

한국정보올림피아드 초등부 경시부문 문제해결을 통한 알고리즘 교재 개발 및 적용

김병수 · 김종훈*

제주대학교

요 약

프로그래밍의 핵심은 어디까지나 알고리즘 학습에 있으며 이를 통한 창의적이고 논리적인 문제해결력의 향상이 프로그래밍 학습의 목표인 것이다. 그렇다면 어떤 알고리즘들을 어떠한 순서대로 가르치는가에 대한 고민을 좀 더 해 볼 필요가 있으며 그 효과성에 대해서도 연구해 볼 필요가 있을 것이다. 본 연구는 개념적 알고리즘의 내용들을 한국정보올림피아드 초등부 경시부문의 문제들을 이용하여 학습할 수 있도록 알고리즘 학습 교재를 개발하고 이 효과를 검증하였다.

키워드 : 한국정보올림피아드, 알고리즘, 프로그래밍

The Development and Implementation of an Algorithm Instructional Material through the Problem Solving on the KOI Final Test of Elementary Students

Byeong-Su Kim · Jong-Hoon Kim

Jeju National University

ABSTRACT

The core of programming learning is based on an algorithm learning and the promotion of problem solving abilities is the purpose of this learning. Then, we need to think about what kind of algorithms in what order we teach and we need to study the effect of this learning. The purpose of this study is development and implementation of algorithm instructional materials and examine the effect of an algorithm learning with conceptual algorithms in KOI(Korea Olympiad in Informatics) final test of elementary students.

Keywords : KOI, Korea Olympiad in Informatics, Algorithm, Programming

* 교신저자: 김종훈, 제주대학교 초등컴퓨터교육전공

논문투고: 2011-02-25

논문심사: 2011-02-28

심사완료: 2011-12-28

1. 연구의 필요성

교육정보화 및 컴퓨터 교육의 목표는 학습자로 하여금 디지털 정보 지식의 생산자이자 소비자가 되게 하는 것이지 단순한 정보 소비자를 만들자는 것이 아니다. 학생을 정보 생산자가 되게 하려면 학생에게 컴퓨터 관련 기능 기술과 함께 그 기능 기술이 있게 한 컴퓨터 논리, 디지털 논리, 더 나아가 디지털 문화 논리까지도 함께 가르쳐야 한다. 디지털 논리를 습득할 때 학생은 컴퓨터식 사고를 할 수 있게 되고, 컴퓨터식 사고를 할 수 있는 상태에서 습득한 기능·기술일 때 그 기능·기술은 디지털 정보 지식, 더 나아가 디지털 매체에 관련한 프로그램을 개발할 수 있는 능력이 된다. 컴퓨터식 사고가 뒷받침되지 않는 컴퓨터 관련 기능·기술은 정보 소비능력 밖에는 되지 못하는 것이다[8].

이러한 이유로 컴퓨터식 사고, 디지털 논리를 익히기 위해서는 알고리즘 및 프로그래밍 학습이 필요하다는 요구가 최근 늘고 있다[2][7]. 또한, 컴퓨터 프로그래밍이 문제 해결의 과정에서 분석력, 논리력, 창의력을 기르는데 우수한 효과가 있다는 연구 결과가 많이 보고되고 있다[1][3][6][18].

컴퓨터 프로그램은 컴퓨터 언어로 표현된 문제 해결을 위한 수리 논리적 절차이며 컴퓨터 알고리즘은 문제해결을 위한 수리 논리적 방법을 말한다. 프로그램은 알고리즘을 컴퓨터 언어로 표현한 것이므로 문제해결의 핵심은 수리 논리적 해결 방법의 고안에 있다고 할 수 있다. 따라서 프로그램에 의한 문제해결의 핵심은 수리 논리적 해결 방법의 고안에 있다고 할 수 있다. 따라서 프로그램에 의한 문제해결에는 알고리즘의 고안이 가장 중요하다고 볼 수 있다[9].

즉, 많은 교육 전문가들은 알고리즘 교육은 컴퓨터 과학교육의 핵심임을 주장하며 초등 컴퓨터교육에서도 프로그래밍 학습의 중요성이 강조되며 알고리즘의 교육의 필요성을 역설하고 있다[5][12][14][17].

2. 이론적 배경

앞서 말했듯이, 이러한 교육의 요구는 많으나 실

제 우리나라에서의 초등학생을 위한 컴퓨터식 사고 능력 개발을 위한 교육과정은 정보 영재 교육에서만 찾아볼 수 있을 정도이다. 또한 정보 영재 교육은 주로 Basic, C, C++, Java 등의 프로그래밍 언어를 가르치는 이벤트성 교육이나 웹페이지 작성 등의 툴 위주의 수업으로 이루어지고 있다[20].

