

파이썬을 활용한 탐색 알고리즘 수행시간 분석이 초등학생의 논리성에 미치는 효과

양영훈 · 공기표 · 김종훈
제주대학교

요약

본 연구는 초등학생의 논리성 향상을 위해 파이썬을 활용한 탐색 알고리즘 기반 수행시간 비교 및 분석 교육 프로그램을 개발하고 적용하여 그 효과를 분석했다. 본 교육 프로그램은 ○○도내 초등학교 6학년 133명을 대상으로 실시한 사전 요구분석 결과를 활용하였고, ADDIE 모형의 절차에 따라 개발하였다. 개발한 교육 프로그램의 효과를 검증하기 위해서 ○○대학교에서 실시한 교육기부 프로그램의 지원자 25명을 대상으로 6일간 42차시 수업을 진행하였고, GALT검사를 통해 교육의 사전·사후 효과를 비교·분석하였다. 분석해 본 결과, 본 연구에서 개발한 SW교육 프로그램이 초등학생의 논리성에 긍정적인 영향을 줄 수 있다는 것을 알 수 있었다.

키워드 : 파이썬, 탐색 알고리즘, SW교육, 논리성

Effect of search algorithm execution time analysis education on logical thinking of elementary school student

Yunghoon Yang, Gipyoo Kong, Jonghoon Kim
Jeju National University

ABSTRACT

The purpose of this study is to develop a Python search algorithm educational program based on execution time to improve the logical thinking of elementary school students. This educational program was developed based on the process of ADDIE model, utilizing the results of pre-demand analysis conducted for 133students in elementary school in ○○. In order to verify the effectiveness of the developed education program, 25 students who participated in the education donation program at ○○ University conducted 42 hours of classes during 6 days. The GALT test was used to analyze the educational effects of the pre- and post-test. The results of the analysis show that the SW education program developed in this study can positively affect the logic of elementary school students.

Keywords : Python, Search algorithm, Software education, logical thinking

교신저자 : 김종훈(제주대학교 초등컴퓨터교육전공)

논문투고 : 2019-04-21

논문심사 : 2019-04-17

심사완료 : 2019-04-29

1. 서론

최근 세계 경제·사회 환경이 소프트웨어를 중심으로 급격하게 변화하고 있다. 여기서 말하는 소프트웨어 중심사회는 소프트웨어가 혁신, 성장 그리고 가치창출의 중심이 되어서 개인과 기업 그리고 국가의 경쟁력을 좌우하는 사회라고 할 수 있다[24]. 이러한 시대의 흐름을 반영하여 교육의 방향도 변화하고 있다. 기존의 컴퓨터 활용교육 중심에서 SW교육이 정보교육의 주류가 되고 있다. SW교육은 국가의 경쟁력을 향상시킬 수 있는 수단이며 문제해결능력과 논리적 사고력을 길러주고 컴퓨터의 정보처리과정을 이해할 수 있게 한다는 측면에서 교육적 가치를 지닌다. 또한 21세기의 의사소통 방식이 디지털화 되어간다는 점을 고려 할 때, 프로그래밍 능력은 모든 학습자 더 나아가 사회 구성원 전체가 갖춰야 할 기본 소양이 된 것이다[5].

교육선진국이라 할 수 있는 영국, 미국, 핀란드 등에서는 이미 소프트웨어에 관심을 가지고 필수 과목으로 지정하여 교육을 실시하고 있다[10]. 우리나라 역시 2015 개정 교육과정의 역량 중 지식정보처리 역량을 신장시킬 수 있게끔, 학교에서는 2018년부터 SW교육이 의무화되고, 2019년부터 초등학교에서도 5~6학년 학생들이 SW교육을 필수적으로 이수하도록 개정되었다[13].

SW교육을 위해 만든 교육용 프로그래밍 언어의 종류는 다양하다. 하지만, 학생들에게 흥미와 관심을 가지게 하면서 학습하기 쉽고 여러 가지 형태의 응용프로그램으로 확장할 수 있는 언어로 접근하는 것이 알맞다[11]. 텍스트 언어를 배우면 다른 텍스트 프로그래밍 언어에 전이성이 높아서 기본적인 프로그래밍의 지식과 구조를 쌓게 되고 그 지식을 프로그램을 개발하는 데 활용, 적용할 수 있어서 용이하다. 텍스트 프로그래밍 언어가 익숙해지기까지 시간이 오래 걸리지만 복잡한 프로그래밍을 하는 데 유리하고 블록형 프로그래밍 언어에 비해 정보과학사고능력(Computational thinking) 향상에 도움이 된다는 연구결과가 있다[18]. 그러나 대부분의 초등교육현장에서는 초기진입단계가 쉽고, 흥미롭게 접근할 수 있는 요소가 많은 엔트리나 스크래치와 같은 블록코딩 기반의 교육용 언어를 사용하고 있었다.

