



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

수행시간 파이썬 정렬 알고리즘을 중심으로
중심의 초등 알고리즘 교육
프로그램 개발 및 적용

문우종

2019

석사학위논문

수행시간 중심 초등 알고리즘 교육 프로그램
개발 및 적용: 파이썬 정렬 알고리즘을 중심으로

Development and Application of Algorithm Execution
Time Analysis Education Program for Elementary
School Students: Focusing on Python Sort Algorithm

제주대학교 교육대학원

초등컴퓨터교육전공

문 우 종

2019년 8월

석사학위논문

수행시간 중심 초등 알고리즘 교육 프로그램
개발 및 적용: 파이썬 정렬 알고리즘을 중심으로

Development and Application of Algorithm Execution
Time Analysis Education Program for Elementary
School Students: Focusing on Python Sort Algorithm

제주대학교 교육대학원

초등컴퓨터교육전공

문 우 종

2019년 8월

수행시간 중심 초등 알고리즘 교육 프로그램
개발 및 적용: 파이썬 정렬 알고리즘을 중심으로

Development and Application of Algorithm Execution
Time Analysis Education Program for Elementary
School Students: Focusing on Python Sort Algorithm

지도교수 김 종 훈

이 논문을 교육학 석사학위 논문으로 제출함

제주대학교 교육대학원

초등컴퓨터교육전공

문 우 종

2019년 5월

문 우 종의

교육학 석사학위 논문을 인준함

심사위원장 김 중 우



심사위원 박 남 제



심사위원 김 중 훈



제주대학교 교육대학원

2019년 6월



목 차

국문 초록	iv
I. 서론	1
1. 연구의 필요성	1
2. 연구 내용	2
3. 연구 절차	3
II. 이론적 배경	4
1. 파이썬	4
2. 정렬 알고리즘	4
3. 소프트웨어 교육	5
4. 논리적 사고력	6
5. 컴퓨팅 사고력	6
6. 수행시간	7
7. 선행연구 분석	7
III. 수행시간 중심 초등 알고리즘 교육 프로그램 개발	9
1. 요구 분석	9
2. 요구 분석 의사 결정	12
3. 교육 프로그램 설계	13
4. 교육 프로그램 개발	14
IV. 연구의 적용 및 결과 분석	23
1. 연구 대상	23
2. 검사 도구	23
3. 연구 결과	25
4. 연구 결과 분석	28
V. 결론 및 제언	30
참고 문헌	31
ABSTRACT	34
부 록	35

표 목 차

〈표 I-1〉 Dick&Carey 모형에 따른 교육 프로그램 설계 과정	3
〈표 III-1〉 소프트웨어 교육(코딩교육)에 대한 인지도	9
〈표 III-2〉 소프트웨어 교육(코딩교육)에 대한 관심도	9
〈표 III-3〉 소프트웨어 교육(코딩교육) 관련 수업 경험	9
〈표 III-4〉 소프트웨어 교육(코딩교육) 관련 수업 방법/도구	9
〈표 III-5〉 소프트웨어 교육(코딩교육) 시 프로그래밍 언어 지도 의향	10
〈표 III-6〉 알고리즘에 대한 인지도	10
〈표 III-7〉 소프트웨어 교육(코딩교육)에서 알고리즘의 중요성	11
〈표 III-8〉 학생들에게 알고리즘 수행시간에 대한 교육이 필요한지 여부	11
〈표 III-9〉 알고리즘 수행시간 교육 시 생길 수 있는 어려움	11
〈표 III-10〉 알고리즘 수행시간 교육 시 교수 내용	11
〈표 III-11〉 알고리즘 수행시간 교육 시 활용 도구	11
〈표 III-12〉 교육 프로그램 내용 설계	13
〈표 III-13〉 수행시간 중심 파이썬 정렬 알고리즘 교육프로그램 내용	14
〈표 III-14〉 정렬 알고리즘을 통한 수행시간 분석 교사용 과정안(1)	21
〈표 III-15〉 정렬 알고리즘을 통한 수행시간 분석 교사용 과정안(2)	22
〈표 IV-1〉 교육기부 프로그램 지원자 학년과 성별	23
〈표 IV-2〉 논리적 사고력 검사 실험 설계	24
〈표 IV-3〉 컴퓨팅 사고력 검사 실험 설계	24
〈표 IV-4〉 논리적 사고력 검사에 대한 정규성 검정 결과	25
〈표 IV-5〉 컴퓨팅 사고력 검사에 대한 정규성 검정 결과	26
〈표 IV-6〉 논리적 사고력 사전·사후 검사 결과(대응표본 t검증)	27
〈표 IV-7〉 논리적 사고력 사전·사후 검사 결과(윌콕슨 부호-순위 검정)	27
〈표 IV-8〉 컴퓨팅 사고력 사전·사후 검사 결과(윌콕슨 부호-순위 검정)	28

그림 목 차

[그림 Ⅲ-1] 「언플러그드 활동을 통한 정렬 개념 이해」 학생용 교재	…	16
[그림 Ⅲ-2] 「선택 정렬 프로그래밍」 학생용 교재	……………	17
[그림 Ⅲ-3] 「삽입 정렬 프로그래밍」 학생용 교재 일부	……………	18
[그림 Ⅲ-4] 「버블 정렬 프로그래밍」 학생용 교재 일부	……………	19
[그림 Ⅲ-5] 「정렬 알고리즘 수행시간 분석」 학생용 교재	……………	20

국 문 초 록

수행시간 중심 초등 알고리즘 교육 프로그램 개발 및 적용: 파이썬 정렬 알고리즘을 중심으로

문 우 종

제주대학교 교육대학원 초등컴퓨터교육전공
지도교수 김 종 훈

본 연구는 초등학생의 사고력 신장을 위한 교육 방법으로 파이썬 정렬 알고리즘을 중심으로 수행시간 중심 초등 알고리즘 교육 프로그램을 개발하여 적용한 후 그 효과를 검증하였다. 교육 프로그램은 현직 초등학교 교사 100명을 대상으로 실시한 사전 요구 분석 결과를 바탕으로 하여 개발하였다. 개발한 교육 프로그램의 효과를 검증하기 위해 실시한 초등학생 교육기부 프로그램의 지원자 표집에 의한 지원자 표본 25명을 대상으로 6일간 42차시를 진행하였다. 한국교육개발원에서 개발한 '논리적 사고 검사(GALT)', 비버챌린지 2017 검사를 활용하여 사전·사후 검사 결과를 통해 교육적 효과를 분석하였으며, 분석 결과 파이썬 정렬 알고리즘을 중심으로 한 수행시간 중심 초등 알고리즘 교육 프로그램이 초등학생의 논리적 사고력, 컴퓨팅 사고력 향상에 효과적인 것으로 나타났다.

주요어 : 수행시간, 파이썬, 정렬 알고리즘, 논리적 사고력, 컴퓨팅 사고력

I. 서 론

1. 연구의 필요성

2016년 3월 인공지능(AI) 알파고(AlphaGo)와 세계랭킹 1위 바둑기사 이세돌 9단의 바둑 대국에서 인공지능이 승리를 거둔 소위 ‘알파고 쇼크’ 이후, 전 세계적으로 인공지능에 대한 관심이 높아지고 있다. 알파고 이후 페이스북, 아마존, 마이크로소프트, 애플, IBM 등 세계적 IT기업들이 앞다투어 저마다의 인공지능 기술을 내놓기 시작하였으며, 전문가들의 용어라고 여겨졌던 인공지능이 점차 일상적인 용어로 자리 잡고 있다(맹미선, 2017). 제46차 세계경제포럼(다보스포럼)에서는 ‘4차 산업혁명에 따른 미래 일자리 변화 전망’ 보고서를 통해 인공지능의 발달로 2020년까지 세계적으로 사무직 관련 일자리 717만개가 사라지고, 컴퓨터 분야 일자리 210만개가 생겨날 것으로 전망하였다(Schwab, 2016).

4차 산업혁명이 화두가 되며 미래사회에 대비한 교육이 어떻게 이루어져야 하는지에 대한 논의도 활발하게 제시되고 있다. 2017년 국제미래학회에서 발간한 제4차 산업혁명 시대 대한민국 미래보고서에 따르면 교육의 방향을 첨단 기술의 사용화에 대비하여 다양성, 창의성, 유연성을 강화하는 방향으로 변화시켜야 한다. 이에 새롭게 교육과정에 반영되는 코딩(coding) 교육은 단순히 프로그램을 만들어 내는 기술적인 코더(coder) 양성이 아닌, 사고할 수 있는 역량인 컴퓨팅 사고력(Computational thinking)을 길러주는 것이 목적이 되어야 할 것이다(김대식, 2017).

우리나라도 4차 산업혁명의 흐름에 따라 2015 개정 교육과정에서 소프트웨어 교육을 도입하였고, 초등학교에서는 2019년부터 6학년을 대상으로 실과 교과에서 소프트웨어 교육이 시작되었다(교육부, 2015). 그러나 학생들에게 실시되고 있는 소프트웨어 교육은 대다수가 활동지 위주의 언플러그드나 기초적인 블록형 프로그래밍 언어를 사용하는 데 그치고 있다. 언플러그드 교육은 별도의 기자재 없이 간편하게 실시할 수 있으며, 블록형 프로그래밍 언어는 이해와 생산에 있어 다루기가 쉽기 때문에 보인다(소미현, 2017). 실제로 제주도내 5~6학년 초등교사 100명을 대상으로 한 설문 및 고학년 학생 20명을 대상으로 한 면담 조

사 결과, 학생 대상 소프트웨어교육(코딩교육)에서 주로 활용하는 방법/도구는 블록형 프로그래밍언어가 52.2%, 언플러그드가 34.3%, 피지컬 컴퓨팅 도구가 13.4%를 차지하였으며 텍스트 프로그래밍 언어(파이썬, C, 자바 등)를 사용하는 경우는 0%에 불과하였다.

언플러그드 교육은 기초 개념을 파악하는 데는 유용하나 학생들이 사고할 수 있는 역량을 길러주기에는 부족하며, 블록형 프로그래밍 언어는 간단하고 조작성이 쉽지만 다른 프로그래밍 언어와 비교하면 사용할 수 있는 기능이 한정되어 있다. 이에 반해 텍스트 프로그래밍 언어는 블록형 프로그래밍 언어에 비해 정보과학적 사고능력 향상에 도움이 된다는 연구 결과가 있어 미래 사회에 대비한 사고력을 길러주는 교육에는 가장 적합한 교육 방법이다(Myers, 1990).

이에 본 연구에서는 텍스트 프로그래밍 언어 중 학생들에게 프로그래밍을 가르치기 쉬우며 효과가 뛰어나다고 평가되는 파이썬 도구로 선정하여, 이를 활용한 소프트웨어 교육 프로그램을 개발하여 적용하였다. 개발한 교육 프로그램은 사전 요구분석을 통해 주제 및 교육 내용을 선정하였으며 초등학교 4~6학년 학생들 중 지원자표본(volunteer sample) 25명의 학생을 대상으로 42차시 가량 투입하였다. 요구 분석 결과 주제로 선정된 정렬 알고리즘, 수행시간을 핵심 키워드로 잡아 개발한 파이썬 교육 프로그램 실시 후 학생들의 논리적 사고력과 컴퓨팅 사고력의 향상 정도를 검사를 통해 각각 비교하였으며, 논리적 사고력 검사 도구로는 GALT(Group Assessment of Logical Thinking)를, 컴퓨팅 사고력 검사 도구로는 비버챌린지(Bebras Challenge) 2017을 사용하였다.

2. 연구 내용

파이썬 정렬 알고리즘을 중심으로 한 수행시간 중심 초등 알고리즘 교육 프로그램을 개발 및 적용하기 위해 수행한 연구 내용은 다음과 같다.

첫째, 기존에 이루어진 수행시간 중심 알고리즘 교육 연구와 파이썬을 초등 대상으로 적용한 연구를 분석 및 보완하여 초등학생을 대상으로 한 수행시간 중심 알고리즘 교육 프로그램을 개발한다.

둘째, 교사 및 학생을 대상으로 한 사전 요구 분석을 통해 수행시간, 파이썬, 정렬 알고리즘과 관련된 주제 안에서 초등학생에게 적합한 내용을 선정하여 교육 프로그램을 구성한다.

셋째, 개발한 교육 프로그램을 초등학생 4~6학년을 대상으로 실제로 적용해 보고, 교육 전후의 사전·사후 검사를 통해 그 교육적 효과를 검증한다.

3. 연구 절차

본 연구에서는 교수 체제 설계 과정의 일반적 형태를 나타내어 가장 널리 활용되고 있는 Dick & Carey의 ADDIE 모형의 절차에 따라 교육 프로그램을 <표 I-1>과 같이 연구 및 개발하였다(이지연, 2008).

<표 I-1> Dick & Carey 모형에 따른 교육 프로그램 설계 과정

단계	프로그램 연구 및 개발 절차
분석 (Analysis)	<ul style="list-style-type: none"> · 요구 분석(초등교사 100명 대상 설문) - 소프트웨어 교육 인지도, 교수도구 등 11개 문항 · 학습자 분석(5~6학년 20명 대상 면담) - 소프트웨어 교육 수업도구, 알고리즘 인지도 2개 문항
설계 (Design)	<ul style="list-style-type: none"> · 수행목표 명세화 - 수행시간 중심의 초등 알고리즘 교육 프로그램 개발 및 적용 :파이썬 정렬 알고리즘을 중심으로
개발 (Development)	<ul style="list-style-type: none"> · 교수자료 개발 - 42차시 교수학습 과정안, 교재 개발
실행 (Implementation)	<ul style="list-style-type: none"> · 수행시간 중심의 초등 알고리즘 교육 프로그램 개발 후 초등 학생 4~6학년을 대상으로 사전검사 후 교육 실시
평가 (Evaluation)	<ul style="list-style-type: none"> · 사후검사 후 사전검사와 비교하여 교육적 성과 점검 - 논리적 사고력, 컴퓨팅 사고력의 변화 정도 확인

II. 이론적 배경

1. 파이썬(Python)

파이썬(Python)은 1991년 네덜란드 프로그래머인 귀도 반 로섬(Guido van Rossum)이 발표한 고급 프로그래밍 언어로, 플랫폼 독립적이며 인터프리터식, 객체지향적, 동적 타이핑(dynamically typed) 대화형 언어이다(Wiki, 2018).

최근 들어 파이썬은 실무와 교육 양쪽 모두에서 큰 각광을 받고 있다. 2014년 국제 멀티미디어 학술대회인 ACM에서 진행된 조사에 따르면 파이썬은 미국 내 일류 대학에서 프로그래밍 입문 교육에 가장 많이 사용되고 있는 언어이다.(Philip, 2014) 또한 2016년 코드경진대회 서비스를 제공하는 코드이벨은 회원들의 사용 통계에 기반해서 파이썬을 '프로그래밍 인기 순위 1위'로 뽑기도 하였으며(Codeeval, 2016), 프로그래밍 언어 순위를 나타내는 TIOBE에서는 'C++'을 제치고 2018년 9월 3위까지 부상하기도 하였다(Tiobe, 2018).

파이썬은 본래 교육 목적으로 개발된 언어는 아니지만 구문이 간결하여 이해하기 쉽고 프로그래밍의 원리를 익힐 수 있으며, 시각적으로 학습 결과를 확인할 수 있는 등 교육용 프로그래밍 언어로서 요구되는 특징과 부합되어 학생들에게 프로그래밍을 가르치기 좋은 도구이다(유진아, 2008). 실제로 파이썬 입문 서적인 'Think Python'의 저자 앨런 다우니(Allen Downey)는 파이썬을 사용한 프로그래밍 수업은 JAVA를 이용한 수업과 비교했을 때 "학생들이 덜 고생하고, 더 많이 배우고, 더 많은 프로젝트를 즐겁게 수행했다."고 밝혔다(이대현, 2015).

2. 정렬 알고리즘(Sort Algorithm)

정렬(sort)이란 데이터를 일정한 규칙에 따라 재배열하는 것을 의미한다. 데이터의 크기가 작은 수부터 큰 수의 순으로 나열하는 것을 오름차순(ascending order), 큰 수부터 작은 수의 순으로 나열하는 것을 내림차순(descending order)이라고 한다(김종훈, 2013). 정렬을 수행하는 알고리즘은 다양한데, 가

장 대표적인 것으로 선택 정렬, 삽입 정렬, 버블 정렬이 있다. 선택 정렬(selection sort)은 정렬되지 않은 데이터들에 대해 가장 작은 데이터를 찾아 가장 앞의 데이터와 교환해나가는 방식으로 일상생활에서 가장 많이 사용되는 정렬이다. 삽입 정렬(insertion sort)은 아직 정렬되지 않은 임의의 데이터를 이미 정렬된 부분의 적절한 위치에 삽입해 가며 정렬하는 방식이다. 버블 정렬(bubble sort)은 거품 정렬이라고도 하는데, 서로 이웃한 데이터들을 비교하며 가장 큰 데이터를 가장 뒤로 보내며 정렬하는 방식으로 정렬 방법 중 가장 간단한 정렬 방법이다(김민수, 2004).