즉, 알고리즘과 프로그래밍 교육에 대한 필요성은 인지하면서도 막상 무엇을 어떤 순서로 가르쳐야 할지에 대한 매뉴얼이라든지, 심도 있는 연구가 부족하다는 의미이다.

또한 이에 대한 학습을 효과적으로 이끌어 나가기 위한 방법을 고찰할 필요가 있다. 본 연구에서는 알고리즘 애니메이션을 기법을 적용하여 순차적인 그림 제시로 이해도를 최대한 높일 수 있도록 하였다. 이러한 학습은 언플러그드(Unplugged) 학습 방법의 한 종류라고도 할 수 있겠다. 이러한 알고리즘 학습 이후에 실제 프로그래밍을 학습할 수 있도록 교재의 순서를 구성하였다.

2.1 개념적 알고리즘

문교식(2007)은 개념적 알고리즘을 알고리즘 학습에서 중심으로 다루어져야 할 부분이라고 제시하였다.

2.1.1 개념적 알고리즘의 정의

첫째, 개념적 알고리즘은 문제해결의 집단적, 포괄적 설계에 해당한다.

둘째, 동일한 문제에 대해 동일한 개념적 알고리즘을 사용하더라도 다수의 서로 다른 특성을 갖는 개별적 알고리즘들이 가능하다. 여기서 개별적 알고리즘이란, 특정한 문제를 해결하기 위한 구체적이고 문제 종속적인 알고리즘을 말한다. 예를 들면, 정렬이라는 특정한 문제에 대한 구체적이고 문제 종속적인 알고리즘들의 예를 들면 병합 정렬, 퀵 정렬이 이에 해당한다.

셋째, 동일한 문제에 대해 서로 다른 개념적 알고리즘을 사용할 수 있다. 정렬의 예를 들면, 선택 정렬은 탐욕(Greedy) 알고리즘이지만 병합 정렬은 분

할 정복 알고리즘이다.

넷째, 개별적 알고리즘은 필요하다면 하나 이상의 개념적 알고리즘을 충돌 없이 상호 보완적으로 사용할 수 있다.

2.1.2 개념적 알고리즘의 유형과 학습 순서

이전의 교재들과 본 논문에서 개발한 교재를 비교할 때 차이점은 학습 내용에 대한 선별과 순서의 위계적 정렬에 있다. 또한 단순히 C 언어 학습을 위한 소스 코딩이 목적이 아니라 알고리즘을 이해하고 도식화, 추상화하여 코드로 바꾸기까지의 단계를 거쳐 가는 과정에 대한 학습이 구체적으로 제시되었다.

다음은 한국정보올림피아드에서 소개하는 알고리즘과 초등부 경시대회 기출문제를 통해 자주 활용되는 개념적 알고리즘의 유형이다[9][22][13][25].

이 알고리즘들을 상호간의 위계성과 학습 난이도, 활용도를 고려하여 교재 개발 시에도 다음의 순서대로 알고리즘들을 제시한다.

- **반복** : 순환, 루프라고 부르기도 하며 어떤 특정한 조건이 성립할 때까지 반복되어 실행하게 된다. 이 개념이 사용되지 않는 알고리즘은 거의 없을 정도로 가장 기초적인 개념이다(N개의 수 합산하기, N개의 수에서 최대값 구하기 등).

- **탐욕** : 욕심쟁이 알고리즘이라고도 하며 여러 선택에 있어서 앞으로의 선택을 고려하지 않고 각각의 선택마다 가장 최선의 선택을 하여 최적의 해를 찾기를 기대하는 알고리즘이다(선택정렬, 최소비용 신장 나무, 최단 경로 문제 등).

- **재귀** : 임의의 함수가 자신을 다시 호출하는 것을 의미하는데, 재귀호출을 사용하는 함수는 반드시 종료되는 조건이 설정되어야 한다. 재귀의 개념이 알고리즘에서 중요한 이유는 논리적 표현 자체가 재귀를 사용하는 경우가 흔하며 특정 문제 해결에 있어서 쉽고 자연스러운 경우가 많다(피보나치수열 등).

- **분할 정복** : 크기가 n인 문제를 좀 더 작은 크기의 같은 문제들로 쪼갠 다음 각각의 쪼개진 문제를 풀어서 얻은 답을 가지고 원래 문제의 답을 구하는 방법이다(병합 정렬, 퀵 정렬, 이진 탐색, 하노이 탑 문제 등).

- **동적계획법** : 다이나믹 프로그래밍, 동적 프로그래밍이라고도 하며 순서가 정해진 항목과 관련된 문제에서 한 번 계산한 부분해를 저장해 두고 필요할 때마다 그 결과를 참조함으로써 부분해의 중복 계산을 방지하는 방법이다(피보나치수열, 배낭 문제, 최단 거리 문제 등).

- **백트래킹** : 검색 공간의 모든 가능한 배치 방법에 대해 어떤 작업을 반복하기 위한 조직적인 방법이다(바둑, 장기 등의 게임 연산, 그래프 색칠 문제, N-여왕 문제 등).