이에 따라 본 연구에서는 초등학생의 논리성을 향상시키기 위해 텍스트 프로그래밍 언어인 파이썬을 활용

하였고, 사전 요구분석을 하여 파이썬을 통한 탐색 알고리즘에 수행시간 비교 및 분석에 중점을 두어서 교재 및 프로그램을 개발하였다. 초등학교 4, 5, 6학년 학생들 중에서 지원자 표본 25명의 학생을 대상으로 투입하였다. 논리성은 GALT검사지의 논리력 사고력 항목인 ‘보존(Preservation)’, ‘비율(Ratio)’, ‘변인통제(Variable control)’, ‘확률(Probability)’, ‘상관(Correlation)’, ‘조합(Combination)’ 6가지 요소들로 구분하여 설정하였다.

2. 이론적 배경

2.1 SW교육

SW교육이란 현실 속에 존재하는 다양한 문제에 대해 해결방법을 ‘컴퓨터’ 기반으로 찾는 것을 말한다. 즉, 컴퓨터를 이용하여 자료를 수집, 분석하고 효율적으로 해결하는 과정을 통해 사고력을 향상시키는 교육이라고 할 수 있다[27]. 학습자는 소프트웨어 교육을 통해 컴퓨터가 문제를 해결하는 방식에 대해 이해하게 되며, 컴퓨터의 문제해결 방법과 비슷한 논리로 문제에 접근하게 된다. 이러한 일련의 과정을 거치면서 자연스럽게 프로그래밍 언어를 학습하게 되고 문제해결능력 및 사고력 향상을 기대할 수 있게 된다[12].

SW교육에서는 창의융합 인재를 양성하는 것을 목표로 삼으며 ‘생활과 소프트웨어’, ‘알고리즘과 프로그래밍’ 그리고 ‘컴퓨팅과 문제해결’의 3가지 영역을 통해 정보윤리의식과 태도를 통해 실생활에서 발생하는 문제들을 ‘컴퓨팅 사고력’으로 해결할 수 있는 역량을 지닌 인재로 키워내는 교육이라고 할 수 있다[7]. 비교적 최근에 들어서야 SW교육에 관한 연구가 활발하게 진행되고 있기에 SW교육의 개념적 구분이 아직까지는 명확히 합의된 것은 아니지만 공통적으로 ‘컴퓨팅 사고’를 통해 학습자의 문제 해결력 향상에 좀 더 중점을 두고 있는 교육 방법이라고 말할 수 있다[20][21].

2.2 파이썬(Python)

Python은 1991년에 네덜란드의 귀도 반 로섬(Guido

van Rossum)이 개발한 객체지향 프로그래밍 언어로, 국내에서도 많이 알려져 있는 공개 소프트웨어 중 하나이다. 외국에서는 학습의 목적은 물론 실용적인 부분에서도 많이 사용되고 있는 프로그래밍 언어이다. 그 대표적인 예를 보면, 구글(Google)이나 인포시크(Infoseek)에서 사용되는 검색 프로그램들, 야후의 인터넷 서비스 프로그램, NASA, 유튜브(Youtube) 등이 Python으로 개발되었다. 그리고 Python은 윈도우(Windows)와 리눅스(Linux) 그리고 매킨토시(Mac) 등 대부분의 운영체제를 지원하기 때문에 이식성과 확장성에 있어서 좋다[25]. 그리고 파이썬 프로그래밍 언어는 비교적 배우기 쉽고 그래픽 처리 기능이 단순해서 프로그래밍을 처음 접하는 초보자가 배우기에 적절하다. 또한 앱이나 웹 형태로도 개발하기 유용하기 때문에 융합형 교육을 한 프로그래밍 언어로도 활용 가능성이 높다는 장점을 가지고 있다[26].

2.3 논리적 사고

논리적 사고력의 개념에 대해서 학자들마다 의견이 다르며 합의된 의견을 도출하기에는 어려움이 있어 보인다[19]. 하지만 대체로 논리적 사고는 귀납적-연역적 추리와 관련되며 보다 넓은 비판적 사고라고 볼 수 있다[6][2]. 좁은 의미의 논리적 사고는 전통적인 논리학의 근간이 되며 논증에 있어서 형식 논리에 보다 중점을 두고 있다. 반면 넓은 의미의 논리적 사고는 사고의 형식적인 측면을 기초로 하지만 사고 내용의 타당성 여부를 고려하는 비판적인 측면이 부가되는 정신능력을 일컫는다[3].

본 연구에서는 논리적 사고에 대한 개념을 ‘대상들을 논리적으로 분석 비교하고, 비판적으로 볼 수 있으며 주어진 대상들 간의 관계를 타당성에 근거를 두고 사고하는 추리 능력’으로 정의하고자 한다.

2.4 텍스트형 프로그래밍 언어

본래 텍스트형 프로그래밍 언어는 컴퓨터 프로그래밍에 필요한 명령어들을 문자로 입력하는 방식으로 정확하게 구문을 사용해야 한다는 것과 언어 습득을 위해 오랜 시간이 걸린다는 단점들이 있다. 하지만 프로그램의 작성 능력이 향상됨에 따라 프로그래밍 시간을 단축하고, 비교적 짧은 코드로 다양한 프로그래밍을 만들 수

있다는 장점이 있다. 그리고 디버깅을 통해서 분석적이고 논리적인 사고가 향상될 수 있다[22].