3. 소프트웨어 교육

소프트웨어 교육은 컴퓨터 과학(computer science)의 이론적 기초를 바탕으로 주어진 문제를 창의적으로 해결하는 것을 의미하며, 학습자는 소프트웨어 교육을 통해 자신이 원하는 것을 상상하여 프로젝트를 창작하고 타인과 아이디어를 공유하는 과정을 통해 창의성을 신장시킬 수 있다(Brennan, 2013).

우리나라에서도 2015 개정 교육과정의 역량 중 하나인 지식정보처리 역량을 신장시킬 수 있도록, 중학교에서는 2018년부터 소프트웨어 교육이 의무화되며, 초등학교에서도 2019년부터 5-6학년 학생들이 소프트웨어 교육을 필수적으로 이수하도록 개정되었다.

초·중등학교에서 이루어지는 소프트웨어 교육은 프로그램 개발 역량보다는 정보윤리의식과 태도를 바탕으로 실생활의 문제를 컴퓨팅 사고로 해결할 수 있도록 하는 것에 역점을 둔다. 또한 소프트웨어 교육은 지식 위주의 교육보다는 수행 위주의 교육을 통하여 디지털 사회의 필수적 요소인 컴퓨팅 사고력의 의미와 중요성을 학습자 스스로 인식하고 그 가치를 확인할 수 있도록 교육 방법을 설계한다. 소프트웨어 교육을 통하여 '컴퓨팅 사고력을 가진 창의·융합인재'를 기르는 것을 목표로 하고 있으며, 초등학교에서는 체험과 활동을 중심으로 건전한 정보윤리의식을 바탕으로 알고리즘과 프로그래밍을 체험하여 실생활의 다양한 문제를 이해할 수 있도록 하는 것이 목표이다(교육부, 2015).

4. 논리적 사고력

논리적 사고력은 연역적 추리와 귀납적 추리로 분류할 수 있으며, 이전에 비해 수준 높은 비판적 사고를 일컫는다. 피아제의 인지 발달 단계에 의하면 초등학교 고학년들은 형식적 조작기의 입문단계에 해당한다. 이 단계의 학습자들은 스스로 자신의 관심과 요구가 무엇인지 파악할 수 있으며, 주어진 문제를 해결하기 위해 필요한 정보를 찾고 선택하여 활용할 수 있도록 문제 해결력을 길러주어야 한다. 이는 형식적 조작기의 논리적 사고력 형성의 필요성을 말해주고 있다(강자영, 2018). 논리적 사고력은 논리적인 규칙에 따라 사고하는 능력으로서, 문제를 구체화하고 전후관계나 인과관계 등을 파악하여 문제를 분석하고 해결하는 능력이라고 정리할 수 있다(김우찬, 2016).

컴퓨터는 사람보다 더 논리적이며 컴퓨터와 소통할 수 있는 컴퓨터 언어는 논리적 사고에 기초하고 있다. 이에 소프트웨어 코딩 교육은 논리적 사고력을 기반으로 일상생활 속에서 만날 수 있는 문제를 해결하는 것이다(김대욱, 2019).

5. 컴퓨팅 사고력

1980년대 시모어 페퍼트(Seymour Papert)가 아동의 절차적 사고력에 대한 연구를 통해 사고 과정으로서 컴퓨팅을 체계화하고 컴퓨팅 기기를 통해 문제를 해결하는 과정에서 컴퓨팅 사고라는 용어를 사용하였으며(류미영, 2015), 2006년 자넷 윙(Jeannette M. Wing) 교수가 컴퓨팅의 기본 개념과 원리를 바탕으로 문제를 효율적으로 해결할 수 있는 사고 능력으로서 컴퓨팅 사고력을 제안하였다. 윙 교수는 모든 학습자가 컴퓨팅 사고력을 모든 사람이 갖추어야 할 핵심 역량으로서 반드시 배우고 학습해야 한다고 주장하였으며, 컴퓨팅 사고력은 컴퓨터 과학의 중심 개념으로서 미국과 영국을 중심으로 컴퓨팅 사고력을 정규 교육 과정에 포함시키기 위한 연구들이 지속적으로 수행되고 있다(함성진, 2014).

6. 수행시간

수행시간은 입력(input)에 따라 알고리즘이 실행되어 출력(output)되기까지 소요되는 시간을 의미한다. 알고리즘의 성능은 수행시간과 필요한 기억 장치의 양에 따라 평가할 수 있는데 대부분 수행 시간이 더 중요한 의미를 지닌다. 알고리즘 수행시간을 분석하여 계산한 결과를 시간 복잡도(Time Complexity)라고 하며, 알고리즘의 핵심이 되는 연산의 횟수를 세어 성능을 파악할 수 있다. 알고리즘 수행시간에 따라 알고리즘 내의 연산 횟수가 크게 달라지기 때문에 효율적인 문제 해결을 위해서는 알고리즘 수행시간에 초점을 두어야 한다(전산용어편찬위원회, 2005). 본 연구에서는 수행시간을 중심으로 주어진 문제를 파이썬으로 해결하는 알고리즘 교육에 초점을 두었다.

7. 선행연구 분석

임민영의 연구에서는 초등학교 컴퓨터교육에서 검색과 정렬 알고리즘의 학습 내용을 재구성해 지도해보고, 검색과 정렬 알고리즘을 초등학생들이 이해할 수 있는지 알아보았다. 그 결과 학습내용을 초등학생들의 수준에 알맞게 가르쳤을 경우 학생들은 충분히 이해할 수 있으며 정의적인 부분 역시 많은 흥미를 느낀 것으로 나타났다. 검색과 정렬 알고리즘을 동시에 다루었기 때문에, 어떤 주제가 컴퓨터 학습에 효과적인가에 대한 연구가 더 필요함을 제언하였다(임민영, 2018). 본 연구에서는 교사 대상 요구 분석 결과를 바탕으로 정렬 알고리즘을 주제로 초등학생들의 수준에 알맞은 소프트웨어 교육 교재를 개발하여, 수행시간 중심의 파이썬 정렬 알고리즘 교육이 초등학생의 사고력에 미치는 영향을 연구하였다.

박대륜의 연구에서는 초등학생을 위한 로봇 활용 파이썬 학습 프로그램을 개발하여 초등학교 6학년을 대상으로 적용하였다. 그 결과 로봇 활용 파이썬 학습 프로그램을 통해 컴퓨팅 사고력이 향상되었으며, 초등학교 학생들에게도 텍스트 기

반 프로그래밍 언어를 사용한 학습 프로그램이 유의미함을 확인하였다(박대륜, 2018). 본 연구에서는 텍스트 기반 프로그래밍 언어인 파이썬을 활용한 학습 프로그램이 초등학생들에게 실제로 유의미한지, 컴퓨팅 사고력과 더불어 논리적 사고력 향상에도 영향을 주는지 연구하였다.

배성종의 연구에서는 파이썬 프로그래밍을 통한 실제적인 문제 해결을 적용하였다. 연구 결과 프로그래밍의 기본 개념과 구조의 원리를 학습하는 데 성취도가 향상되었으며, 나아가 컴퓨팅 사고 능력을 향상시키는 데 도움이 된다고 하였다. 또한 문법에 대한 주입식 학습보다는 실제적인 문제 상황의 해결 과정을 학습 설계하여 적용함으로써 학습의 참여와 성취도를 높였다(배성종, 2018). 이에 본 연구에서는 파이썬의 기초 문법들을 예제들을 통해 익히고, 실제적인 문제들(리스트를 활용해 퀴즈 만들어보고 서로 풀어보기, 정렬 알고리즘을 통해 n번째로 키가 큰 학생 찾기 등)을 해결해보며 학습 참여와 성취도를 높이도록 교재를 설계하였다.

문미예의 연구에서는 2015 개정 교육과정의 성취기준에 맞게 파이썬을 초등학생들에게 적용한 프로그래밍 교육 방안을 개발하였다. 연구 결과 파이썬을 통해 알고리즘의 구성과 절차, 프로그래밍의 논리 구조를 익히고 텍스트 프로그래밍으로 긍정적 전이가 가능할 것으로 보았다(문미예, 2018). 다만 이러한 교육 방안을 실제 학생들에게 적용해보지 않은 한계가 있어, 본 연구에서는 실제 초등학교 고학년 학생들을 대상으로 파이썬을 활용한 프로그래밍 교육을 직접 실시하여 논리적 사고력, 컴퓨팅 사고력 변화에 미치는 영향을 분석하였다.

텍스트 프로그래밍 언어인 파이썬이나 정렬 알고리즘 교육을 초등학생들에게 적용한 선행 연구 사례들을 정리해본 결과 텍스트 프로그래밍 언어인 파이썬과 정렬 알고리즘은 초등학생 대상으로 교육할 수 있으며, 학생들의 사고 기능 향상에 긍정적인 영향을 가져온 것으로 나타났다. 이에 본 연구에서는 파이썬을 활용한 정렬 알고리즘의 수행시간 주제로 잡아 각 알고리즘의 수행시간을 비교·분석하는 교육 프로그램을 개발하여 학생들에게 적용해보고, 교육 전후 학생들의 논리적, 컴퓨팅 사고력 향상에 영향을 주는지 결과를 분석하였다.

Ⅲ. 수행시간 중심 초등 알고리즘 교육 프로그램 개발

1. 요구 분석

프로그램 설계 및 개발을 위해 학습자들의 요구를 분석하고자 초등학교 교사 100명, 초등학교 5~6학년 학생 20명을 대상으로 관련 설문을 실시하였다.

가. 설문 대상에 대한 정보 관련 설문

〈표 Ⅲ-1〉 소프트웨어 교육(코딩교육)에 대한 인지도

	매우 낮다	낮다	보통이다	조금 높다	매우 높다
교사 수	3	12	14	33	38

〈표 Ⅲ-2〉 소프트웨어 교육(코딩교육)에 대한 관심도

	매우 낮다	낮다	보통이다	조금 높다	매우 높다
교사 수	5	5	15	48	27

〈표 Ⅲ-3〉 소프트웨어 교육(코딩교육) 관련 수업 경험

	있다	없다
교사 수	67	33

〈표 Ⅲ-4〉 소프트웨어 교육(코딩교육) 관련 수업 방법/도구

	언플러그드	피지컬 컴퓨팅	교육용 프로그래밍 언어	프로그래밍 언어
교사 수	44	18	38	0
학생 수	10	3	6	1

〈표 III-5〉 소프트웨어 교육(코딩교육) 시 프로그래밍 언어 지도 의향

	매우 낮다	낮다	보통이다	조금 높다	매우 높다
교사 수	21	19	30	17	13

설문 대상인 교사 100명은 〈표 III-1〉, 〈표 III-2〉에 의하면 소프트웨어 교육에 대한 인지도가 대부분 높으며 관심도 높았다. 〈표 III-3〉의 실제로 수업해 본 경험은 교사 3명 중 2명이 해당하였으며, 소프트웨어 교육에서 주로 활용하는 수업 방법 결과는 〈표 III-4〉와 같다. 〈표 III-4〉에 의하면 교사는 소프트웨어 교육 방법 중 언플러그드에 대한 선호도가 가장 높고 프로그래밍 언어가 가장 낮았으며, 프로그래밍 언어 지도 의향이 있는 교사는 〈표 III-5〉와 같이 30% 정도로 나타났다.

초등학교 5~6학년 학생들을 대상으로는 소프트웨어 수업 시 주로 사용하는 도구를 면담하여 〈표 III-4〉에 함께 정리하였으며, 그 결과 언플러그드 교육 경험이 가장 많았으며 프로그래밍 언어를 경험한 학생들은 한 명에 불과하였다.

위의 내용으로 볼 때 소프트웨어 교육에 대한 인지도와 관심도가 높은 초등학교 교사들도 대부분 언플러그드, 교육용 프로그래밍 언어로 수업을 지도하고 있으며 프로그래밍 언어 지도 의향은 높지 않았다. 소프트웨어 교육 시 프로그래밍 언어로 교육을 받아본 경험이 있는 학생이 거의 없다는 학생 면담 결과도 이를 뒷받침하고 있다.

나. 알고리즘 교육 프로그램 관련 설문

〈표 III-6〉 알고리즘에 대한 인지도

	매우 낮다	낮다	보통이다	조금 높다	매우 높다
교사 수	6	10	20	36	28
학생 수	9	4	6	0	1

〈표 III-7〉 소프트웨어 교육(코딩교육)에서 알고리즘의 중요성

	매우 낮다	낮다	보통이다	조금 높다	매우 높다
교사 수	2	2	19	41	36

〈표 III-8〉 학생들에게 알고리즘 수행시간에 대한 교육이 필요한지 여부

	전혀 그렇지 않다	그렇지 않다	보통이다	조금 그렇다	매우 그렇다
교사 수	0	2	19	53	26

〈표 III-9〉 알고리즘 수행시간 교육 시 생길 수 있는 어려움

	교육내용이 광범위함	초등학생이 이해하기 어려움	교육에 적합한 도구의 부재	학생들의 흥미 부족	많은 시간이 필요함
교사 수	10	66	10	3	11

〈표 III-10〉 알고리즘 수행시간 교육 시 교수 내용

	자료 구조	정렬	탐색	알고리즘 이론
교사 수	36	68	33	10

〈표 III-11〉 알고리즘 수행시간 교육 시 활용 도구

	언플러그드	피지컬 컴퓨팅	교육용 프로그래밍 언어	프로그래밍 언어
교사 수	43	17	33	7

소프트웨어 교육에서 알고리즘이 차지하는 중요성에 대한 질문에 대다수의 교사는 <표 III-7>을 볼 때 알고리즘이 중요하다고 생각하고 있으며, <표 III-8>처럼 수행시간에 대한 교육도 꼭 필요하다고 응답하였다. 알고리즘 수행시간에 대한 교육을 할 때 어려운 점에 대한 조사 결과는 <표 III-9>와 같으며, 초등학생들이 이해하기 어렵다고 생각하는 교사들이 많았고 가르칠 시간 부족, 적절한 교수도구의 부재가 뒤를 이었다. 효율적인 알고리즘 작성을 위해 알고리즘의 수행시간을 교육한다면 어떤 내용을 교육할지에 대한 조사 결과는 <표 III-10>과 같다. 교사들은 정렬, 탐색, 데이터 구조, 알고리즘 이론(백트래킹, 분할정복법 등) 순으로 제재를 선호하는 것으로 나타났다. <표 III-11>에서는 알고리즘 수행시간을 교육할 때 사용할 수업 도구를 조사하였으며 언플러그드, 교육용 프로그래밍 언어, 피지컬 컴퓨팅, 프로그래밍 언어 순으로 응답하였다.

2. 요구분석 의사결정

요구 분석 결과를 분석한 결과 다음과 같은 요구를 도출할 수 있다.

첫째, 현행 소프트웨어 교육은 대부분 언플러그드나 교육용 프로그래밍 언어 교육에 그치고 있다. 이런 결과는 <표 III-4>에서 텍스트 프로그래밍 언어를 사용하여 교육해 본 경험이 있는 교사가 조사 대상 100명 중 0명, 학생은 20명 중 1명이라는 것에서 찾아볼 수 있다. 물론 블록형 프로그래밍 언어를 활용한 교육은 간단하고 조작성이 쉬워 학생들의 이해를 돕기 쉬우나, 앞서 정리했듯 가르칠 수 있는 내용이 한정적이고 텍스트 프로그래밍 언어에 비해 사고능력 향상도가 저조하다. 이에 본 교육에서는 텍스트 프로그래밍 언어 중 학생들에게 프로그래밍을 가르치기 좋은 파이썬을 수업 도구로 선정하였다.

둘째, 소프트웨어 교육에서 대다수의 교사가 알고리즘의 중요성을 인지하고, 알고리즘 수행시간에 대한 교육이 필요하다는 데 공감하고 있다. 다만 알고리즘의 수행시간에 대한 교육은 내용의 난이도가 높다고 생각하며, 이를 가르칠 시간이 부족하여 제대로 된 교육이 이루어지지 못했다. 이에 본 연구에서는 초등학생들이 이해할 수 있도록 알고리즘의 수행시간의 개념을 실제 사례를 통해 보다 쉽

게 소개하고, 프로그램을 작성할 때 알고리즘 수행시간을 단축시킬 수 있는 방법을 찾아보는 과정에서 학생들의 논리적 사고력 향상을 꾀하는 데 중점을 두었다.

셋째, 알고리즘 수행시간 교육에 가장 적절한 수업 내용으로는 정렬이 가장 적절하다고 평가하고 있으며, 실제로 정렬 알고리즘을 통한 교육도 많이 이루어지고 있다. 이에 파이썬 정렬 알고리즘을 기반으로 한 수행시간 중심의 초등 알고리즘 교육을 통해 학생들의 논리적 사고력과 컴퓨팅 사고력을 향상시킬 수 있도록 본 프로그램을 구성하였다.

