Demšar, Irena, Demšar, Janez (2015), CS Unplugged: Experiences and Extensions, *ISSEP 2015*, 106-117.
 [2] Bell, Tim, Vahrenhold, Jan (2018), CS Unplugged – How Is It Used, and Does It Work?, *Adventures Between Lower Bounds and Higher Altitudes*,

497-521.
 [3] Pang, H., Zhou, L., Chang, X. (2011), Research on Course System Construction on Computer Science & Technology Specialty, *International Conference on Advances in Information Technology and Education*, 495-499.
 [4] Inceoglu, M. M. (2005), A Discrete Mathematics Package for Computer Science and Engineering Students, *Computational science and its applications, ICCSA 2005*, 538-546.
 [5] Bell, T. , Rosamond, F., Casey, N. (2012), Computer Science Unplugged and Related Projects in Math and Computer Science Popularization, *The Multivariate Algorithmic Revolution and Beyond*, 398-456.
 [6] Falkner, Katrina , Vivian, Rebecca (2015), A review of Computer Science resources for learning and teaching with K-12 computing curricula: an Australian case study, *Computer science education*, 25(4), 390-429.
 [7] Younjun Lee, Chungwon Choi (2014), *CS2013 analysis for improving college computer science curriculum*, Korean Society of Computer Information.
 [8] Sun-Gwan Han (2019), *Digital Contents for Learning Computer Science using Unplugged CS*.
 [9] A. Tucker (2003), A Model Curriculum for K - 12 Computer Science: Final Report of the ACM K - 12 Task Force Curriculum, ACM, site(http://www.acm.org/curric_vols/k12final1022.pdf)
 [10] H. S. Han, S. K. Han (2009), A Case Study on Information Education for Pre-Service Teacher using Unplugged Computing, *Journal of The Korean Association of Information Education*, 13(1), 23-30.
 [11] J. H. Park (2017), The Effect on Unplugged Computer Science Edutainment Activity based on Computing Fairy Tales, *Journal of the Korea Entertainment Industry Association*, 11(8), 281-288.
 [12] Ford, V., Siraj, A., Haynes, A., Brown, Capture the flag unplugged, *SIGCSE 2017*, 225 - 230.
 [13] <http://csunplugged.mines.edu/index.html>
 [14] <https://k12cs.org> K-12 Computer Science

저자소개



전 석 주

2002 한국과학기술원 컴퓨터 공학 박사

2004~현재 서울교육대학교 컴퓨터교육과 교수

관심분야 : 컴퓨터교육, 프로그래밍방법, 데이터마이닝, 멀티미디어DB

E-mail : chunsj@snue.ac.kr



조 윤 주

2017 서울교육대학교 대학원 컴퓨터교육전공 석사

2018~현재 서울교육대학교 대학원 컴퓨터교육전공 박사과정

2014~현재 서울신내초등학교 교사

관심분야 : 컴퓨터교육, 프로그래밍교육, AI교육

E-mail : yunju0514@gmail.com