텍스트형 프로그래밍 언어를 사용하는 학습 프로그램에서 먼저 고려해야 하는 것은 학습자들의 인지적 부담의 해소 문제이다. 학습이 이루어지는 동안 학습자는 프로그래밍 언어의 문법 학습에 치중하기보다는 문제를 해결하는 과정과 방법이 더욱 중요하다[15].

2.5 탐색 알고리즘

탐색 알고리즘은 기억 공간에 저장된 데이터 혹은 주어진 입력 데이터 집합에서 어떠한 조건이나 성질을 만족하는 데이터를 탐색하는 알고리즘으로써, 자료 구조 및 알고리즘 과목에서 가장 기본이 되는 알고리즘이다. 탐색 알고리즘은 실제 프로그래밍에서 아주 많이 쓰이고 있으나, 기본 개념이 간단하고 직관적이라서 교육에 있어 수월한 알고리즘으로 알려져 있다[4][9].

본 연구에서는 탐색 알고리즘 중에서 주어진 데이터 집합에서 원하는 데이터를 처음부터 순차적으로 찾는 선형 탐색(Linear Search)와 정렬된 데이터 집합에 대해 이분화하면서 탐색하는 방법인 이진 탐색(Binary Search)을 중점적으로 비교하면서 수행시간을 분석하고자 한다.

2.6 알고리즘 수행시간

알고리즘 수행시간은 알고리즘이 어떤 문제를 해결하는 데 걸리는 시간을 말한다. 동일한 문제를 해결하는 데 있어 같은 결과를 도출하더라도 얼마나 오랜 시간이 걸리는지에 따라서 컴퓨터의 효율이 결정되기 때문에 알고리즘 성능을 평가하는데 중요한 기준이 된다. 알고리즘 수행시간을 시간복잡도(Time Complexity)라고도 하며 명령문의 수행 빈도수를 측정하여 분석할 수 있다. 본 연구에서는 단순히 문제를 해결하는 것을 넘어서 최소한의 시간으로 문제를 해결하는 것에 대한 고민과 중요성을 느껴볼 수 있는 알고리즘 교육에 중점을 두었다.[1].

2.7 선행연구분석

김진동, 양권우(2010)의 연구에서는 초등학생에게 실

생활 속 사례들에서 여러 가지 알고리즘 학습을 실시하였다. 그 결과 논리적 사고력에 있어서 유의미한 차이가 있음을 보였다[8]. 본 연구에서는 이를 발전하여, 컴퓨터를 활용한 SW교육에서 알고리즘에 대한 교육이 논리적 사고력에 미치는 영향을 분석하였다.

박대륜(2018)의 연구에서는 초등학교 6학년을 대상으로 로봇 활용 파이썬 학습 프로그램을 개발하고 적용하였다. 그 결과로 로봇 활용 파이썬 학습 프로그램은 학생들의 컴퓨팅 사고력(CT)에 긍정적인 영향을 주었으며 초등학교 학생들에게도 텍스트형 프로그래밍 언어를 이용한 학습 프로그램이 의미 있다는 점을 밝혔다[15].

문미예(2018)의 연구에서는 블록형 프로그래밍 학습이 가지는 한계점들을 극복하기 위해서 텍스트형 프로그래밍 언어인 파이썬으로 교육하기 위한 방안을 개발하였다. 그 결과, 학생들에게 긍정적인 동기를 부여하고 학생들의 아이디어와 생각을 구현할 수 있는 학습도구가 될 수 있을 것이라는 결론을 내렸다. 다만, 학생들에게 실제로 적용을 해보지 못한 한계가 있음을 밝혔다[14].

유인환(2018)의 연구에서는 텍스트 기반 프로그래밍 언어인 파이썬을 적용한 SW교육이 기존의 코드블록 중심의 SW교육에 다양성을 추구할 수 있으며, 앞으로의 활용 가능성에 의미를 발견할 수 있었다고 밝혔다. 다만 한계점으로 초등학교 학생에게 영어의 의미 파악, 코딩 방식 자체에 대한 어려움이 있었고 보완 대책이 필요함을 제안하였다[23].

3. 연구 방법

3.1. 연구 가설

귀무가설: 파이썬을 활용한 탐색알고리즘 수행시간 중심 SW교육이 학습자의 논리성에 미치는 긍정적인 영향이 없다.

대립가설: 파이썬을 활용한 탐색알고리즘 수행시간 중심 SW교육이 학습자의 논리성에 미치는 긍정적인 영향이 있다.

3.2 연구대상

본 연구에서 개발한 프로그램의 효과를 알아보기 위

해 OO대학교에서 교육 기부 프로그램을 실시하였다. 프로그램의 지원자 표집에 의한 지원자 표본(Volunteer sample) 25명의 학생을 선정하였다. 연구 대상 관련 학년 및 성별은 <Table 1>과 같다.