3. 교육 프로그램 설계

교육 프로그램은 학습자들의 수준과 프로그래밍 난이도를 고려하여 파이썬 정렬 알고리즘을 중심으로 수행시간 중심의 초등 알고리즘 교육에 초점을 두어 학습 내용을 <표 III-12>와 같이 구성하였다. 오리엔테이션 및 사전 검사(1~3차시), 파이썬 설치 및 파이썬 UI 소개(4~7차시), 변수와 연산자(8~11차시), 선택 구조와 논리 연산(12~14차시), 반복 구조와 리스트(15~18차시), 선택 정렬, 삽입 정렬, 버블 정렬(19~21차시), 정렬 알고리즘을 통한 알고리즘 수행시간 분석(22~28차시), 개인별 프로젝트 계획서 작성 및 제작(29~35차시), 개인별 프로젝트 과제 발표 및 사후 검사(36~42차시)로 진행하였다.

<표 III-12> 교육 프로그램 내용 설계

차시	프로그램 내용
1	오리엔테이션 및 사전 검사
2	파이썬 설치 및 파이썬 UI 소개
3~4	변수와 연산자
5~7	선택 구조와 논리 연산
8~14	반복 구조와 리스트
15~21	선택 정렬, 삽입 정렬, 버블 정렬
22~28	정렬 알고리즘을 통한 알고리즘 수행시간 분석
29~35	개인별 프로젝트 계획서 작성 및 프로그램 제작
36~42	개인별 프로젝트 과제 발표 및 사후 검사

4. 교육 프로그램 개발

교육 프로그램은 <표 III-13>과 같이 파이썬 기초 프로그래밍 언어 교육과 언플러그드를 활용한 정렬 알고리즘의 개념을 기반으로 수행시간 중심의 알고리즘 교육을 하는 데 초점을 두어 개발하였다. 또한 초등학교 교사들을 대상으로 한 요구분석 결과를 토대로 학습 시 어려울 것이라고 생각되는 학생 이해도를 높이기 위해 학생들이 주변에서 경험할 수 있는 학습 소재로 선정하였으며, 당일 학습한 내용을 직접 수행해보는 개별 과제를 매일 부여하고 수행하며 이해도를 높일 수 있도록 하였다. 교육 프로그램은 42차시 분량의 학생용 교재와 수업을 위한 교수 학습 과정안으로 개발하였다.

가. 교육 내용

<표 III-13> 수행시간 중심 초등 알고리즘 교육프로그램 내용

차시	일자	주제	활동내용	비고
1	8.13. (월)	OT 및 파이썬 기초	<ul style="list-style-type: none"> • 오리엔테이션 및 사전검사지 투입 • 파이썬 기초기능 익히기 <ul style="list-style-type: none"> - 변수, 연산자 • 선택구조와 논리연산을 이용한 프로그램 <ul style="list-style-type: none"> - 조건문(if, elif, else) 	<개별과제1> 선택구조와 논리연산을 이용한 프로그램 제작
2	8.14. (화)	리스트와 반복구조	<ul style="list-style-type: none"> • 정보의 입력 <ul style="list-style-type: none"> - input의 사용 • 반복구조 <ul style="list-style-type: none"> - for, range를 사용한 반복구조 - while을 사용한 반복 구조 • 리스트의 사용 <ul style="list-style-type: none"> - 리스트를 이용한 간단한 게임 만들기 	<개별과제2> 반복구조와 리스트를 이용한 프로그램 만들기

3	8.15. (수)	정렬	<ul style="list-style-type: none"> • 언플러그드 활동을 통한 다양한 정렬의 이해 <ul style="list-style-type: none"> - 서로 다른 개수의 동전이 든 필름통의 무게 비교 • 정렬 알고리즘 <ul style="list-style-type: none"> - 버블정렬, 선택정렬, 퀵정렬 • 세 정렬 알고리즘의 수행시간 비교하기 	<개별과제3> 정렬프로그램 적용하여 주변 문제 해결하기
4	8.16. (목)	알고리즘 수행시간	<ul style="list-style-type: none"> • 언플러그드 활동을 통한 수행시간 개념의 이해 <ul style="list-style-type: none"> - 최소 비용 신장 트리, 최단경로 알고리즘 • 간단한 프로그램 작성하기 <ul style="list-style-type: none"> - 피보나치 수열, 가위바위보 게임 	<개별과제4> 자유주제 프로그램 주제 선정
5	8.17. (금)	프로젝트 작품 제작하기	<ul style="list-style-type: none"> • 개인별 프로젝트 작품 계획서 작성하기 • 개인별 프로젝트 작품 제작하기 	<개별과제5> 자유주제 프로그램 제작
6	8.18. (토)	프로젝트 작품 발표하기	<ul style="list-style-type: none"> • 개인별 프로젝트 작품 발표하기 • 사후검사지 투입 	

나. 학생용 교재 개발

학생용 교재는 <표 III-12>, <표 III-13>을 기반으로 제작하였으며 요구 분석의 내용을 기반으로 초등학생들에게 너무 어렵지 않도록 프로그램 구조는 제시하되, 알고리즘의 핵심이 되는 부분은 비워두어 빈 칸에 어떤 내용이 들어가야 하는지 생각해보고 프로그램을 작성해보며 알고리즘을 학습하도록 구성하였다.

수행시간 중심 초등 알고리즘 교육 프로그램에서 핵심이 되는 정렬 알고리즘의 수행시간을 비교하는 부분 중 일부인 15~28/42차시의 '정렬 알고리즘을 통한 수행시간 분석' 수업에 사용된 학생용 교재는 [그림 III-1]~[그림 III-5]와 같다. 수업은 언플러그드로 정렬의 필요성과 정렬 방법을 이해하고, 파이썬으로 프로그래밍해본 뒤, 정렬 방법 간의 수행시간을 비교·분석해보는 순으로 진행하였다.

제3장 정렬을 통한 알고리즘 수행시간 분석

☆ 가장 큰 수와 가장 작은 수 찾기

<보기1> ○○ 초등학교 5학년 1반 학생들의 키

140	161	137	125	163	151	127	139	144	154
152	160	128	141	136	126	166	165	140	129
135	143	164	150	155	138	142	162	153	145

- 1) <보기1>에서 '가장 키가 큰 학생의 키와 '가장 키가 작은 학생의 키'를 찾아보세요.
- 2) <보기1>에서 10번째로 키가 큰 학생의 키를 적어보세요.

<보기2> ○○ 초등학교 5학년 2반 학생들의 키

125	126	127	128	129	131	132	133	134	135
136	137	138	139	140	145	146	147	148	149
151	152	154	155	156	160	161	162	164	165

- 3) <보기2>에서 '가장 키가 큰 학생의 키와 '가장 키가 작은 학생의 키'를 찾아보세요.
- 4) <보기2>에서 10번째로 키가 큰 학생의 키를 적어보세요.
- 5) <보기1>과 <보기2> 중 어떤 것이 문제를 해결하기 쉬웠나요? 그 이유는 무엇일까요?

<정렬이란?> 주어진 자료를 어떤 기준에 의하여 순서대로 나열하는 것
 데이터는 무작위로 나열되었을 때보다 어떤 규칙에 따라 순서대로 정렬되어 있을 때 훨씬 유용하게 사용될 수 있습니다. 컴퓨터에서는 수많은 파일을 이름, 크기, 날짜순으로 정렬 가능하며 여러 가지 데이터도 규칙에 따라 정렬하는 것이 가능합니다.
 정렬 방법도 여러 종류가 있는데 어떤 정렬 방법을 사용하느냐에 따라 컴퓨터의 처리 속도(알고리즘의 수행시간)가 달라집니다.

☆ 정렬 방법 체험하기(1) - 선택 정렬



- 1) 평형 저울을 이용하여 필름통 2개를 비교해가며 가장 가벼운 필름통을 찾습니다.
 평형 저울을 한 번 쓸 때마다 카운터를 눌러 평형 저울 사용 횟수를 셉니다.
※ 한 번에 용기 2개의 무게만 비교합니다. 반드시 평형 저울을 사용해 비교합니다.
- 2) 가장 가벼운 필름통을 찾으면 아래 활동지의 ㉠에 놓고, 남은 필름통 4개 중에 가장 가벼운 필름통을 찾아 ㉡에 놓습니다.
- 3) 위의 과정을 반복하여 가벼운 것부터 무거운 것까지 순서대로 활동지에 놓습니다.
- 4) 정렬이 잘 되었는지 필름통을 열어 확인합니다.
- 5) 정렬이 잘 되었으면 카운터 수의 합계를 구해봅니다.

<활동지>

	가벼운 <----->	무거운
용기	㉠	㉡
비교 횟수		

용 비교 횟수:

*네이버 TV, 소프트웨어야 놀자 <https://terms.naver.com/entry.nhn?docId=4369097&cid=59801&categoryId=59801>

[그림 III-1] 「언플러그드 활동을 통한 정렬 개념 이해」 학생용 교재

[그림 III-1]은 정렬 알고리즘의 수행시간 비교 분석을 도입하기 이전 언플러그드 활동을 통해 정렬의 필요성과 정렬방법 중 하나인 선택 정렬의 개념을 알아보는 학생용 교재의 일부이다. 학생들이 주변에서 쉽게 접할 수 있는 소재인 '키'가 큰 학생과 작은 학생을 찾아보는 과정 속에서 어떤 정렬 방법을 사용하느냐에 따라 처리 속도(알고리즘 수행시간)이 달라진다는 사실을 파악하며 정렬 알고리즘의 필요성과 수행시간 비교의 필요성을 깨닫고, 선택 정렬의 방법으로 서로 다른 개수의 동전이 들어 있는 필름통을 정렬해보며 원리를 파악하도록 구성하였다.

☆ 정렬 프로그래밍(A) - 선택 정렬

선택 정렬(selection sort)은 컴퓨터가 사용하는 정렬 방법의 하나로, 정렬대상이 위치한 곳과 정렬 결과를 놓을 곳으로 나누고, 가장 가벼운 것을 골라 정렬결과 정소를 옮겨놓습니다. 그 다음으로 가벼운 것을 골라 옮겨 놓는 것을 반복하여 더는 정렬 대상에 아무것도 없을 때까지 반복합니다.

a	최소 위치	b	컴퓨터의 수행 절차
0	0	1	0번의 값을 임시 저장하고 1~4번과 비교해서 가장 작은 값을 0번에 저장하고, 임시 저장된 수를 바꾼 자리의 수에 다시 넣는다.
		2	
		3	
		4	
1	1	2	1번의 값을 임시 저장하고 2~4번과 비교해서 가장 작은 값을 1번에 저장하고, 임시 저장된 수를 바꾼 자리의 수에 다시 넣는다.
		3	
		4	
		4	

a	최소 위치	b	컴퓨터의 수행 절차
2	2	3	첫 번째, 두 번째 데이터를 제외한 나머지 데이터에서 가장 작은 데이터인 8을 세 번째 데이터인 15와 교환한다.
		4	
3	3	4	첫 번째, 두 번째, 세 번째 데이터를 제외한 나머지 데이터에서 가장 작은 데이터인 11을 네 번째 데이터인 15와 교환한다.
		4	

```

import random
def selection_sort(A):
    for i in range(____):
        for j in range(i+1, len(a)):
            if ____ > ____:
                a[i], a[j] = _____
    return a

a = [random.randint(0,100) for k in range(5)]
print(a)
selection_sort(a)
print(a)
                
```

[실행]

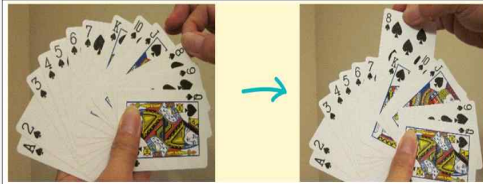
[84, 6, 56, 1, 31]

[1, 6, 31, 56, 84]

[그림 III-2] 「선택 정렬 프로그래밍」 학생용 교재

[그림 III-2]은 언플러그드로 알아본 선택 정렬을 파이썬 프로그래밍으로 알아보는 학생용 교재이다. 서로 다른 개수의 동전이 들어 있는 필름통으로 정렬하는 과정을 그림으로 정리하여 그림의 빈 칸에 들어갈 숫자를 채워보며 선택 정렬이 어떤 알고리즘으로 구성되어 있는지 파악하고, 프로그래밍을 위해 세 개의 변수(a, b, 최소 위치)가 필요함을 파악하도록 하였다. 이후 파이썬 선택 정렬 프로그램에서 핵심이 되는 부분을 빈 칸으로 두어 스스로 채워보며 선택 정렬 알고리즘을 파악하도록 교재를 설계하였다.

☆ 정렬 방법 체험하기(2) - 삽입 정렬



<무작위로 섞여 있는 카드를 왼손에 잡고 순서대로 정리하는 게임>

- 1) 트럼프 카드에서 같은 모양 13장의 카드를 꺼낸 다음 무작위로 섞어서 뒤집어 놓습니다.
- 2) 카드를 A부터 K까지 정리하는 전략을 생각해보고 아래 활동지에 기록합니다.
(카드 순서: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K)
- 3) 선생님의 신호에 따라 왼손에 카드를 잡고 오른손으로만 카드를 정리합니다.
- 4) 정리가 완료되면 카드를 내려놓고, 스톱워치를 정지시켜 기록을 확인합니다.
- 5) 정리를 효과적으로 할 수 있는 전략을 찾아봅시다.

<활동지>

전략	기록

a	key	b	컴퓨터의 수행 절차
3	3번 (3)	2	<p>네 번째 데이터인 3과 앞에 위치한 1, 11, 15의 크기를 비교한다. 3은 1보다 크고 11, 15보다 작으므로 11과 15를 한 칸씩 뒤로 보내고 3을 11 앞에 위치시킨다.</p>
4	4번 (8)	3	<p>마지막 데이터인 8과 앞에 위치한 1, 3, 11, 15의 크기를 비교한다. 1, 3보다 크고 11, 15보다 작으므로 11과 15를 한 칸씩 뒤로 보내고 8을 11 앞에 위치시킨다.</p>

```

import random
ds = [random.randint(1, 100) for x in range(5)]
print(ds)

for a in range(1, _____):
    key = ds[a]
    b = a-1
    while b>=0 and ds[b]> _____
        ds[b+1] = ds[b]
        b = b-1
    ds[b+1] = _____
print(ds)
print(a)
                
```

[실행]

[25, 22, 88, 64, 3]

[3, 22, 25, 64, 88]

[그림 III-3] 「삽입 정렬 프로그래밍」 학생용 교재 일부

[그림 III-3]은 삽입 정렬을 언플러그드를 통해 체험하고 파이썬 프로그래밍으로 알고리즘을 알아보는 학생용 교재의 일부이다. 학생들은 무작위로 섞여 있는 카드를 순서대로 정리해보며 전략을 생각해보고, 자연스럽게 삽입의 개념을 파악하여 삽입 정렬의 방법을 깨닫도록 하였다. 이후 파이썬 프로그래밍으로 삽입 정렬의 핵심이 되는 알고리즘 부분을 빈 칸에서 채워보며 삽입 정렬 알고리즘 구현을 위해 3가지의 변수(a, key, b)가 필요함을 깨닫고, 삽입 정렬 알고리즘을 파악하도록 교재를 구성하였다.

☆ 정렬 방법 체험하기(3) - 버블 정렬

<무작위로 섞여 있는 카드를 버블을 이용하여 오름차순으로 정리하는 게임>

2GB, 4GB, 8GB, 16GB, 32GB, 64GB, 128GB, 256GB

※양나라 외, 컴퓨터 없이 배우는 아린이 코딩, 계림북스(2016)

- 메모리 카드를 뒤집은 다음 잘 섞어 일렬로 놓습니다.
- 왼쪽에서 처음과 두 번째 간에 놓인 카드 2장을 뒤집어 숫자를 비교합니다.

비교
- 숫자가 작은 카드를 왼쪽에, 숫자가 큰 카드를 오른쪽에 둡니다.

정렬
- 숫자가 안 보이도록 카드를 다시 뒤집습니다.
- 두 번째와 세 번째 카드를 뒤집어 숫자를 비교해 작은 수를 왼쪽에 둡니다.