<Table 1> Grade and gender of the subject

class	Male	Female	Total
4 Grade	5	0	5
5 Grade	6	2	8
6 Grade	7	5	12
Total	18	7	25

3.3 교육 프로그램

본 연구에서는 교육 프로그램 개발의 모형 중 하나인 Dick과 Carey의 ADDIE 모형에 따라 교육 프로그램을 <Table 2>와 같이 개발하였다. 총 5단계인 분석, 설계, 개발, 실행, 평가를 순서대로 실시하였고 학습자의 요구분석 결과에 따라 교육 프로그램을 설계 및 개발하였고 수업을 통해 실행했으며 그 결과를 논리성 검사를 통해 평가하였다.

<Table 2> Educational Program Development Plan

Analysis (분석)	학습자 분석 사전 요구분석(고학년 대상 설문)
Design (설계)	수행목표 설정 및 활용도구 선정 -파이썬을 활용한 탐색알고리즘 수행 시간 교육의 효과 평가도구 설계 -논리성 검사(GALT)
Development (개발)	교수·학습 과정안(42차시) 학생 활동지(42차시)
Implementation (실행)	SW교육 실시
Evaluation (평가)	학습자 사후 논리성 검사(GALT)

3.3.1 요구분석

본 연구는 ADDIE모형의 절차에 따라 Rossett의 요구 분석 모형을 사용하였다. Rossett 모형은 기업 교육에서 활용되는 교육 요구 분석 모형으로써 요구 분석의 실행자들이 적용하기 쉬운 안내를 제공한다. 요구분석은 초등

학교 6학년 133(남72, 여61)명을 대상으로 실시하였다.

<Table 3> SW educational experience

	O	X
Students	82(61.7%)	51(38.3%)

SW교육을 받아 본 적 있는지의 여부는 <Table 3>과 같다. 과반수 이상의 학생들이 SW교육 경험이 있음을 알 수 있었다.

<Table 4> SW education Interests

Interests	Students
Very likely	19(14.3%)
Somewhat likely	35(26.3%)
Neutral	46(34.6%)
Somewhat unlikely	16(12%)
Very unlikely	17(12.8%)

SW교육에 대한 학생들의 관심도는 <Table 4>의 결과와 같다.

<Table 5> SW education Necessity

Necessity	Students
Very likely	29(21.8%)
Somewhat likely	44(33.1%)
Neutral	43(32.3%)
Somewhat unlikely	12(9%)
Very unlikely	5(3.8%)

SW교육의 필요성에 대해서는 <Table 5>을 볼 때 다수의 학생들도 동의하는 것으로 드러났다.

<Table 6> SW education using tool (N=82)

Tools	Students
EPL(Scratch, entry etc)	52(63.4%)
Physical Computing	19(23.2%)
Unplugged	6(7.3%)
Computer language	4(4.9%)

SW교육을 받아본 적 있는 학생들(N=82) 대상으로 SW교육에서 주로 받아본 수업 도구에 대한 결과는

<Table 6>와 같았다. 주로 EPL(Scratch, entry etc)을 통한 교육이 주를 이루었고, 뒤를 이어 Physical Computing을 활용한 것으로 나타났다. 반면, Computer language의 경우는 가장 낮은 비중을 차지하였다.

<Table 7> Necessity of algorithm execution time training

Necessity	Students
Very likely	14(10.5%)
Somewhat likely	24(18%)
Neutral	78(58.6%)
Somewhat unlikely	8(6%)
Very unlikely	9(6.8%)

알고리즘 수행시간에 대한 교육의 필요성에 대해서 <Table 7>와 같았다. '모르겠다'라고 답한 학생이 과반수이었는데, 학생들에게는 '알고리즘 수행시간'에 대한 개념이 다소 어렵거나 생소하였을 것이라고 추측된다. 또한 긍정응답(28.5%)이 부정응답(12.8%)에 비해 더 높은 비중을 차지하였음을 알 수 있다.

3.3.2 프로그램 설계 및 개발

파이썬을 활용한 SW교육 프로그램을 설계하기 위해서 교육 대상자의 학습수준 및 프로그래밍의 난이도를 고려하여 구성하였다. 특히 탐색알고리즘 중에서는 가장 기초가 되는 선형탐색과 초등학생들도 이해하기 쉬운 이진탐색을 선택하여 수행시간을 비교하고 분석할 수 있도록 교재를 구성하였다. 학습내용은 아래의 <Table 8>과 같다.

<Table 8> The Theme of education program

Hour	Learning Theme
1-7	<ul style="list-style-type: none"> • Orientation • Learning Python basic functions • Input of pre-test
9-14	<ul style="list-style-type: none"> • Print, Input code practice • Practice random code • Practice arithmetic, condition code
15-21	<ul style="list-style-type: none"> • Utilize code we have learned • Practice list code
22-28	<ul style="list-style-type: none"> • Understanding Linear search • Understanding Binary search
29-35	<ul style="list-style-type: none"> • Creating a project plan for individual projects • Creating individual project objects.
36-42	<ul style="list-style-type: none"> • Present your individual project work • Input of post-test

파이썬과 같은 텍스트형 프로그래밍 언어는 처음 접하는 학생이 많기 때문에 파이썬의 기초 문법을 1일 차에 집중적으로 학습할 수 있도록 구성했다. 2일 차에는 다방면으로 활용이 가능한 랜덤 코드, 산술 코드, 관계 코드 등을 학습하도록 구성하고 3일 차에는 그동안 배웠던 코드 등을 활용해보고 연습해보는 시간을 가지도록 구성했다. 또한 탐색 알고리즘의 기초가 되는 리스트를 학습하도록 하였다. 4일 차에 탐색 알고리즘 중에서 선형 탐색과 이진 탐색을 비교해보고 어떤 방식으로 작동하는지에 대해서 살펴보고 학생들이 직접 코딩을 해보며, 두 탐색 방법의 차이점과 수행시간 차이를 분석해보게 한다. 5일 차에는 개별 프로젝트 학습법을 활용하여 학습자가 주도적으로 다양한 문제를 해결하고 과제를 수행하도록 하고 6일 차에 발표 및 사후 검사를 하며 교육 프로그램을 종료하게 구성하였다.

특히 알고리즘 수행시간 분석에 초점을 맞추어 1~3일 차에 기본 문법, 조건문, 반복문 등을 학습할 때에도 어떤 방식으로 코딩을 해야 수행시간이 줄어들 지에 대해 고민해보고 같은 결과가 나오지만 수행시간이 다른 코

Q. '컴퓨터'를 100번 출력한다면?

<pre>print('컴퓨터') print('컴퓨터') print('컴퓨터') . . . print('컴퓨터') print('컴퓨터') print('컴퓨터')</pre>	<pre><실행> '컴퓨터' '컴퓨터' '컴퓨터' . . . '컴퓨터' '컴퓨터' '컴퓨터'</pre>
--	---

⇒이 프로그래밍의 문제점은 뭘까요?

Q. while문을 이용해서 '컴퓨터'를 100번 출력하기 위한 알고리즘을 만들어 보시오.

<pre>i = 100 while (): print('컴퓨터') i=i-1</pre>	<pre><실행> '컴퓨터' '컴퓨터' '컴퓨터' . . . '컴퓨터' '컴퓨터' '컴퓨터'</pre>
--	---

(Fig.1) Example of SW educational program 1

드를 비교해보고 분석해보고 더 나은 방법은 없을 지 고민해보는 과정을 거치도록 지도하였다. 4~5일차에 이루어지는 두 가지 탐색프로그램(선형탐색, 이진탐색) 학습에서는 어떠한 상황에서 어떠한 방법이 더 효율적인 지에 대한 답을 스스로 찾아갈 수 있게 지도하였다.

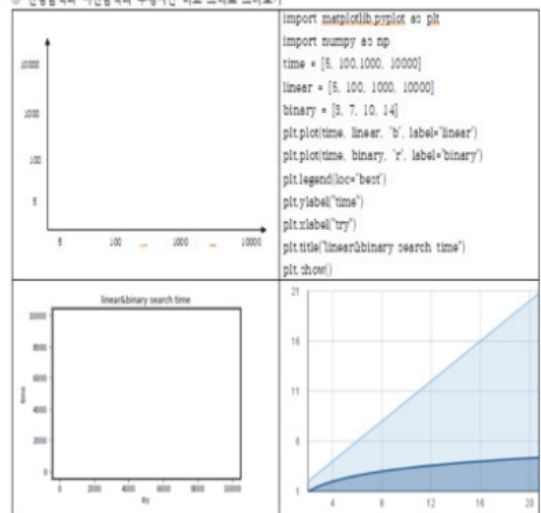
(Fig.1)과 (Fig.2)는 교육 프로그램 수업 중 알고리즘

○ 선형탐색과 이진탐색의 수행시간 비교

	선형탐색		
	최악의 경우	최악의 경우	평균
5개의 숫자 탐색	1번 째 탐색	5번 째 탐색	$(1+2+3+4+5)/5 = 3$ 번 째 탐색
100개의 숫자 탐색			
1000개의 숫자 탐색			
10000개의 숫자 탐색			

	이진탐색	
	최악의 경우	최악의 경우
5개의 숫자 탐색	1번 째 탐색	3번 째 탐색
100개의 숫자 탐색		
1000개의 숫자 탐색		
10000개의 숫자 탐색		

○ 선형탐색과 이진탐색의 수행시간 비교 그래프 그려보기



(Fig.2) Example of SW educational program 2

의 수행시간을 비교해보는 활동과 탐색 방법에 따른 수행시간을 비교 분석해보는 활동을 수업 교재에서 발췌하였다.

여러 가지 탐색 방법이 있지만, 초등학생 수준을 고려하여 가장 기초적인 ‘선형 탐색’과 ‘이진 탐색’의 수행시간을 비교하는 활동으로, 학생들이 직접 최적의 경우와 최악의 경우, 그리고 탐색을 성공하는 평균을 계산해본다. 그리고 이를 파이썬을 활용하여 그래프로 그려보아 시각적으로 비교 및 분석을 한다.