비교
- 위와 같은 방식으로 일곱 번째와 여덟 번째 카드까지 비교합니다.
- 다시 첫 번째와 두 번째 카드를 비교하여 자리바꿈이 없을 때까지 반복합니다.

a	b	b+1	컴퓨터의 수행 절차
1	2	3	마찬가지로 방식을 적용해 세 번째 데이터인 11과 네 번째 데이터인 8의 위치를 바꾼다. 두 번째로 큰 데이터인 11이 뒤에서 두 번째에 위치하게 된다. <div style="text-align: center;"> </div>
2	0	1	처음부터 다시 시작한다. 첫 번째 데이터인 1과 두 번째 데이터인 3의 크기를 비교하는데 앞에 위치한 1이 작으므로 그대로 둔다. <div style="text-align: center;"> </div>
2	1	2	두 번째 데이터인 3과 세 번째 데이터인 8의 크기를 비교하는데, 앞에 위치한 3이 작으므로 그대로 둔다. 세 번째로 큰 데이터인 8이 뒤에서 세 번째에 위치하게 된다. <div style="text-align: center;"> </div>
3	0	1	처음부터 다시 시작한다. 첫 번째 데이터인 1과 두 번째 데이터인 3의 크기를 비교하는데, 앞에 위치한 1이 작으므로 그대로 둔다. 데이터들에 대한 정렬이 완료된다. <div style="text-align: center;"> </div>

```

import random
ds = [random.randint(1, 100) for x in range(5)]
print(ds)
for a in range(0, _____):
    for b in range(0, len(ds)-1-a):
        if ds[b] > ds[b+1]:
            ds[b], ds[b+1] = _____
print(ds)
    
```

[실행]

[25, 22, 88, 64, 3]

[3, 22, 25, 64, 88]

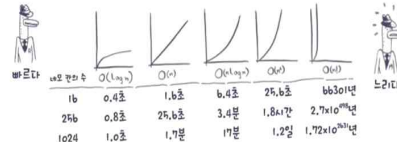
[그림 III-4] 「버블 정렬 프로그래밍」 학생용 교재 일부

[그림 III-4]는 버블 정렬을 언플러그드를 통해 체험하고 파이썬 프로그래밍으로 알고리즘을 알아보는 학생용 교재의 일부이다. 학생들은 무작위로 섞여 있는 카드를 양 옆 카드와 비교하며 정렬하는 버블 정렬의 방법으로 정렬해보고, 자연스럽게 버블의 개념을 파악하여 버블 정렬의 방법을 깨닫도록 하였다. 이후 파이썬 프로그래밍으로 버블 정렬의 핵심이 되는 알고리즘 부분을 빈 칸에서 채워보며 버블 정렬 알고리즘 구현을 위해 3가지의 변수(a, b, b+1)가 필요함을 깨닫고, 버블 정렬 알고리즘을 파악하도록 교재를 구성하였다.

[정렬 알고리즘 프로그래밍 정리]

정렬	파이썬 프로그래밍	설명
선택정렬	<pre>import random def selection_sort(A): for i in range(____): for j in range(i+1, len(A)): if ____ > ____: a[j], a[i] = _____ return a a = [random.randint(0,100) for k in range(5)] print(a) selection_sort(a) print(a)</pre>	정렬되지 않은 데이터들에 대해 가장 작은 데이터를 찾아 가장 앞의 데이터와 교환해나가는 방식
삽입정렬	<pre>import random ds = [random.randint(1, 100) for x in range(5)] print(ds) for a in range(1, ____): key = ds[a] b = a-1 while b>=0 and ds[b]> ____: ds[b+1] = ds[b] b = b-1 ds[b+1] = ____ print(ds) print(a)</pre>	아직 정렬되지 않은 임의의 데이터를 이미 정렬된 부분의 적절한 위치에 삽입해 가며 정렬하는 방식
버블정렬	<pre>import random ds = [random.randint(1, 100) for x in range(5)] print(ds) for a in range(0, ____): for b in range(0, len(ds)-1-a): if ds[b] > ds[b+1]: ds[b], ds[b+1] = _____ print(ds)</pre>	서로 이웃한 데이터들을 비교하며 가장 큰 데이터를 가장 뒤로 보내며 정렬하는 방식

☆ 그래프를 이용해서 알고리즘의 수행시간 알아보기



*아디트야 바르카바(2017), Hello Coding 그림으로 개념을 이해하는 알고리즘, 한빛미디어, p. 19.
어떤 알고리즘이 효율적인지 알아보기 위해서는 그래프를 그려서 기울기가 더 작은 것인지 알아보면 됩니다. 기울기가 클수록(왼쪽→오른쪽) 시간이 오래 걸리기 때문에 비효율적입니다.

[정렬 알고리즘의 수행시간 비교]

- 가로축(데이터 개수-100, 200, 400, 800, 1600) / 세로축(30번 수행-실행시간(ms) 평균)
- sortedData: 오름차순 데이터 / randomData: 임의 순서의 데이터 / reversedData: 내림차순 데이터

정렬	그래프	설명
선택정렬		선택정렬은 데이터의 정렬 상태에 관계없이 늘 같은 시간이 소요된다. 오름차순 100(ms) 임의순서 100(ms) 내림차순 100(ms)
삽입정렬		정렬된 데이터는 삽입정렬이 매우 빠르다. 오름차순 5(ms) 임의순서 125(ms) 내림차순 250(ms)
버블정렬		데이터 크기가 커질수록 실행시간이 가파르게 증가한다. 오름차순 100(ms) 임의순서 250(ms) 내림차순 400(ms)

[그림 III-5] 「정렬 알고리즘 수행시간 분석」 학생용 교재

[그림 III-5]는 앞서 정리한 3가지 정렬 방법인 선택, 삽입, 버블정렬의 수행시간을 비교해보는 학생용 교재이다. 세 정렬 방법의 알고리즘을 파이썬 프로그램의 빈 칸을 채우며 다시 한 번 정리해보고, 세 정렬을 그래프로 나타낸 뒤 그래프의 기울기를 관찰하며 정렬 알고리즘의 수행시간을 비교하도록 하였다. 주어지는 데이터 크기에 따라 수행시간에 어떤 차이가 나타나는지 관찰하고, 어떤 정렬 방법이 효율적인가 따져보며 수행시간의 개념과 수행시간 분석의 필요성을 깨닫도록 교재를 구성하였다.

다. 교사용 과정안 개발

교사용 과정은 각 주제별로 일반적인 수업에서 가장 많이 사용되는 '도입-전개-정리'의 구조를 사용하여 구성하였으며, 파이썬 정렬 알고리즘을 중심으로 한 수행시간 초등 알고리즘에 대한 교사용 과정안은 <표 III-14>, <표 III-15>와 같다.

<표 III-14> 정렬 알고리즘을 통한 수행시간 분석 교사용 과정안(1)

일시	2018. 8. 15.(수)		대상	4~6학년 25명
주제	선택정렬, 삽입정렬		차시	15~21/42
활동명	선택정렬, 삽입정렬 알아보기	소요시간	300분	
학습목표	1. 버블정렬, 선택정렬, 삽입정렬의 원리를 이해할 수 있다. 2. 버블정렬, 선택정렬, 삽입정렬을 활용한 프로그램을 만들 수 있다.			
학습단계	학습요소	교수·학습 활동		
도입	동기 유발	▷ 정렬이 필요한 상황을 제시한다.		
	학습목표 제시	▷ 정렬의 필요성을 알고 정렬을 활용한 프로그램을 만들 수 있다.		
전개	선택정렬 익히기	▷ 평형 저울을 이용해 필름통 무게 비교하기(언플러그드) - 용기 2개씩 무게를 비교하며 선택정렬의 원리 익히기 ▷ 선택정렬 진행과정을 도식화하여 프로그램 완성하기		
	삽입정렬 익히기	▷ 무작위 배열된 트럼프 카드를 정리하기(언플러그드) - 왼손으로 카드를 잡고 오른손으로 순서대로 정리하며 삽입정렬의 원리 익히기 ▷ 삽입정렬 진행과정을 도식화하여 프로그램 완성하기		
	선택정렬, 삽입정렬 비교하기	▷ 선택정렬, 삽입정렬 비교하기 - 선택정렬과 삽입정렬에 알맞은 상황 찾아보기 - 상황별로 선택정렬, 삽입정렬을 이용해 프로그램을 만들어보고 더 효율적인 프로그램 찾아보기		
정리	학습내용 정리 및 과제 부여	▷ 선택정렬, 삽입정렬 학습 내용 복습 ▷ 선택정렬, 삽입정렬을 활용한 자유 주제 프로그램 제작 과제 부여		

〈표 III-15〉 정렬 알고리즘을 통한 수행시간 분석 교사용 과정안(2)

일시	2018. 8. 16.(목)		대상	4~6학년 25명
주제	버블정렬, 알고리즘 수행시간		차시	22~28/42
활동명	버블정렬, 알고리즘 수행시간 알아보기		소요시간	300분
학습목표	1. 버블정렬을 활용한 프로그램을 만들 수 있다. 2. 버블정렬, 선택정렬, 삽입정렬을 통해 수행시간을 비교하여 설명할 수 있다.			
학습단계	학습요소	교수·학습 활동		
도입	동기 유발	▷ 선택정렬, 삽입정렬과 다른 버블정렬이 사용되는 상황을 제시한다.		
	학습목표 제시	▷ 버블정렬을 활용한 프로그램을 만들어보고, 선택/삽입정렬과 비교해보며 수행시간을 비교하여 설명할 수 있다.		
전개	버블정렬 익히기	▷ 무작위 배열된 메모리 카드를 정리하기(언플러그드) - 왼쪽에서부터 메모리 카드를 2장씩 서로 비교하며 버블정렬의 원리 익히기 ▷ 버블정렬 진행과정을 도식화하여 프로그램 완성하기		
	수행시간 비교하기	▷ 그래프를 통한 정렬 알고리즘의 수행시간 비교해보기 - 그래프 기울기에 대한 이해 - 그래프의 기울기를 관찰해보고 자료의 크기에 따른 효율적인 정렬 알고리즘에 대한 의견 나누기		
정리	학습내용 정리 및 과제 부여	▷ 버블정렬, 수행시간 학습 내용 복습 ▷ 버블정렬을 활용한 자유 주제 프로그램 제작 과제 부여		

IV. 연구의 적용 및 결과 분석

1. 연구 대상

본 연구의 대상은 교육기부 프로그램의 지원자 표집에 의한 지원자 표본 (volunteer sample)으로 초등학교 4~6학년 중 25명을 선정하여 진행하였다. 연구 대상의 학년과 성별은 <표 IV- 1>과 같다.

<표 IV-1> 교육기부 프로그램 지원자 학년과 성별

	남	여	합계
4학년	3	2	5
5학년	4	3	7
6학년	8	5	13
합계	15	10	25

2. 검사도구

본 연구에서는 파이썬 정렬 알고리즘을 중심으로 수행시간 중심의 초등 알고리즘 교육 프로그램을 개발하고 적용하여 학습자의 논리적 사고력, 컴퓨팅 사고력의 변화를 분석하고 그 효과를 검증하는 것을 목표로 하였다.

논리적 사고력의 검사는 한국교육개발원에서 개발한 표준화된 과학적 사고 기능 검사의 하위 검사 중 하나인 GALT(Group Assessment of Logical Thinking)로 실시하였으며, GALT는 초등학생들을 대상으로 검사의 적절성이 검토·확인되어 표준화된 논리적 사고 검사 도구로 여겨지기 때문에 선정하였다 (한종하, 1991).

컴퓨팅 사고력 검사는 한국 비버챌린지를 주관하는 Bebras Korea에서 개발

한 비버챌린지(Bebras Challenge) 2017로 실시하였으며, 초등학교 5~6학년용 문제를 제공하여 연령에 따라 적절한 컴퓨팅 사고력을 측정할 수 있는 도구라고 알려져 있어 선정하였다.

교육 프로그램의 효과를 분석하기 위해 교육 전후에 사전·사후 검사를 실시하였으며, 이 실험 설계 과정을 도식화하면 <표 IV-2>, <표 IV-3>과 같다.

<표 IV-2> 논리적 사고력 검사 실험 설계

	사전 검사	적용	사후 검사
실험 집단 (25명)	O ₁	X	O ₂

X: 파이썬 정렬 알고리즘 중심으로 수행시간 중심 초등 알고리즘 교육 프로그램 시행
O₁: 사전검사(GALT test Figure A style)
- Paired t-test
O₂: 사후검사(GALT test Figure B style)
- Paired t-test

<표 IV-3> 컴퓨팅 사고력 검사 실험 설계

	사전 검사	적용	사후 검사
실험 집단 (25명)	O ₁	X	O ₂

X: 파이썬 정렬 알고리즘 중심으로 수행시간 중심 초등 알고리즘 교육 프로그램 시행
O₁: 사전검사(Bebras Challenge 2017 A style)
- Paired t-test
O₂: 사후검사(Bebras Challenge 2017 B style)
-Paired t-test

사전검사와 사후검사 결과 교육적 효과 검증을 위한 통계는 IBM의 SPSS(Statistical Package for the Social Sciences) 14.0을 활용하여 처리하였다.

3. 연구 결과

가. 정규성 검정

표본의 크기가 25명으로 '10 ≤ n < 30'에 위치하기 때문에, 실험 집단의 논리적 사고력 검사와 컴퓨팅 사고력 검사 결과가 정규성을 확보하고 있는지 정규성 검정을 각각 시행하였다. 이 때 교육 전후의 결과를 비교하기 때문에 자료의 독립성이 만족되지 않으므로, 정규성이 확보되는 경우에는 대응표본 t검정(Paired t test)을, 정규성을 만족하지 못하는 항목의 경우에는 비모수 검정 중 대응표본 t검정에 대응하는 부호 순위 검정(Wilcoxon signed rank test)을 실시하였다. 비모수 통계검정은 모수 검정과 다르게 정규분포를 가정하지 않고 주어진 데이터에서 직접 확률을 계산하여 통계학적 검정을 하는 방법으로, 부호 순위 검정은 그 중에서도 자료의 독립성이 만족되지 않는 경우에 적합한 방법(이훈영, 2010)이기 때문에 선정하였다.

1) 논리적 사고력 검사에 대한 정규성 검정

논리적 사고력 검사에 대한 샤피로-윌크(Shapiro-Wilks) 정규성 검사 결과는 <표 IV-4> 와 같다.

<표 IV-4> 논리적 사고력 검사에 대한 정규성 검정 결과

하위 요소	평균	표준편차	최댓값	최솟값	통계량	유의확률
보존 논리	1.240	1.200	3	-1	.920	.051
비율 논리	.4400	.9609	2	-1	.882	.008*
변인통제 논리	.4400	1.083	3	-1	.868	.004*
확률 논리	.1600	1.143	2	-2	.924	.063
상관 논리	.0800	.4000	1	-1	.547	.000*
조합 논리	.2800	.6137	1	-1	.764	.000*

*p < .05

논리적 사고력 검사에 대한 정규성 검정 결과 ‘비율논리’, ‘변인통제논리’, ‘상관논리’, ‘조합논리’ 지수는 유의확률(p)이 0.05 미만으로 귀무가설을 기각하여 정규성을 만족하지 않았다. ‘보존논리’, ‘확률논리’ 지수는 유의확률(p)이 각각 ‘0.051’, ‘0.058’로 .05보다 크게 나타나 귀무가설을 채택하여 정규성을 만족하다고 가정하였다.

2) 컴퓨팅 사고력 검사에 대한 정규성 검정

컴퓨팅 사고력 검사에 대한 샤피로-윌크(Shapiro-Wilks) 정규성 검사 결과는 <표 IV-5> 와 같다.

<표 IV-5> 컴퓨팅 사고력 검사에 대한 정규성 검정 결과

	평균	표준편차	최대값	최소값	통계량	유의확률
컴퓨팅 사고력	.2800	.5416	2	0	.569	.000*

* $p < .05$

컴퓨팅 사고력 검사에 대한 정규성 검정 결과 유의확률(p)이 0.05 미만으로 귀무가설을 기각하여 정규성을 만족하지 않는 것으로 나타났다.

나. 논리적 사고력 집단 내 비교

논리적 사고 검사(GALT) 결과 중 샤피로-윌크(Shapiro-Wilks) 검사에서 정규성을 확보한 요소는 대응표본 t검정(Paired T-test)을, 정규성을 확보하지 못한 요소는 비모수검사인 윌콕슨의 부호-순위 검정(Wilcoxon's signed ranks test)을 실시하였다.

〈표 IV-6〉 논리적 사고력 사전·사후 검사 결과(대응표본 t검증)

하위 항목	학생 수	사전 검사		사후 검사		t	유의확률
		평균	표준편차	평균	표준편차		
보존 논리	25	2.04	.789	3.28	.890	-5.167	.000**
확률 논리	25	.920	.909	1.080	.862	-.700	.491

****p<.01**

대응표본 t검정 결과인 〈표 IV-6〉에서 ‘보존논리’의 평균점수는 사전 2.04에서 사후 3.28로 1.24점 상승하였고, 유의확률은 0.0001로 평균점수의 유의미한 상승이 있는 것으로 나타났다. 반면 ‘확률논리’는 평균 0.920에서 1.080으로 0.16 상승하긴 하였으나, 유의확률은 0.491로 평균의 변화는 통계적으로 유의미하지 않은 것으로 나타났다.