3.4 검사도구

검사도구는 논리적 사고 검사인 GALT를 사용하였다. GALT(Group Assessment of Logical Thinking; Roadrangka et al., 1983)은 1983년 Vantipa Roadrangka, Russell H.yeany, Michael J.Padilla가 공동으로 개발한 논리적 사고력 검사지이다. 총 문항은 21문항으로 각 문항에는 답과 그 이유를 선택할 수 있는 선다형 형식으로 6가지 논리 유형을 측정할 수 있게 구성되었다. 보존논리 4문항, 확률논리 2문항, 상관논리 2문항 그리고 조합논리는 3문항으로 대부분의 문항에는 문제 상황을 설명하는 짧은 문장과 그림이 제시되어있다. GALT개발자들은 시간과 같은 제약이 있을 경우를 대비해 각 논리 유형별로 2문항씩을 뽑아서 총 6유형 12문항으로 구성된 축소본 GALT로 구성하기도 했다. GALT검사지는 각 논리유형들이 태도를 비롯한 과학에 관련된 또 다른 변인을 구성하는 요인과의 관계를 확인하는 연구에서 이용하기 용이하며, 여러 선행연구들에서도 각 논리유형들과 학업성취도, 과학탐구능력, 등과 같은 또 다른 요인들과의 관계를 확인하기 위해 GALT를 이용한다[16][17].

3.5 연구 설계 및 처치

본 연구에서는 도내 초등학생 4~6학년 25명을 대상으로 오리엔테이션, 사전·사후의 논리성 검사, 프로젝트 발표 등을 포함하여 총 6일간 42차시 수업을 진행하였다. 교육은 외부 변인 통제가 보다 용이하도록 하기 위해서 학기 중 주말반 수업 형식을 지양하고 방학 기간에 집중교육의 형태로 실시하였다. 강사 1인이 전체 학습을 진행하고 보조 강사 2인이 학생들에게 도움을 주

었다. 개인별로 데스크탑 컴퓨터를 사용했으며, 파이썬을 별도로 설치하지 않고 웹페이지(Jupyter.org)에서 사용할 수 있도록 환경을 구축하였다.

교육 프로그램의 효과를 검증하기 위해 논리적 사고력 검사인 GALT를 이용하여 사전·사후검사를 실시하였으며 연구절차를 도식화하면 <Table 9>와 같다.

<Table 9> Experimental design

	Pre-test	Treatment	Post-test
class	O1	X	O2
X	: SW education program		
O1 O2	: Pre-Post test(GALT test)		

4. 연구결과

4.1. 교육 프로그램 효과 검증

파이썬의 탐색 알고리즘 기반 수행시간 SW교육으로 인해 학생들의 논리성에 어떠한 영향을 끼쳤는가를 분석하기 위한 검증을 실시하였다.

4.1.1 논리성 검사 정규성 검증

표본의 크기가 30명 이상을 만족하지 못하였기에 검증 전에 실험 집단의 논리성 검사 결과가 정규성을 만족하고 있는지를 정규성 검정을 통해 알아보았다. 정규성 검정은 보편적으로 실시하는 샤피로-윌크(Shapiro-Wilks) 검사를 실시하였고, 그 결과는 아래의 <Table 10>과 같다.

<Table 10> Normality test

Subscale	Descriptive Statistics(N=25)				p
	M	SD	Max	Min	
Preservation	.3600	.8602	3	-1	.001**
Ratio	.5200	1.228	3	-2	.167
Variable control	.1600	.6879	2	-1	.000**
Probability	.0400	1.059	2	-2	.056
Correlation	.1600	.5537	1	-1	.000**
Combination	.4000	.7071	2	-1	.001**

*p<.05 **p<.01

정규성 검정을 한 결과, ‘보존논리’, ‘변인통제논리’,

‘상관논리’, ‘조합논리’는 유의확률(p)이 각각 0.001, 0.000, 0.000, 0.001으로 나타나면서 귀무가설을 기각해서 정규성을 만족하지 않은 것으로 나타났다. 반면, ‘비율논리’, ‘확률논리’에서는 유의확률(p)이 각각 0.167, 0.056으로 유의수준인 .05보다 크기에 귀무가설이 채택되어 정규분포임을 알 수 있었다.

4.1.2 논리성 사전·사후 검사

사전·사후 검사 결과, 논리성의 변화를 알아보기 위해서 정규성을 확보한 요소들은 대응표본 t검정을 실시하고, 정규성을 확보하지 못한 요소들은 비모수검정 중 하나인 Wilcoxon의 부호-순위 검정을 실시하였다.