〈표 IV-7〉 논리적 사고력 사전·사후 검사 결과(윌콕슨 부호-순위 검정)

하위 항목	학생 수	사전 검사		사후 검사		Z	유의확률
		평균	표준편차	평균	표준편차		
비율 논리	25	2.280	2.111	2.720	1.814	-2.147 ^b	.032*
변인통제 논리	25	.8800	.9273	1.320	1.180	-1.979 ^b	.048*
상관 논리	25	.1600	.3741	.2400	.4358	-1.000 ^b	.317
조합 논리	25	1.160	.6879	1.440	.7118	-2.111 ^b	.035*

*p<.05, b. 음의 순위를 기준으로

〈표 IV-7〉의 부호-순위 검정 결과를 살펴보면 ‘비율논리’의 평균 점수는 사전 2.280에서 사후 2.720으로 0.44 증가하였고 유의확률은 0.032로 나타났다. ‘변인통제 논리’의 평균 점수는 사전 .8800에서 사후 1.320으로 0.44 증가하였

고 유의확률은 0.048로 나타났다. ‘조합논리’의 평균 점수는 사전 1.160에서 사후 1.440으로 0.280 증가하였으며, 유의확률은 0.035로 나타나 ‘비율논리’, ‘변인통제 논리’, ‘조합논리’에서 유의미한 평균 점수의 상승이 있는 것으로 나타났다. 반면 ‘상관논리’의 평균점수는 사전 0.1600에서 사후 0.2400으로 0.08 증가하였으나, 유의확률은 0.317로 평균의 변화는 통계적으로 유의미하지 않았다.

다. 컴퓨팅 사고력 집단 내 비교

컴퓨팅 사고력 검사는 정규성을 만족하지 못한 관계로 윌콕슨의 부호-순위 검정을 실시하여 비교하였으며, 비교·분석한 결과는 <표 IV-8>과 같다.

<표 IV-8> 컴퓨팅 사고력 사전·사후 검사 결과(윌콕슨 부호-순위 검정)

	학생 수	사전 검사		사후 검사		Z	유의확률
		평균	표준편차	평균	표준편차		
컴퓨팅 사고력	25	7.600	1.979	7.880	1.900	-2.333 ^b	.020*

*p<.05, b. 음의 순위를 기준으로

<표 IV-8>의 부호-순위 검정 결과를 살펴보면 컴퓨팅 사고력의 평균점수는 사전 7.600에서 사후 7.880으로 0.280 증가하였으며, 유의확률은 0.020으로 통계적으로 유의미하였다.

4. 연구 결과 분석

파이썬 정렬 알고리즘을 중심으로 한 수행시간 중심 초등 알고리즘 교육 프로그램을 적용하기 전에 실시한 논리적 사고력 사전 검사와 컴퓨팅 사고력 사전 검사 결과를 분석하기에 앞서 실험집단이 정규분포를 갖추었는지 확인하기 위해 샤피로-윌크(Shapiro-Wilks) 검사를 실시하였다. 정규성 검정 결과를 토대로 정

규성을 확보한 요소는 대응표본 t검정(Paired T-test)을, 정규성을 확보하지 못한 요소는 비모수검사인 윌콕슨의 부호-순위 검정(Wilcoxon's signed ranks test)을 각각 실시하였다.

논리적 사고력은 <표 IV-6>, <표 IV-7>에서 볼 때 '보존논리', '비율논리', '변인통제논리', '조합논리'의 4가지 항목에서는 통계적으로도 유의미한 향상과 더불어 사전 검사 대비 사후 검사 결과의 평균이 향상되었으며, '확률논리'와 '상관논리'의 2가지 항목에서는 통계적으로는 유의하지 않았으나 역시 사전 검사 대비 사후 검사 결과 평균이 향상되었다. 컴퓨팅 사고력은 <표 IV-8>에서 통계적으로 유의미한 향상을 보였으며 사전 검사 대비 사후 검사 평균 역시 향상되었다. 이러한 변화는 학생들이 텍스트 프로그래밍 언어인 파이썬의 기초 문법을 학습하고, 알고리즘의 빈 칸을 채워 넣으며 수행시간을 분석하여 효율성을 비교하는 본 교육 활동이 학생들의 사고력 향상에 긍정적인 영향을 가져왔기 때문으로 분석할 수 있다.

따라서 본 연구를 통해 파이썬 정렬 알고리즘을 중심으로 한 수행시간 중심의 초등 알고리즘 교육은 초등학생의 논리적 사고력과 컴퓨팅 사고력 향상에 긍정적인 영향을 미친다는 것을 입증하였으며, 텍스트 프로그래밍 언어인 파이썬과 어렵게만 느껴지는 수행시간, 알고리즘의 학습 요소도 초등학생들의 수준에 맞게 구성한다면 소프트웨어 교육 도구로 충분히 활용될 수 있음을 제시하였다.

V. 결론 및 제언

본 연구에서는 파이썬 정렬 알고리즘을 중심으로 수행시간 중심 초등 알고리즘 교육 프로그램이 논리적 사고력과 컴퓨팅 사고력에 미치는 영향을 파악하고자 하였다. 이를 위해 ADDIE 모형의 절차에 따라 사전 요구 분석을 토대로 교육 프로그램을 개발하고 6일간의 집중 이수 교육을 적용한 뒤 사전·사후 검사를 통해 논리적 사고력과 컴퓨팅 사고력의 변화를 검증하였다.

검사 결과 본 연구에서 개발한 교육 프로그램은 초등학생의 논리적 사고력과 컴퓨팅 사고력 향상에 통계적으로 유의미한 효과를 가진 것으로 나타났으며, 학생들은 텍스트 프로그래밍 언어인 파이썬을 다루며 언플러그드나 교육용 프로그래밍 언어로 학습할 때보다 더 많은 성취감을 느껴 교육의 만족도도 높았다. 이에 초등학교 고학년 학생들을 대상으로 한 프로그래밍 교육에서 파이썬이 활용될 가치는 충분하며, 초등학생 수준에 적합한 주제를 선정하여 교육할 경우 알고리즘과 수행시간의 개념도 충분히 유의미함을 알 수 있다.

다만, 본 연구의 실험집단은 25명으로서 일반적인 상관연구에 통상적으로 필요하다고 여겨지는 30명 이상의 참여자를 확보하지 못하여 일반화하는 데 한계가 있다. 또한 본 연구에서 개발한 프로그램은 별도의 비교집단 없이 실험집단의 사전·사후 검사 결과만으로 검증이 이루어졌기 때문에, 본 프로그램으로 인한 교육만으로 사후 검사 결과가 상승했는지 상관관계를 분명하게 규정할 수 없다는 문제가 있다.

추후의 연구에서는 상관연구에 필요한 30명 이상의 참여자를 확보하고, 실험집단과 비교집단을 구성하여 변인 체계적으로 통제된 뒤에 연구 결과에 대한 각 요인들을 좀 더 체계적으로 분석할 필요가 있다.

참 고 문 헌

- 강자영. (2018). **초등학교 소프트웨어 교육이 논리적 사고력에 미치는 영향 분석**. 목포대학교 교육대학원.
- 교육부. (2015). **교육부 소프트웨어 교육 정책**.
- 교육부. (2015). **소프트웨어 지향적 사회를 위한 인재 육성 방안**.
- 김대식. (2017). 4차 산업혁명 시대 초등교육의 정체성과 역할. **한국교육사상 연구회학술지**, 31(4), 23~45.
- 김대욱. (2019). 컴퓨팅 사고력에 기초한 유아를 위한 언플러그드 코딩의 개념과 전략. **국제문화기술진흥원**, 5(1), 297~303.
- 김민수. (2004). **여러 가지 정렬 알고리즘에 관한 연구**. 인제대학교 교육대학원.
- 김우찬. (2016). **소프트웨어교육이 초등학교 5학년 학생들의 논리적 사고력과 창의적 문제해결력에 미치는 효과**. 경희대학교 대학원.
- 김은지. (2017). **파이썬으로 정렬 알고리즘 구현하기**. (<http://ejklike.github.io/2017/03/04/sorting-algorithms-with-python.html>)
- 김종훈, 김종진. (2013). **컴퓨터개론**. 서울: 한빛아카데미.
- 류미영, 한선관. (2015). 초등 SW교육을 위한 CT교육 프로그램 개발. **한국정보교육학회 학술논문집**, 19(1), 11~20.
- 맹미선. (2017). **알파고 쇼크와 '4차 산업혁명' 담론의 확산: 과학기술 유행어 (Buzzword)의 수사적 기능 분석을 중심으로**. 서울대학교 대학원.
- 문미예, 김갑수. (2018). 초등학생을 위한 Python 프로그래밍 언어 교육방안 연구. **한국정보교육학회 학술논문집**, 9(1), 33~41.
- 박대륜. (2018). **초등학생을 위한 로봇 활용 파이썬 학습 프로그램 개발**. 대구교육대학교 교육대학원.
- 배성중(2018). **컴퓨팅 사고능력 향상을 위한 파이썬 프로그래밍 학습방법 연구**. 동국대학교 교육대학원.
- 변상용. (2004). **쉽게 배우는 자료구조**. 서울: 홍릉과학출판사.
- 소미현. (2017). **블록형 프로그래밍 학습에서 텍스트형 프로그래밍 학습으로의 전이**. 고려대학교 교육대학원.

- 소프트웨어야 놀자. (2018). **CS언플러그드:가벼운 것부터 무거운 것까지 - 정렬 알고리즘**. (<https://tv.naver.com/playsw>)
- 위키백과. (2018). **파이썬**. (<https://ko.wikipedia.org/wiki/>).
- 유진아. (2008). **공개소프트웨어 Python을 이용한 프로그래밍 교육에 관한 연구**. 단국대학교 교육대학원.
- 이대현. (2015). 인터랙티브 미디어의 효율적 구현을 위한 대안 연구 - 파이썬을 활용한 게임 개발 교육 사례를 중심으로. **한국영상학회지**, 13(1). 145~156.
- 이지연. (2008). **교육방법 및 교육공학**. 서울: 서현사.
- 이훈영. (2010). **이훈영 교수의 통계학**. 서울: 청람
- 임민영. (2006). **초등학교 컴퓨터교육에서 검색과 정렬 알고리즘 학습가능성에 관한 연구**. 진주교육대학교 교육대학원.
- 전산용어사전편찬위원회. (2005). **컴퓨터 인터넷 IT용어 대사전**. 서울: 일진사.
- 전용호. (2016). **구성주의 학습이론을 적용한 프로그래밍 언어 교육에 관한 연구**. 연세대학교 교육대학원.
- 한중하. (1991). **과학적 사고 기능 시험**. 한국교육개발원.
- 함성진. 김순화. 박세영. 송기상. (2014). 융합적사고력 신장을 위한 초등학생용 CT기반 융합인재교육(CT-STEAM) 프로그램 개발. **한국컴퓨터교육학회 학술논문집**. 17(6). 81~91.
- Aditya Bargava. (2017). **그림으로 개념을 이해하는 알고리즘**(김도형 역). 서울: 한빛미디어. (원저 2016 출판)
- Brennan, K., & Resnick, M. (2013). Imagining, Creating, Playing, Sharing, Reflecting: How Online Community Supports Young People as Designers of Interactive Media. In C. Mouza and N. Lavigne(eds.), *Emerging Technologies for Classroom*, Springer. 253-268.
- Codeeval. (2016). **Most Popular Coding Languages of 2016**. (<http://blog.codeeval.com>).
- Klaus Schwab. (2016). *Shaping the Fourth Industrial Revolution*.

New present publishing co.

Myers Brad A. (1990). Taxonomies of visual programming and program visualization. **Journal of Visual Languages & Computing**, 1(1). 97-123.

Philip Guo. (2014. 7. 7.). Python Is Now the Most Popular Introductory Teaching Language at Top U.S Universities. **Communications of the ACM**. Retrieved November 18, 2018, from <https://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-u-s-universities/fulltext>

TIOBE. (2018. 9. 30.). TIOBE Index for September 2018. Retrieved November 18, 2018, from <https://www.tiobe.com/tiobe-index/>

A B S T R A C T *

Development and Application of Algorithm Execution Time Analysis Education Program for Elementary School Students: Focusing on Python Sort Algorithm

Moon, Woo Jong

Major in Elementary Computer Education
Graduate School of Education
Jeju National University

Supervised by Professor Kim, Jong Hoon

This study developed and applied algorithm execution time analysis education program specifically focusing on python sort algorithm for developing thinking ability of elementary school students, and verified its effectiveness. In order to verify the effectiveness of the education program developed, the 42nd session was conducted for six days for 25 applicants based on the candidate list of the elementary school education donation program. The results of pre- and post-mortem tests were used to analyze educational effects by utilizing 'logical thinking test(GALT)' and 'bebras challenge 2017' developed by the Korea Educational Development Institute, and the results showed that algorithm execution time analysis education program which focusing on python sort algorithm, is effective in improving logical thinking and computational thinking skills of elementary school students.

* A thesis submitted to the committee of Graduate School of Education, Jeju National University in partial fulfillment of the requirements for the degree of Master of Education conferred in April, 2019.

부 록

[부록 1] 교육 프로그램 교재

[부록 2] 교육 프로그램 과정안

[부록 3] 사전 설문조사 문항

[부록 1] 교육프로그램 교재

<교육기부 프로그램> - 꿈을 키우는 초등 창의 컴퓨터 교실(14기)



수행시간 중심의 파이썬(Python) 정렬 알고리즘 교육

_____초등학교

이름 _____

교육기간 2018. 8. 13.(월) - 8. 18.(토)

강사 문우종

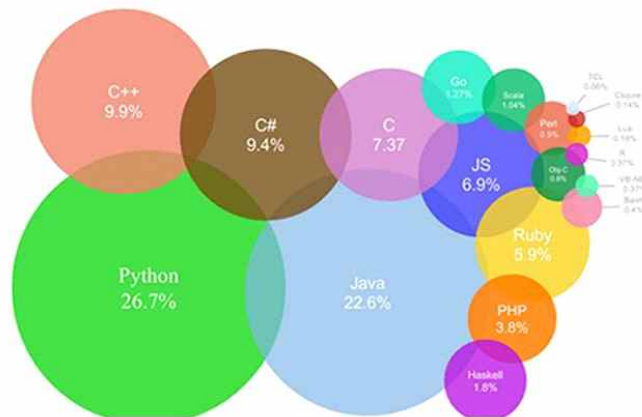
제0강 파이썬이란?

파이썬(python)은 프로그램을 설계하는 프로그래밍 언어 중 하나입니다.

파이썬은 1991년, 네덜란드 수학자 귀도 반 로섬에 의해 개발되었습니다.

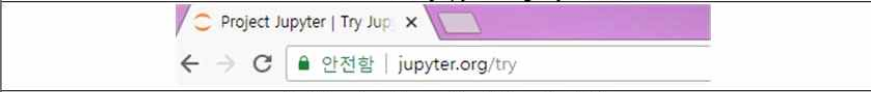
프로그래밍 언어에는 파이썬 뿐만 아니라 C, 자바, 자바 스크립트 등 여러 가지가 있지만 파이썬이 가장 널리 사용됩니다. 코드 경진대회 서비스를 제공하는 [코드이벨]은 파이썬을 '2016년 프로그래밍 인기 순위 1위'로 꼽기도 했습니다.

Most Popular Coding Languages of 2016

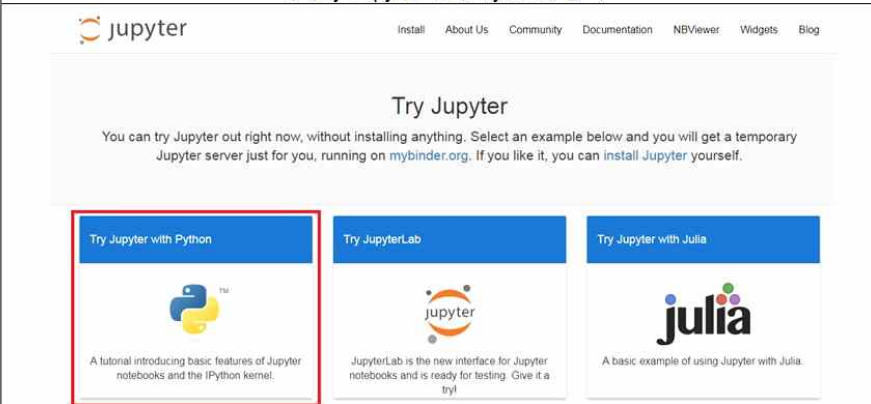


이처럼 파이썬이 널리 사용되고 있는 것은 간결하고 쉽다는데서 그 원인을 찾을 수 있습니다. 문법이 다른 프로그래밍 언어에 비해 어렵지 않을 뿐 아니라 표현하는 구조도 사람이 대화하는 형식을 이용함으로써 초보자도 쉽게 배울 수 있다는 강점이 있습니다. 파이썬은 웹 개발뿐만 아니라 데이터 분석, 머신러닝, 그래픽, 학술 연구 등 여러 분야에서 활용되고 있습니다.