<Table 11> Changes in Logical Thinking(Paired T-test)

Subscale	N	Pre-Test		Post-Test		t	p
		M	SD	M	SD		
Ratio	25	2.56	1.895	3.08	2.178	-2.116	.045*
Probability	25	1.16	.943	1.20	.764	-.189	.852

*p<.05 **p<.01

대응표본 t검정 결과, ‘비율논리’의 평균점수는 2.56에서 3.08로 0.52점 상승하였으며, 유의확률은 0.045로 평균점수에 있어 유의미한 상승이 있음을 알 수 있다. 반면에 ‘확률논리’의 경우 평균 1.16에서 1.20으로 0.04점 상승하였으나 유의확률이 0.854로 유의미하지 않은 것으로 나타났다.

<Table 12> Changes in Logical Thinking(Wilcoxon’s test)

Subscale	N	Pre-Test		Post-Test		Z	p
		M	SD	M	SD		
Preservation	25	2.88	1.054	3.24	.879	-2.000b	.046*
Variable control	25	1.48	1.418	1.64	1.497	-1.155b	.248
Correlation	25	.240	.436	.400	.577	-1.414b	.157
Combination	25	1.20	.707	1.60	.645	-2.500b	.012*

*p<.05 **p<.01, b. 음의 순위를 기준으로

본 <Table 12>에서는 정규성을 만족하지 못한 요소들에 대해 부호-순위 검정을 한 결과로 ‘보존논리’의 평균점수는 2.88에서 3.24로 0.36점 상승하였고 유의확률이

.046으로 나타났다. ‘조합논리’의 평균점수는 1.20에서 1.60으로 0.40점 상승하였으며 유의확률은 .012로 나타났다. 즉, ‘보존논리’와 ‘조합논리’에서는 평균점수가 유의미하게 상승했음을 알 수 있다. 반면, ‘변인통제논리’의 평균점수는 1.48에서 1.64로 0.16점 상승하고 ‘상관논리’의 평균점수는 0.24에서 0.4로 0.16점 상승하였으나 p<.05를 만족하지 못하여 유의미하지 않은 것으로 나타났다.

4.2 연구 결과 분석

논리적 사고 검사(GALT) 결과 중 정규성을 확보한 논리성 요소들은 대응표본 t검정을 실시하였고, 정규성을 확보하지 못한 논리성 요소들에 대해서는 wilcoxon의 부호 순위 검정을 실시하였다.

정규성을 만족한 두 가지 논리성 요소에 대해 대응표본 t검정 결과 ‘비율논리’에서 유의미한 향상이 있었고, 정규성을 만족하지 못한 네 가지 논리성 요소에 대해 부호-순위 검정을 한 결과 ‘보존논리’, ‘조합논리’에서 유의미한 향상을 보였다. 정리하자면 탐색 알고리즘을 기반으로 한 교육 프로그램을 통해 GALT 논리적 사고 검사의 논리유형에서 확률논리, 변인통제논리, 상관논리의 경우에는 평균점수는 향상되었으나 유의 확률을 충족하지는 못하였으나 비율논리, 보존논리, 조합논리에서는 유의미한 향상을 가져왔다.

이번 연구를 통해 파이썬을 활용한 탐색 알고리즘 기반 수행시간 분석 SW교육이 초등학생의 논리성을 향상에 긍정적인 영향을 줄 수 있다는 것과 파이썬을 활용한 SW교육이 초등학생에게 충분히 활용될 수 있음을 알 수 있었다.

5. 결론

본 연구에서는 ○○도내에 있는 초등학생을 대상으로 논리성을 향상시키기 위해 파이썬을 활용하여 탐색 알고리즘을 기반으로 한 교육 프로그램을 선보이고 ADDIE 모형의 개발 단계에 따라서 교육 프로그램을 개발하고 적용해보았다. 교육을 진행하면서 외부의 변인 통제가 보다 용이하고 연구의 신뢰성을 높이기 위하여 학기 중

주말반 수업 형식을 지양하고 방학 기간을 활용하여 총 6일, 42차시 집중 프로젝트 교육을 실시하였다. 또한 배운 내용 및 앞으로 배울 내용과 관련된 과제를 매일 제시하고 수행하도록 지도하여 짧은 기간이지만 밀도 있는 학습을 할 수 있게 프로젝트를 진행하였다. 논리성 사전 사후 검사를 해본 결과, 본 연구에서 개발한 교육 프로그램을 통한 SW교육이 초등학생의 논리성에 긍정적인 영향을 줄 수 있다는 것을 알 수 있었다.

하지만, 본 연구에서 실험집단은 상관연구에 필요한 30명 이상의 참여자를 투입하지 못하였기 때문에 일반화하는 데에는 한계가 있다. 그리고 본 연구에서 개발한 교육 프로그램에 비교집단 없이 실험집단에만 투입하였기 때문에 논리성에 긍정적인 영향을 미친다는 점이 본 연구에서 개발한 교육 프로그램의 영향 때문인지 상관관계를 분석할 수 없다는 문제가 있다. 또한 한정된 시간으로 인하여 탐색 알고리즘 교육에서 선형 탐색과 이진 탐색만을 대상으로 수행시간 비교 및 분석을 하였는데, 학생들이 직접 구현에 있어 한계점을 가지고 있었다. 추후의 연구에서는 30명 이상 다수의 참여자를 대상으로 실험집단과 비교집단을 구성하고 연구결과에 대해서 각 요인들 간의 상관관계를 비교, 분석할 필요가 있다.