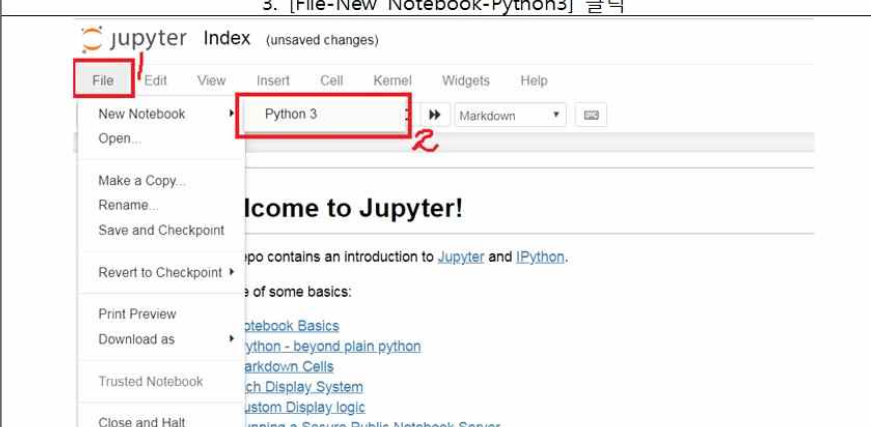
1. Chrome에서 'jupyter.org/try' 입력



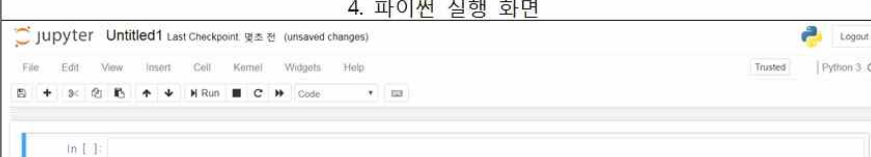
2. [Try Jupyter with Python] 클릭



3. [File-New Notebook-Python3] 클릭



4. 파이썬 실행 화면



제 1 강 연산과 변수

☆ 연산 기호 알아보기

8+3	[실행] 13 5 24 1.75 1 3 8
8-3	
8*3	
7/4	
7//4	
7%4	
2**3	

[미션] 실행 결과를 보고 연산 기호의 기능을 정리하여 보시오.

+		//	
-		%	
*		**	
/			

☆ 변수의 개념

a = 1 b = 2 a+b	[실행] 3
변수는 '변할 수 있는 수'라는 뜻입니다. 변수에는 숫자나 문자를 넣어 보관했다가 필요할 때 호출하여 사용할 수 있습니다. 변수에는 하나의 값만 저장할 수 있으므로 여러 개의 변수를 사용하려면 이름을 따로 붙여야 합니다.	

[미션] 세 개의 변수에 3, 4, 5를 각각 저장하여 세 변수의 합을 구하는 프로그램을 작성하시오.

☆ 정보 입력하기

name = input("당신의 이름은: ") print("안녕하세요", name)	[실행] 당신의 이름은: 홍길동 안녕하세요 홍길동
input을 사용하면 사용자가 정보를 입력하기를 기다립니다. 입력받은 정보는 앞의 변수에 저장됩니다.	

[미션] 두 개의 변수에 사용자가 원하는 숫자를 입력하여 더하는 프로그램을 작성하시오.

☆ 센티미터 단위의 길이를 미터와 센티미터로 변환하기

```
cm = int(input('센티미터 단위 길이: '))
m = cm // _____
cm = cm % 100
print(m, '미터', cm, '센티미터')
```

[실행]
센티미터 단위 길이: 130
1 미터 30 센티미터

[미션] 미터 단위의 길이를 입력받아 킬로미터와 미터로 변환하는 프로그램을 작성하시오.

☆ 초 단위의 시간을 시간, 분, 초로 변환하기

```
s = int(input('초 단위 시간: '))
m = s // _____
s = s % _____
print(m, '분', s, '초')
```

[실행]
초 단위 시간: 200
3분 20초

[미션] 초 단위 시간을 입력받아 '시간', '분', '초'로 변환하는 프로그램을 작성하시오.

☆ 조건문(if, else)의 이해

집에서 학교까지 택시를 타고 가려면 3000원이 필요합니다. 가지고 있는 돈을 입력했을 때 택시를 탈 수 있는지 확인하는 프로그램을 만들어 봅시다.

```
money = int(input('가지고 있는 돈: '))
if money >= _____:
    print("택시를 탈 수 있습니다.")
else:
    print("택시를 타지 못합니다.")
```

[실행]
가지고 있는 돈: 4000
택시를 탈 수 있습니다.
가지고 있는 돈: 2000
택시를 타지 못합니다.

☆ 두 수 중 큰 수 찾기

```
a = int(input('수1: '))
b = int(input('수2: '))
if a>b:
    print('큰 수: ', a)
else:
    print('큰 수: ', b)
```

[실행]
수1: 12
수2: 34
큰 수: 34

[미션] 두 수의 차를 구하는 작성하시오.

☆ 홀수 짝수 판별하기

<pre>a = int(input('자연수: ')) if a%2 == _____: print(a, ':짝수') else: print(b, ':홀수')</pre>	<p>[실행]</p> <p>자연수: 15</p> <p>15: 홀수</p> <p>자연수: 6</p> <p>6: 짝수</p>
---	---

[미션] 3의 배수를 판별하는 프로그램을 작성하시오.

☆ 나이에 따른 입장료 구하기

동물원에 입장할 때 '8세 미만은 무료', '8세 이상에서 60세 미만은 5천원', '60세 이상은 무료'라고 합니다. 나이를 입력하면 입장료를 알려주는 프로그램을 작성해 봅시다.

<pre>age = int(input('나이: ')) if age<8 _____ age >=60: print('무료') else: print('5000원')</pre>	<p>[실행]</p> <p>나이: 7</p> <p>무료</p> <p>나이: 20</p> <p>5000원</p>
---	---

조건을 판단하기 위해 사용하는 연산자에는 or, and, not이 있습니다.

A or B	A와 B 둘 중에 하나만 참이면 참이다.
A and B	A와 B 모두 참이어야 참이다.
not A	A가 거짓이면 참이다.

[미션] 위와 동일하게 동작하는 프로그램을 작성하시오.

<pre>age = int(input('나이: ')) if _____: print('5000원') else: print('무료')</pre>
--

[개별과제1]
선택구조(if, else)나 논리 연산(or, and, not)을 이용한 프로그램을 작성하시오.

제2강 반복 구조와 리스트

☆ 1부터 10까지 출력하기(while을 사용한 반복)

<pre>i = 1 while i<=10: print(i, end=' ') i=i+1</pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">[실행]</div> 1 2 3 4 5 6 7 8 9 10
<pre>i = 0 while _____: i=i+1 print(i, end=' ')</pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">[실행]</div> 1 2 3 4 5 6 7 8 9 10
while 조건: 문장	while은 조건이 참인 동안 아래 문장을 반복해서 실행한다.

[미션] 10부터 5까지 출력하는 프로그램을 작성하시오.

☆ 1부터 10까지 출력하기(for를 사용한 반복)

<pre>for i in range(10): print(i+1, end=' ')</pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">[실행]</div> 1 2 3 4 5 6 7 8 9 10
<pre>for i in range(1, 11, 1): print(i, end=' ')</pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;">[실행]</div> 1 2 3 4 5 6 7 8 9 10
range(5): 0, 1, 2, 3, 4	range(a)는 0부터 'a-1'까지의 수
range(1, 5, 2): 1, 3, 5, 7	range(a, b, c)은 'a'부터 'b-1'까지 'c'만큼 수
[연습] 각 범위에 해당하는 숫자를 적어보세요. range(4): range(0, 4): range(1, 5, 2):	

[미션] 10부터 5까지 출력하는 프로그램을 작성하시오.

☆ 1부터 10까지의 합 구하기

<pre>sum = 0 for i in range(____, ____): sum = sum + i print(sum)</pre>	[실행] 55
<pre>sum = 0 i = 0 while ____: sum = sum + i i = i + 1 print(sum)</pre>	[실행] 55

[미션] 1부터 '입력받은 수'까지의 합을 구하는 프로그램을 작성하시오.

☆ 1부터 20까지의 합 구하기

<pre>sum = 0 for i in range(____, ____): sum = sum + 1 print(sum)</pre>	[실행] 55
<pre>sum = 0 i = 0 while ____: sum = sum + i i = i + 1 print(sum)</pre>	[실행] 1 2 3 4 5 6 7 8 9 10

[미션] 1부터 '입력받은 수'까지의 합을 구하는 프로그램을 작성하시오.

☆ 약수 구하기

<pre>n = int(input('정수: ')) for i in range(1, n+1): if ____=0: print(i, end=' ')</pre>	[실행] 55
--	------------

[미션] 약수와 약수의 개수를 구하는 프로그램을 작성하시오.

☆ 임의의 수의 덧셈(random의 활용)

<pre>import random a = random.randint(1, 10) b = random.randint(1, 10) print(a, '+', b, '=') c = int(input()) if a + b == c: print('정답') else: print('땡!')</pre>	<p>[실행]</p> <p>3 + 5 =</p> <p>8</p> <p>정답</p>
<p>import random 명령어를 사용하여 random을 사용하겠다고 알리고, random.randint(a, b)를 입력하면 a에서 b까지의 수 중 하나를 임의로 출력합니다.</p>	

☆ 숫자 맞추기 게임(random의 활용)

<pre>import random n = random.randint(1, 10) while True: a = int(input("이 숫자는?: ")) if a == n: print("정답") break if a < n: print("UP") if a > n: print("DOWN")</pre>	<p>[실행]</p> <p>이 숫자는?: 5</p> <p>DOWN</p> <p>이 숫자는?: 3</p> <p>UP</p> <p>이 숫자는?: 4</p> <p>정답</p>
<p>while True: 문장</p>	<p>조건이 참인 동안 문장을 계속 실행 → 영원히 반복</p>
<p>break</p>	<p>반복을 멈추어 주는 명령어</p>
<p>단순하게 정해진 횟수를 반복할 때는 for 명령어가 유용하지만, 반복을 계속할지 중단할지 판단하거나 무한 반복을 해야 할 때는 while 명령어가 유용하다.</p>	

☆ 리스트에서 항목 랜덤하게 선택하기

<pre>import random food = ['짜장면', '짬뽕', '볶음밥'] select = random.choice(food) print(select)</pre>	<div style="border: 1px solid black; padding: 2px;">[실행] 짜장면</div>
<p>데이터를 저장할 때 변수에는 하나의 값밖에 저장하지 못합니다. 리스트는 변수와는 다르게 여러 개의 값을 저장할 수 있습니다. 리스트에 저장하는 값은 대괄호 [] 로 표현합니다. 리스트 안에 있는 값은 0부터 번호가 매겨집니다.</p>	

☆ 리스트 탐색하기

<pre>a = [10, 20, 30, 40, 50] key = int(input('키: ')) if key in a: print(a.index(key), '에서 탐색') else: print('실패')</pre>	<div style="border: 1px solid black; padding: 2px;">[실행] 키: 50 4 에서 탐색</div>
---	--

[미션] 국가 이름을 입력하면 수도를 알려주는 프로그램을 작성하시오.

국가	한국	미국	영국
수도	서울	워싱턴	런던

☆ 리스트를 이용한 퀴즈 만들기

<pre>pro = ['동물의 왕은?', '새 중의 왕은?', '바다의 왕은?'] answer = ['사자', '독수리', '고래'] for i in range(len(pro)): ans = input(pro[i]) if ans == answer[i]: print('정답!') else: print('땡')</pre>	<div style="border: 1px solid black; padding: 2px;">[실행] 동물의 왕은?사자 정답! 새 중의 왕은?독수리 정답! 바다의 왕은?뱀 땡</div>
---	---

[개별과제2]
반복구조(for, while)를 이용한 프로그램을 작성하시오.

☆ 입력받은 세 수 중 가장 큰 수 찾기(1)

```
a = int(input('자연수1: '))
b = int(input('자연수2: '))
c = int(input('자연수3: '))

if a>b:
    if a>c:
        print('큰 수:', a)
    else:
        print('큰 수:', c)
elif b>c:
    print('큰 수:', b)
else:
    print('큰 수:', c)
```

[실행]
자연수1: 5
자연수2: 2
자연수3: 8
큰 수: 8

☆ 입력받은 세 수 중 가장 큰 수 찾기(2)

```
a = int(input('자연수1: '))
b = int(input('자연수2: '))
c = int(input('자연수3: '))

maxx = a
if b>maxx:
    maxx = b
if c>maxx:
    maxx = c

print('큰 수:', maxx)
```

[실행]
자연수1: 5
자연수2: 2
자연수3: 8
큰 수: 8

위의 두 프로그램은 입력받은 세 수의 크기를 비교하여 가장 큰 수를 출력하는 프로그램입니다. 같은 숫자를 입력하면 같은 결과를 출력하지만 프로그램을 비교해보면 차이가 있습니다. 위 프로그램 중 어떤 것이 더 효율적일까요? (실행 과정을 적어봅시다.)

[미션] 네 개의 수를 비교하는 프로그램을 작성하시오.

☆ 20초 맞추기 게임 프로그램

```
import time
input("엔터를 누르고 20초를 셉니다.")
start = time.time()

input("20초 후에 다시 엔터를 누릅니다.")
end = time.time()

et = end - start
print("실제 시간 :", et, "초")
print("차이 :", abs(et - 20), "초")
```

[실행]
엔터를 누르고 20초를 셉니다.
20초 후에 다시 엔터를 누릅니다.
실제 시간 : 4.869029760360718 초
차이 : 15.130970239639282 초

☆ 타자 게임 프로그램

```
import random
import time
w = ["cat", "dog", "fox", "monkey", "mouse", "panda", "frog", "snake", "wolf"]
n = 1
print("[타자 게임] 준비되면 엔터!")
input()
start = time.time()
q = random.choice(w)
while n <= 3:
    print("*문제", n)
    print(q)
    x = input()
    if q == x:
        print("통과!")
        n = n + 1
        q = random.choice(w)
    else:
        print("오타! 다시 도전!")
end = time.time()
et = end - start
et = format(et, ".2f")
print("타자 시간 :", et, "초")
```

[실행]
[타자 게임] 준비되면 엔터!

*문제 1
wolf
wolf
통과!
*문제 2
cat
cat
통과!
*문제 3
wolf
wolf
통과!
타자 시간 : 3.03 초

☆ 계산 게임 프로그램

```
import random
def make_question():
    a = random.randint(1, 10)
    b = random.randint(1, 10)
    op = random.randint(1, 3)
    q = str(a)
    if op == 1:
        q = q + "+"
    if op == 2:
        q = q + "-"
    if op == 3:
        q = q + "*"
    q = q + str(b)
    return q

sc1 = 0
sc2 = 0

for x in range(3):
    q = make_question()
    print(q)
    ans = input("=")
    r = int(ans)

    if eval(q) == r:
        print("정답!")
        sc1 = sc1 + 1
    else:
        print("오답!")
        sc2 = sc2 + 1

print("정답 :", sc1, "오답 :", sc2)
if sc2 == 0:
    print("대단하네요!")
```

[실행]
2*9
= 18
정답!
7*10
= 70
정답!
10-6
= 4
정답!
정답 : 3 오답 : 0
당신은 천재입니다!

☆ 10, 20, 30, ... , 100 리스트에 저장하고 거꾸로 출력하기

```
a = []
for i in range(_____):
    a.append(i)
for i in range(_____):
    print(a[i], end=' ')
```

[실행]
100 90 80 70 60 50 40 30 20 10

☆ 중첩 반복문

```
for i in range(1, 4):
    for j in range(1, 3):
        print(i, j)
```

[실행]
1 1
1 2
2 1
2 2
3 1
3 2

☆ 직각삼각형 모양으로 수 출력하기

```
for i in range(1, 6):
    for j in range(1, _____):
        print(_____, end=' ')
    print()
```

[실행]
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

[미션] 실행 결과가 각각 다음과 같이 출력되는 프로그램을 작성하시오.

[실행] 1 2 3 3 4 5 4 5 6 7 5 6 7 8 9	[실행] 1 2 3 4 5 1 2 3 4 1 2 3 1 2 1	[실행] 5 4 3 2 1 4 3 2 1 3 2 1 2 1 1
---	---	---

☆ 리스트 항목 거꾸로 뒤집기

```
a = [x*10 for x in range(1, 11)]
for i in range(0, ____):
    temp = a[i]
    a[i] = _____
    _____ = temp
print(a)
```

[실행]
[100, 90, 80, 70, 60,
50, 40, 30, 20, 10]

☆ 리스트 수정하기

```
a = [10, 30, 30]
a[1] = 20
print(a)

del(a[1])
print(a)

a.insert(1, 20)
print(a)

a.remove(30)
print(a)
```

[실행]
[10, 20, 30]
[10, 30]
[10, 20, 30]
[10, 20]

☆ 리스트 최댓값 구하기

```
import random
a = [random.randint(1, 100) for x in range(10)]
print(a)
m = a[0]
for i in range(1, 10):
    if ____:
        m = a[i]
print('최댓값', m)
```

[실행]
[47, 20, 23, 32, 34, 9,
62, 9, 72, 56]
최댓값 72

[개별과제3]
리스트를 이용한 프로그램을 작성하시오.

☆ 숫자 야구 게임 프로그램

<pre> import random # 게임을 위한 랜덤 숫자 생성 rn = ["0", "0", "0"] rn[0] = str(random.randrange(1, 9, 1)) rn[1] = rn[0] rn[2] = rn[0] while (rn[0] == rn[1]): rn[1] = str(random.randrange(1, 9, 1)) while (rn[0] == rn[2] or rn[1] == rn[2]): rn[2] = str(random.randrange(1, 9, 1)) t_cnt = 0 # 시도횟수 s_cnt = 0 # 스트라이크 갯수 b_cnt = 0 # 볼 갯수 print("숫자야구게임을 시작합니다 !!!") print("-----") while (s_cnt < 3): num = str(input("숫자 3자리를 입력하세요 : ")) s_cnt = 0 b_cnt = 0 for i in range(0, 3): for j in range(0, 3): if(num[i] == str(rn[j]) and i == j): s_cnt += 1 elif(num[i] == str(rn[j]) and i != j): b_cnt += 1 print("결과 : [", s_cnt, "] Strike [", b_cnt, "] Ball") t_cnt += 1 print("-----") print(t_cnt, "번 만에 정답을 맞추셨습니다.") </pre>	<p>[실행]</p> <p>숫자야구게임을 시작합니다 !!!</p> <p>-----</p> <p>숫자 3자리를 입력하세요 : 123</p> <p>결과 : [0] Strike [0] Ball</p> <p>숫자 3자리를 입력하세요 : 456</p> <p>결과 : [2] Strike [0] Ball</p> <p>숫자 3자리를 입력하세요 : 457</p> <p>결과 : [3] Strike [0] Ball</p> <p>-----</p> <p>3 번 만에 정답을 맞추셨습니다.</p>
---	---

제3강 정렬을 통한 알고리즘 수행시간 분석

☆ 가장 큰 수와 가장 작은 수 찾기

<보기1> ○○ 초등학교 5학년 1반 학생들의 키

140	161	137	125	163	151	127	139	144	154
152	160	128	141	136	126	166	165	140	129
135	143	164	150	155	138	142	162	153	145

- 1) <보기1>에서 '가장 키가 큰 학생의 키'와 '가장 키가 작은 학생의 키'를 찾아보세요.
- 2) <보기1>에서 10번째로 키가 큰 학생의 키를 적어보세요.

<보기2> ○○ 초등학교 5학년 2반 학생들의 키

125	126	127	128	129	131	132	133	134	135
136	137	138	139	140	145	146	147	148	149
151	152	154	155	156	160	161	162	164	165

- 3) <보기2>에서 '가장 키가 큰 학생의 키'와 '가장 키가 작은 학생의 키'를 찾아보세요.
- 4) <보기2>에서 10번째로 키가 큰 학생의 키를 적어보세요.
- 5) <보기1>과 <보기2> 중 어떤 것이 문제를 해결하기 쉬웠나요? 그 이유는 무엇일까요?

<정렬이란?> 주어진 자료를 어떤 기준에 의하여 순서대로 나열하는 것
데이터는 무작위로 나열되었을 때보다 어떤 규칙에 따라 순서대로 정렬되어 있을 때 훨씬 유용하게 사용될 수 있습니다. 컴퓨터에서는 수많은 파일을 이름, 크기, 날짜순으로 정렬 가능하며 여러 가지 데이터도 규칙에 따라 정렬하는 것이 가능합니다.
정렬 방법도 여러 종류가 있는데 어떤 정렬 방법을 사용하느냐에 따라 컴퓨터의 처리 속도(알고리즘의 수행시간)가 달라집니다.

☆ 정렬 방법 체험하기(1) - 선택 정렬



- 1) 평형 저울을 이용하여 필름통 2개를 비교해가며 가장 가벼운 필름통을 찾습니다.
평형 저울을 한 번 쓸 때마다 카운터를 눌러 평형 저울 사용 횟수를 셉니다.
※ 한 번에 용기 2개의 무게만 비교합니다. 반드시 평형 저울을 사용해 비교합니다.
- 2) 가장 가벼운 필름통을 찾으면 아래 활동지의 ㉠에 놓고, 남은 필름통 4개 중에 가장 가벼운 필름통을 찾아 ㉡에 놓습니다.
- 3) 위의 과정을 반복하여 가벼운 것부터 무거운 것까지 순서대로 활동지에 놓습니다.
- 4) 정렬이 잘 되었는지 필름통을 열어 확인합니다.
- 5) 정렬이 잘 되었으면 카운터 수의 합계를 구해봅니다.

<활동지>

	가벼움	< ----- >	무거움	
용기	㉠	㉡	㉢	㉣
비교 횟수				

총 비교 횟수:

『네이버 TV, 소프트웨어야 놀자 <https://terms.naver.com/entry.nhn?docId=4369097&cid=59931&categoryId=59931>』

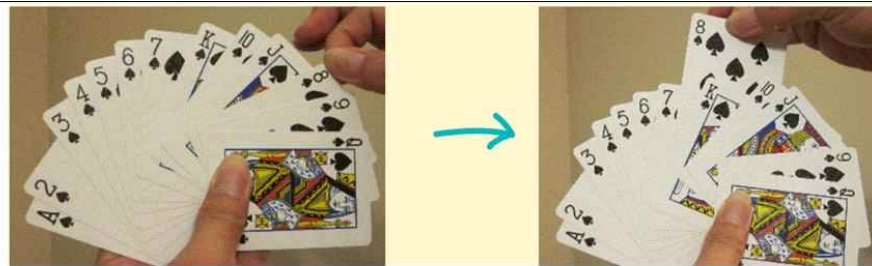
☆ 정렬 프로그래밍(1) - 선택 정렬

선택 정렬(selection sort)은 컴퓨터가 사용하는 정렬 방법의 하나로, 정렬대상이 위치한 곳과 정렬 결과를 놓을 곳으로 나누고, 가장 가벼운 것을 골라 정렬결과 장소를 옮겨놓습니다. 그 다음으로 가벼운 것을 골라 옮겨 놓는 것을 반복하여 더는 정렬 대상에 아무것도 없을 때까지 반복합니다.

a	최소 위치	b	컴퓨터의 수행 절차
0	0	1	0번의 값을 임시 저장하고 1~4번과 비교해서 가장 작은 값을 0번에 저장하고, 임시 저장된 수를 바꾼 자리의 수에 다시 넣는다.
		2	
		3	
		4	
1	1	2	1번의 값을 임시 저장하고 2~4번과 비교해서 가장 작은 값을 1번에 저장하고, 임시 저장된 수를 바꾼 자리의 수에 다시 넣는다.
		3	
		4	

a	최소 위치	b	컴퓨터의 수행 절차
2	2	3	<p>첫 번째, 두 번째 데이터를 제외한 나머지 데이터에서 가장 작은 데이터인 8을 세 번째 데이터인 15와 교환한다.</p>
		4	
3	3	4	<p>첫 번째, 두 번째, 세 번째 데이터를 제외한 나머지 데이터에서 가장 작은 데이터인 11을 네 번째 데이터인 15와 교환한다.</p>
			<pre> import random def selection_sort(A): for i in range(____): for j in range(i+1, len(a)): if ____ > ____: a[i], a[j] = _____ return a a = [random.randint(0,100) for k in range(5)] print(a) selection_sort(a) print(a) </pre> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"> <p>[실행] [84, 6, 56, 1, 31] [1, 6, 31, 56, 84]</p> </div>

☆ 정렬 방법 체험하기(2) - 삽입 정렬



<무작위로 섞여 있는 카드를 왼손에 잡고 순서대로 정리하는 게임>

- 1) 트럼프 카드에서 같은 모양 13장의 카드를 꺼낸 다음 무작위로 섞어서 뒤집어 놓습니다.
- 2) 카드를 A부터 K까지 정리하는 전략을 생각해보고 아래 활동지에 기록합니다.
(카드 순서: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K)
- 3) 선생님의 신호에 따라 왼손에 카드를 잡고 오른손으로만 카드를 정리합니다.
- 4) 정리가 완료되면 카드를 내려놓고, 스톱워치를 정지시켜 기록을 확인합니다.
- 5) 정리를 효과적으로 할 수 있는 전략을 찾아봅시다.

<활동지>

전략	기록

☆ 정렬 프로그래밍(2) - 삽입 정렬

삽입정렬은 새로운 데이터를 이미 정렬된 데이터들 사이의 적절한 자리에 집어넣는 정렬방법입니다.

a	key	b	컴퓨터의 수행 절차
1	1번 (11)	0	<p>두 번째 데이터 11과 바로 앞에 위치한 15의 크기를 비교한다. 두 번째 데이터가 작으므로 15를 한 칸 뒤로 보내고 11을 15 앞에 위치시킨다.</p> <p>The diagram illustrates the insertion of the key 11 into the array [15, 11, 1, 3, 8]. It shows 11 being compared with 15, shifted to the right, and then inserted before 15, resulting in the array [11, 15, 1, 3, 8].</p>
2	2번 (1)	1	<p>세 번째 데이터인 1과 앞에 위치한 11, 15의 크기를 비교한다. 1이 가장 작으므로 11과 15를 한 칸씩 뒤로 보내고 1을 가장 앞에 위치시킨다.</p> <p>The diagram illustrates the insertion of the key 1 into the array [11, 15, 1, 3, 8]. It shows 1 being compared with 11 and 15, shifted to the left, and then inserted at the beginning, resulting in the array [1, 11, 15, 3, 8].</p>

a	key	b	컴퓨터의 수행 절차
3	3번 (3)	2	<p>네 번째 데이터인 3과 앞에 위치한 1, 11, 15의 크기를 비교한다. 3은 1보다 크고 11, 15보다 작으므로 11과 15를 한 칸씩 뒤로 보내고 3을 11 앞에 위치시킨다.</p>
4	4번 (8)	3	<p>마지막 데이터인 8과 앞에 위치한 1, 3, 11, 15의 크기를 비교한다. 8은 1, 3보다 크고 11, 15보다 작으므로 11과 15를 한 칸씩 뒤로 보내고 8을 11 앞에 위치시킨다.</p>
			<pre> import random ds = [random.randint(1, 100) for x in range(5)] print(ds) for a in range(1, _____): key = ds[a] b = a-1 while b>=0 and ds[b]>_____ ds[b+1] = ds[b] b = b-1 ds[b+1] = _____ print(ds) print(a) </pre> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto; margin-right: auto;"> <p>[실행] [25, 22, 88, 64, 3] [3, 22, 25, 64, 88]</p> </div>

☆ 정렬 방법 체험하기(3) - 버블 정렬

<무작위로 섞여 있는 카드를 버블을 이용하여 오름차순으로 정리하는 게임>

『양나라 외, 컴퓨터 없이 배우는 어린이 코딩, 계림북스(2016)』

- 1) 메모리 카드를 뒤집은 다음 잘 섞어 일렬로 놓습니다.
- 2) 왼쪽에서 처음과 두 번째 칸에 놓인 카드 2장을 뒤집어 숫자를 비교합니다.
- 3) 숫자가 작은 카드를 왼쪽에, 숫자가 큰 카드를 오른쪽에 둡니다.
- 4) 숫자가 안 보이도록 카드를 다시 뒤집습니다.
- 5) 두 번째와 세 번째 카드를 뒤집어 숫자를 비교해 작은 수를 왼쪽에 둡니다.
- 6) 위와 같은 방식으로 일곱 번째와 여덟 번째 카드까지 비교합니다.
- 7) 다시 첫 번째와 두 번째 카드를 비교하여 자리바꿈이 없을 때까지 반복합니다.

☆ 정렬 프로그래밍(3) - 버블 정렬

버블정렬은 이웃하는 값끼리 크기를 비교하여 순서를 정렬하는 방법입니다.			
a	b	b+1	컴퓨터의 수행 절차
0	0	1	<p>첫 번째 데이터인 15와 두 번째 데이터인 11을 비교해 큰 데이터를 뒤로 위치시킨다. 15가 크므로 둘의 위치를 바꾼다.</p>
0	1	2	<p>두 번째 데이터인 15와 세 번째 데이터인 1을 비교하는데, 앞에 위치한 15가 크므로 둘의 위치를 바꾼다.</p>
0	2	3	<p>마찬가지 방식을 적용해 세 번째 데이터인 15와 네 번째 데이터인 3의 위치를 바꾼다.</p>

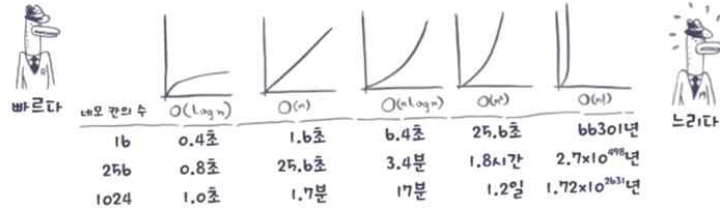
a	b	b+1	컴퓨터의 수행 절차
0	3	4	<p>마찬가지 방식을 적용해 네 번째 데이터인 15와 마지막 데이터인 8의 위치를 바꾼다. 가장 큰 데이터인 15가 가장 뒤에 위치하게 된다.</p>
1	0	1	<p>처음부터 다시 시작한다. 첫 번째 데이터인 11과 두 번째 데이터인 1의 크기를 비교하는데 앞에 위치한 11이 크므로 둘의 위치를 바꾼다.</p>
1	1	2	<p>마찬가지 방식을 적용해 두 번째 데이터인 11과 세 번째 데이터인 3의 위치를 바꾼다.</p>

a	b	b+1	컴퓨터의 수행 절차
1	2	3	<p>마찬가지 방식을 적용해 세 번째 데이터인 11과 네 번째 데이터인 8의 위치를 바꾼다. 두 번째로 큰 데이터인 11이 뒤에서 두 번째에 위치하게 된다.</p>
2	0	1	<p>처음부터 다시 시작한다. 첫 번째 데이터인 1과 두 번째 데이터인 3의 크기를 비교하는데 앞에 위치한 1이 작으므로 그대로 둔다.</p>
2	1	2	<p>두 번째 데이터인 3과 세 번째 데이터인 8의 크기를 비교하는데, 앞에 위치한 3이 작으므로 그대로 둔다. 세 번째로 큰 데이터인 8이 뒤에서 세 번째에 위치하게 된다.</p>
3	0	1	<p>처음부터 다시 시작한다. 첫 번째 데이터인 1과 두 번째 데이터인 3의 크기를 비교하는데, 앞에 위치한 1이 작으므로 그대로 둔다. 데이터들에 대한 정렬이 완료된다.</p>
			<pre>import random ds = [random.randint(1, 100) for x in range(5)] print(ds) for a in range(0, _____): for b in range(0, len(ds)-1-a): if ds[b] > ds[b+1]: ds[b], ds[b+1] = _____ print(ds)</pre> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin-left: auto;"> <p>[실행] [25, 22, 88, 64, 3] [3, 22, 25, 64, 88]</p> </div>

[정렬 알고리즘 프로그래밍 정리]

정렬	파이썬 프로그래밍	설명
선택정렬	<pre>import random def selection_sort(A): for i in range(_____): for j in range(i+1, len(a)): if ____ > ____: a[i], a[j] = _____ return a a = [random.randint(0,100) for k in range(5)] print(a) selection_sort(a) print(a)</pre>	정렬되지 않은 데이터들에 대해 가장 작은 데이터를 찾아 가장 앞의 데이터와 교환해나가는 방식
삽입정렬	<pre>import random ds = [random.randint(1, 100) for x in range(5)] print(ds) for a in range(1, _____): key = ds[a] b = a-1 while b>=0 and ds[b]>_____: ds[b+1] = ds[b] b = b-1 ds[b+1] = _____ print(ds) print(a)</pre>	아직 정렬되지 않은 임의의 데이터를 이미 정렬된 부분의 적절한 위치에 삽입해 가며 정렬하는 방식
버블정렬	<pre>import random ds = [random.randint(1, 100) for x in range(5)] print(ds) for a in range(0, _____): for b in range(0, len(ds)-1-a): if ds[b] > ds[b+1]: ds[b], ds[b+1] = _____ print(ds)</pre>	서로 이웃한 데이터들을 비교하며 가장 큰 데이터를 가장 뒤로 보내며 정렬하는 방식

☆ 그래프를 이용해서 알고리즘의 수행시간 알아보기



『아디트야 바르가바(2017), Hello Coding 그림으로 개념을 이해하는 알고리즘, 한빛미디어, p. 19』

어떤 알고리즘이 효율적인지 알아보기 위해서는 그래프를 그려서 기울기가 더 작은 것인지 알아보면 됩니다. 기울기가 클수록(왼쪽→오른쪽) 시간이 오래 걸리기 때문에 비효율적입니다.