참고문헌

- [1] Computer Language Dictionary Compilation Committee(2005). *Computer Internet IT Dictionary*. Iljin publishing co.
- [2] Fisher, R.(ed.).(1987). *problem solving in Primary school*. Oxford: Basil Blackwell Ltd. pp. 1-33
- [3] Gwak, Byeong Seon(1985). *A Study on the Development of CAI Program for Improving Logical thinking*. Seoul: Korea Educational Development Institute.
- [4] Jeoung, Inkee (2002), *Development of S/W Component for Search Algorithm Education*.
- [5] Jung, Gu Won(2015). *An application and analysis of the visual and textual programming language for the micro controller education*, Seoul National University of Education. Master's Thesis.
- [6] Kang, Ja Young(2018). *An Analysis of the Effect of Elementary School Software Education on Logical Thinking*. Mokpo University of Education. Master's Thesis.
- [7] Kim, H(2016). A Study of the Direction for Developing Software Education Operating Guide. *Korean Association for Learner-centered Curriculum and Instruction*, 16(8), 529-548.
- [8] Kim, Jin Dong & Yang Gwon Woo(2010). The Effect of Algorithm Learning in Real Life Case on Logical Thinking Ability. *Journal of The Korean Association of Information Education*, 14(4), 555-560.
- [9] Kim, Jong-hoon & Kim Jong-Jin(2013). *An introduction to computers*. Seoul: Hanbit publishing co.
- [10] Kim, Jong-hoon(2015). *Flowchart & C language*. Seoul: Hanbit publishing co.
- [11] K. Kim & H. Kim.(2014). A Case Study on Necessity of Computer Programming for Interdisciplinary Education, *Journal of Digital Convergence*, 12(11), 339-348, 2014.
- [12] Ko, Joonkang(2018) *A Study on the Development of Primary Software Education Program using 3D modelling*, Graduate School of Education at Seoul National University of Education
- [13] Ministry of Education(2015). *Software Education Guideline*.
- [14] Moon, Mi Ye(2018). A Study on the Teaching Methods of Python Programming Language for Elementary School Students. *The Journal of Korean association of computer education*, 9(1), 33-41.
- [15] Park, Dae Ryun(2018). *Developing a robotics learning program for elementary school students*. Daegu National University of Education. Master's Thesis.
- [16] Roadrangka, V., Yeany, R.H., and Padilla. M.J.(1983). *The construction and validation of group assesment of logical thinking(GALT)*. Paper presented at the Annual Meeting of the National

Association for Research in Science Teaching, 5p.

- [17] Ryu, Chun Ryol & Lee, Yong Geun(2010). *An Analysis of the Reliability of Group Assessment of Logical Thinking(GALT) using Generalizability Theory*
- [18] Seo, Sung Won(2010). *The Effect of Computational Thinking Ability using TPL and VPL with Robot Programming Education*, Korea National University of Education. Master's Thesis.
- [19] So Hong-ryol (2000). *Logic and Thought*. Seoul: Ewha Womans University publishing co.
- [20] Sung, Jungsook Hyeoncheol Kim(2015). Analysis on the International Comparison of Computer Education in Schools. *The Journal of Korean association of computer education*, 18(1), 45-54.
- [21] Wing, J. M.(2008). *Computational thinking and thinking about computing*. Royal Society of London. Philosophical Transactions 366(1881), 3717-3726.
- [22] Yoo, Jin Ah(2008). *A study on the education of programing language by using the open source software python*. Dankook University of Education. Master's Thesis.
- [23] Yoo, In Hwan(2018). *The Design of SW Education for Elementary School Using Python and Robots*. Korean Association of Information Education, 149-155.
- [24] Ministry of Science(2014), ICT and Future Planning. <http://www.msip.go.kr>
- [25] Park, Eung Yong(2018). Jump To Python 2nd edition. <http://wikidocs.net/mybook/read/page?pageid=4>
- [26] Python Software Foundation(2017). Python about. <https://www.python.org/about/>
- [27] Software central society(2018). What is SW education for elementary and middle school students? (http://www.software.kr/um/um03/um0307/um030702/um030702List.do?s_gubun=BK281#this)

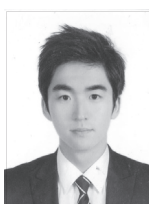
저자 소개

양영훈



2014. 제주대학교 과학교육학부 컴퓨터교육과 박사 수료
관심분야: 컴퓨터교육
e-mail : atriple1981@naver.com

공기표



2015 ~ 현재 제주대학교 초등컴퓨터교육전공 석사과정
관심분야: 컴퓨터교육, SW교육
e-mail : kkcc89@naver.com

김종훈



1999~현재 제주대학교 초등컴퓨터교육전공 교수
관심분야: 컴퓨터교육
e-mail : jkim0858@jejunu.ac.kr