[정렬 알고리즘의 수행시간 비교]

1. 가로축(데이터 개수-100, 200, 400, 800, 1600) / 세로축(30번 수행-실행시간(ms) 평균)
2. sortedData: 오름차순 데이터 / randomData: 임의 순서의 데이터 / reversedData: 내림차순 데이터

정렬	그래프	설명						
선택정렬		선택정렬은 데이터의 정렬 상태에 관계없이 늘 같은 시간이 소요된다. <table border="1"> <tr><td>오름차순</td><td>100(ms)</td></tr> <tr><td>임의순서</td><td>100(ms)</td></tr> <tr><td>내림차순</td><td>100(ms)</td></tr> </table>	오름차순	100(ms)	임의순서	100(ms)	내림차순	100(ms)
오름차순	100(ms)							
임의순서	100(ms)							
내림차순	100(ms)							
삽입정렬		정렬된 데이터는 삽입정렬이 매우 빠르다. <table border="1"> <tr><td>오름차순</td><td>5(ms)</td></tr> <tr><td>임의순서</td><td>125(ms)</td></tr> <tr><td>내림차순</td><td>250(ms)</td></tr> </table>	오름차순	5(ms)	임의순서	125(ms)	내림차순	250(ms)
오름차순	5(ms)							
임의순서	125(ms)							
내림차순	250(ms)							
버블정렬		데이터 크기가 커질수록 실행시간이 가파르게 증가한다. <table border="1"> <tr><td>오름차순</td><td>100(ms)</td></tr> <tr><td>임의순서</td><td>250(ms)</td></tr> <tr><td>내림차순</td><td>400(ms)</td></tr> </table>	오름차순	100(ms)	임의순서	250(ms)	내림차순	400(ms)
오름차순	100(ms)							
임의순서	250(ms)							
내림차순	400(ms)							

[최종과제]

선택정렬, 삽입정렬, 버블정렬 중 한 가지 정렬 방법을 이용하여 생활 속의 다양한 데이터(키, 몸무게, 성적, 생일 등)를 정렬하는 프로그램을 작성하시오.

- 발표일: 8월 18일(토)
- 임의 순서로 나열된 데이터 사용

[부록 2] 교육프로그램 과정안

일시	2018. 8. 13.(월)		대상	4~6학년 25명
주제	파이썬 기초		차시	1~7/42
활동명	변수, 연산자, 선택구조와 논리 연산	소요시간	300분	
학습목표	1. 파이썬의 변수와 연산자를 이용한 프로그램을 만들 수 있다. 2. 선택구조와 논리연산을 이용한 프로그램을 만들 수 있다.			
학습단계	학습요소	교수·학습 활동		
도입	오리엔테이션 및 사진검사	▷ 오리엔테이션 및 사전검사지 투입 - 논리적 사고력 검사지, 비버챌린지 2017 도입 - 교육프로그램 일정, 운영 방법 안내		
	학습목표 제시	▷ 변수, 연산자, 선택구조와 논리연산의 필요성을 알고 이를 활용한 프로그램을 만들 수 있다.		
전개	파이썬 기초	▷ 파이썬 설치 방법 - 에디터 설치 방법, Try Jupyter 사용법		
	연산자, 변수 익히기	▷ 연산 기호, 변수의 개념, 정보 입력받기 - 사용자가 원하는 숫자를 입력하여 변환하는 계산기 만들기 ▷ 단위의 변환 - 길이 단위, 시간 단위 변환해보기		
	선택 구조	▷ 홀수 짝수 판별기, 나이에 따른 입장료 구하기 프로그램 만들기 - 조건문(if, else)의 이해		
	논리 연산자	▷ 논리 연산자 or, and, not 익히기		
정리	학습내용 정리 및 과제 부여	▷ 연산자, 변수, 선택 구조, 논리 연산자 학습 내용 복습 ▷ 선택 구조, 논리 연산자를 활용한 자유 주제 프로그램 제작 과제 부여		

일시	2018. 8. 14.(화)	대상	4~6학년 25명
주제	반복 구조와 리스트	차시	8~14/42
활동명	반복 구조와 리스트의 활용	소요시간	300분
학습목표	1. 반복 구조의 필요성을 알고 반복 구조를 활용한 프로그램을 만들 수 있다. 2. 리스트의 필요성을 알고 리스트를 활용한 프로그램을 만들 수 있다.		
학습단계	학습요소	교수·학습 활동	
도입	동기 유발	▷ 반복이 필요한 상황(단순 명령어를 여러 번 입력해야 하는 상황), 리스트가 필요한 상황(변수에 하나의 값이 아닌 여러 개의 값을 저장해야 하는 상황)을 제시한다.	
	학습목표 제시	▷ 반복 구조, 리스트의 필요성을 알고 반복 구조와 리스트를 활용한 프로그램을 만들 수 있다.	
전개	반복구조 명령어 익히기	▷ 1부터 10까지 출력하는 프로그램 만들기 - while, for를 사용한 반복문 만들어보기 - while과 for 명령어의 차이점 생각해보기	
	반복구조 명령어 실습	▷ 1부터 20까지의 합 구하기, 약수 구하는 프로그램 만들기 - 문제에 더 적합한 반복 명령어를 사용하여 문제 해결하기 - 사용하지 않은 명령어를 사용하여 결과가 같은 프로그램 만들기 ▷ random 명령어를 사용한 프로그램 만들기 - for와 while이 각각 적합한 상황 알아보기	
	리스트 사용법 익히기	▷ 리스트 생성, 변수와의 차이 이해하기 - 대괄호 []를 이용한 리스트의 생성 - 변수와 리스트의 차이점 이해하기	
	리스트 실습	▷ 리스트를 이용한 퀴즈 프로그램, 타자 게임 프로그램 만들기 - 퀴즈, 타자 게임 프로그램을 만들어보며 다양한 값을 손쉽게 추 가하는 리스트의 장점, 활용법 익히기	
	수행시간 비교하기	▷ 입력받은 세 수 중 가장 큰 수를 찾는 두 개의 프로그램을 비교 해보고, 수행시간 차이점 생각해보기 - 반복 구조를 활용한 두 개의 프로그램을 비교해보며 수행시간의 차이, 알고리즘의 효율성에 대해 의견 나누기	
정리	학습내용 정리 및 과제 부여	▷ 반복 구조, 리스트 학습 내용 복습 ▷ 반복 구조, 리스트를 활용한 자유 주제 프로그램 제작 과제 부여	

일시	2018. 8. 15.(수)	대상	4~6학년 25명
주제	선택정렬, 삽입정렬	차시	15~21/42
활동명	선택정렬, 삽입정렬 알아보기	소요시간	300분
학습목표	1. 버블정렬, 선택정렬, 삽입정렬의 원리를 이해할 수 있다. 2. 버블정렬, 선택정렬, 삽입정렬을 활용한 프로그램을 만들 수 있다.		
학습단계	학습요소	교수·학습 활동	
도입	동기 유발	▷ 정렬이 필요한 상황을 제시한다.	
	학습목표 제시	▷ 정렬의 필요성을 알고 정렬을 활용한 프로그램을 만들 수 있다.	
전개	선택정렬 익히기	▷ 평형 저울을 이용해 필름통 무게 비교하기(언플러그드) - 용기 2개씩 무게를 비교하며 선택정렬의 원리 익히기 ▷ 선택정렬 진행과정을 도식화하여 프로그램 완성하기	
	삽입정렬 익히기	▷ 무작위 배열된 트럼프 카드를 정리하기(언플러그드) - 왼손으로 카드를 잡고 오른손으로 순서대로 정리하며 삽입정렬의 원리 익히기 ▷ 삽입정렬 진행과정을 도식화하여 프로그램 완성하기	
	선택정렬, 삽입정렬 비교하기	▷ 선택정렬, 삽입정렬 비교하기 - 선택정렬과 삽입정렬에 알맞은 상황 찾아보기 - 상황별로 선택정렬, 삽입정렬을 이용해 프로그램을 만들어보고 더 효율적인 프로그램 찾아보기	
정리	학습내용 정리 및 과제 부여	▷ 선택정렬, 삽입정렬 학습 내용 복습 ▷ 선택정렬, 삽입정렬을 활용한 자유 주제 프로그램 제작 과제 부여	

일시	2018. 8. 16.(목)	대상	4~6학년 25명
주제	버블정렬, 알고리즘 수행시간	차시	22~28/42
활동명	버블정렬, 알고리즘 수행시간 알아보기	소요시간	300분
학습목표	1. 버블정렬을 활용한 프로그램을 만들 수 있다. 2. 버블정렬, 선택정렬, 삽입정렬을 통해 수행시간을 비교하여 설명할 수 있다.		
학습단계	학습요소	교수·학습 활동	
도입	동기 유발	▷ 선택정렬, 삽입정렬과 다른 버블정렬이 사용되는 상황을 제시한다.	
	학습목표 제시	▷ 버블정렬을 활용한 프로그램을 만들어보고, 선택/삽입정렬과 비교해보며 수행시간을 비교하여 설명할 수 있다.	
전개	버블정렬 익히기	▷ 무작위 배열된 메모리 카드를 정리하기(언플러그드) - 왼쪽에서부터 메모리 카드를 2장씩 서로 비교하며 버블정렬의 원리 익히기 ▷ 버블정렬 진행과정을 도식화하여 프로그램 완성하기	
	수행시간 비교하기	▷ 그래프를 통한 정렬 알고리즘의 수행시간 비교해보기 - 그래프의 기울기를 관찰해보고 자료의 크기에 따른 효율적인 정렬 알고리즘에 대한 의견 나누기	
정리	학습내용 정리 및 과제 부여	▷ 버블정렬, 수행시간 학습 내용 복습 ▷ 버블정렬을 활용한 자유 주제 프로그램 제작 과제 부여	

일시	2018. 8. 17.(금)	대상	4~6학년 25명
주제	프로젝트 준비	차시	29~35/42
활동명	나만의 프로그램 만들기	소요시간	300분
학습목표	학습한 내용을 바탕으로 파이썬 프로그램을 제작할 수 있다.		
학습단계	학습요소	교수·학습 활동	
도입	동기 유발	▷ 교사가 만든 실생활에서 활용할 수 있는 파이썬 프로그램 (발표도 우미)을 제시한다.	
	학습목표 제시	▷ 학습한 내용을 바탕으로 파이썬 프로그램을 제작할 수 있다.	
전개	프로그램 계획서 작성하기	▷ 공부한 내용을 되돌아보며 실생활에서 유용한 프로그램 계획서 작성하기 - 브레인스토밍을 통한 프로그램 주제 아이디어 뽑기 - 주제 구현을 위한 프로그램 계획서 작성하기 - 계획서를 돌려 읽어보고 보완할 점 나누기	
	개별 프로그램 제작하기	▷ 계획서를 토대로 개인별 프로그램 제작하기 - 학습한 내용을 토대로 개인이 선택한 주제를 구현할 수 있는 프 로그램을 제작하기	
정리	학습내용 정리 및 과제 부여	▷ 반복 구조, 리스트 학습 내용 복습 ▷ 반복 구조, 리스트를 활용한 자유 주제 프로그램 제작 과제 부여	

일시	2018. 8. 18.(토)	대상	4~6학년 25명
주제	프로젝트 발표	차시	36~42/42
활동명	프로젝트 발표회	소요시간	300분
학습목표	직접 제작한 프로그램을 발표할 수 있다.		
학습단계	학습요소	교수·학습 활동	
도입	학습목표 제시	직접 제작한 프로그램을 발표할 수 있다.	
전개	사후검사	▷ 사후검사지 투입 - 논리적 사고력 검사지, 비버챌린지 2017 도입	
	프로그램 제작 발표회	▷ 프로그램 제작 발표회 - 사전 계획서, 프로그램의 용도 소개 - 사용한 명령어 소개 - 프로그램 실행 방법, 실행 결과 소개 - 프로그램을 만들며 느낀 점, 어려웠던 점 발표 ▷ 프로그램 검토 및 피드백	
정리	학습내용 정리 및 소감 나누기	▷ 교육프로그램 수강 이후 느낀 점 발표하기 ▷ 앞으로 소프트웨어 공부를 더 열심히 하도록 서로 격려하기	

[부록 3] 사전 설문조사 문항

초등학생 창의성 신장을 위한 교육프로그램 개발 및 적용을 위한 설문조사

본 설문은 초등학생을 대상으로 한 알고리즘 수행시간 분석 교육을 위한 설문입니다. 시간 내주셔서 감사드립니다.

(예상 설문 소요 시간: 2-3분)

* 필수항목

[응답자 기본 정보]

1. 1. SW교육(코딩교육)에 대하여 잘 알고 있습니까? *

한 개의 타원형만 표시합니다.

1 2 3 4 5

전혀 모른다. 잘 알고 있다.

2. 2. SW교육(코딩교육)에 대해 얼마나 관심을 가지고 계십니까? *

한 개의 타원형만 표시합니다.

1 2 3 4 5

전혀 관심 없다. 매우 관심이 많다.

3. 3. 학생 대상으로 SW교육(코딩교육)과 관련된 수업을 진행한 적이 있습니까? *

한 개의 타원형만 표시합니다.

- 있다. (3-1번 문항으로) 질문 4(으)로 건너뛰세요.
 없다. (3-2번 문항으로) 질문 5(으)로 건너뛰세요.

4. 3-1. SW교육(코딩교육)에서 자주 활용하시는 방법/도구를 하나만 골라주세요. *

한 개의 타원형만 표시합니다.

- 언플러그드 활동: SW보드게임, 활동지, 신체활동 등
 교육용 프로그래밍 언어(EPL): 스크래치, 엔트리, 엡인벤터 등
 프로그래밍 언어: 파이썬, C, 자바 등
 피지컬 컴퓨팅(로봇 활용): 햄스터봇, 오조봇, 마이크로비트 등

질문 6(으)로 건너뛰세요.

5. 3-2. SW교육(코딩교육) 관련 수업을 진행한다면 어떤 방법/도구를 사용하시겠습니까? *

한 개의 타원형만 표시합니다.

- 언플러그드 활동: SW보드게임, 활동지, 신체활동 등
- 교육용 프로그래밍 언어(EPL): 스크래치, 엔트리, 앱인벤터 등
- 프로그래밍 언어: 파이썬, C, 자바 등
- 피지컬 컴퓨팅(로봇 활용): 햄스터봇, 오조봇, 마이크로비트 등

질문 6(으)로 건너뛰세요.

6. 4. SW교육(코딩교육)을 위해 프로그래밍 언어(파이썬, C, 자바 등)를 가르칠 의향이 있으십니까? *

한 개의 타원형만 표시합니다.

1 2 3 4 5

전혀 없다. 꼭 가르쳐보고 싶다.

[주제 관련]

7. 5. 알고리즘에 대해 얼마나 잘 알고 계십니까? *

* 알고리즘: 어떠한 문제를 해결하기 위한 단계들을 정리한 것
한 개의 타원형만 표시합니다.

1 2 3 4 5

전혀 모른다. 잘 이해하고 있다.

8. 6. SW교육(코딩교육)에서 알고리즘이 차지하는 중요성이 얼마나 된다고 생각하십니까? *

한 개의 타원형만 표시합니다.

1 2 3 4 5

전혀 중요하지 않다. 매우 중요하다.

9. 7. 학생들에게 알고리즘의 수행시간에 대한 교육이 필요하다고 생각하십니까? *

* 알고리즘의 수행시간: 알고리즘이 수행되는 데 걸리는 시간(효율적인 알고리즘을 짜도록 지도)
한 개의 타원형만 표시합니다.

1 2 3 4 5

전혀 필요하지 않다. 꼭 필요하다.

10. 8. 알고리즘의 수행시간에 대한 교육을 할 때 어떤 어려움이 있을 것이라고 생각하십니까? *

한 개의 타원형만 표시합니다.

- 교육내용: 광범위한 주제
- 난이도: 초등학생이 이해하기 어려움
- 도구: 적합한 도구가 없음
- 흥미: 학생들의 반응, 몰입도 부족
- 시간: 많은 시간이 필요함

11. 9. 효율적인 알고리즘 작성을 위해 알고리즘의 수행시간을 교육한다면 어떤 내용을 교육하시겠습니까? *

복수 응답이 가능합니다.
해당 사항에 모두 표시하세요.

- 자료구조
- 정렬
- 탐색
- 알고리즘 이론(백트래킹, 분할정복법, 동적계획법 등)

12. 10. 효율적인 알고리즘 작성을 위해 알고리즘의 수행시간을 교육한다면 어떤 도구를 활용하시겠습니까? *

한 개의 타원형만 표시합니다.

- 언플러그드 활동: SW보드게임, 활동지, 신체활동 등
- 교육용 프로그래밍 언어(EPL): 스크래치, 엔트리, 앱인벤터 등
- 프로그래밍 언어: 파이썬, C, 자바 등
- 피지컬 컴퓨팅(로봇 활용): 햄스터봇, 오조봇, 마이크로비트 등

- 응답해 주셔서 감사합니다.

제공
 Google Forms