2.2 알고리즘 교육 시각화

알고리즘의 내용을 학습자에게 효과적으로 전달하기 위한 방법으로는 매우 다양한 방법이 연구되어 왔다. 이 중 학생들의 수준에 맞으며 교재에 적용하기에 알맞은 방법으로는 그림과 도식, 순서도 등을 이용하는 알고리즘 애니메이션을 가장 적절한 예로 들 수 있다[11].

시각화된 정보는 문서 지향적인 정보 형태보다 많은 양의 정보를 효과적으로 전달해 줄 수 있다[23]. 프로그램 시각화(Program Visualization)는 프로그램의 동작, 제어의 흐름, 그리고 자료의 상태변화 등을 시각적 형태로 표현하는 것을 말한다. 이것은 프로그램의 특정 상태를 정적으로 나타내는 것뿐 아니라, 전체적인 혹은 일정 범위의 수행 과정을 동적으로 나타내는 것도 포함한다. 특별히 후자의 기능을 제공하는 프로그램 시각화를 알고리즘 애니메이션(Algorithm Animation) 혹은 Animated Graphical View라고 한다[24].

프로그램 시각화는 학습자에게 문서로 작성된 프로그램을 보는 것보다 훨씬 더 많은 양의 정보와

더욱 효과적인 이해 체계를 제공해주며 프로그램 수행을 연구하는데 있어서 강력한 접근법을 제공해 준다[23].

2.3 언플러그드 교육

언플러그드(Unplugged) 컴퓨팅 학습은 뉴질랜드의 팀 벨(Tim Bell) 교수가 언플러그드 프로젝트에서 컴퓨터 없이 컴퓨터 과학을 학습할 수 있는 활동들을 개발하여 제안한 교수학습 방법이다[26]. 이러한 교육방법은 컴퓨터 과학으로부터 나온 아이디어들이 학습자를 위한 다양한 문제들과 퍼즐을 이용하여 사용된 활동들이다. 경쟁적인 활동과 협동적인 활동을 이용하여 학습자들이 문제에 대한 호기심과 몰입도를 높일 수 있도록 구성되었다[4].

많은 국내 교육 전문가들도 이 교육방법을 이용하여 학습자가 컴퓨터 과학의 내용을 학습할 수 있도록 하여 논리적 사고력 향상 및 학습태도에 긍정적인 효과를 입증하였다[21][16][10].

2.4 IOI Syllabus

국제정보올림피아드(IOI: International Olympiad in Informatics) 산하의 국제과학위원회(International Scientific Committee)에서는 IOI 경시가 대학교육 이전인 중·고등학생들을 대상으로 하는 경시인 만큼 참가학생들이 선행학습에 의한 경쟁 보다는 각자의 지적 능력과 창의력으로 경쟁할 수 있도록 하기 위하여 2006년도 멕시코 Merida에서 개최된 제 18회 IOI에서 경시문제 출제에 대한 수준과 그에 관련된 정보과학의 교육을 위한 교과와 내용을 제시하기 위하여 IOI Syllabus안을 발표하여 앞으로의 IOI 경시 문제 출제에 반영하고 있다[27]. <표 1>은 앞서 설명한 IOI Syllabus에 대한 내용을 요약한 것이다.

<표 1> IOI Syllabus

1. 수 학
1) 수와 기하 정수·분수의 연산, 점·선분·다각형 등

2) 이산구조

함수: 관계와 집합

기본논리학: 명제·술어논리, 진리표 등

증명기법: 재귀호출의 정의, 연역·귀납법 등

순열과 조합: 순열, 조합, 경우의 수 세기

그래프와 트리: 기본적 성질, 탐색 기법 등

2. 컴퓨터 과학

1) 프로그래밍의 기본

기초 프로그래밍: 언어의 구조와 기능에 대한 지식

알고리즘과 문제해결: 전략과 구현

기초 자료구조: 기본 자료구조에 대한 지식

재귀호출: 개념과 구현, 백트래킹의 구현

2) 알고리즘과 복잡도

기본 알고리즘: 사전·사후 조건, 복잡도 계산

설계 전략: 탐욕·분할정복·동적계획법

계산 알고리즘: 정수에 관한 수치계산

기하 알고리즘: 선분, 단순 다각형 관련 알고리즘

3) 소프트웨어 공학

작은 규모의 단독 개발 프로젝트의 기본 기법

4) 컴퓨터 사용 능력

컴퓨터의 기본 구조와 작동에 대한 이해

2.5 한국정보올림피아드 경시대회

<표 2> 한국정보올림피아드 소개

1. 개최일시	매년 7월 초 ~ 7월 중순
2. 출제방향	수학적 지식 및 논리적 사고능력을 필요로 하는 알고리즘과 그 구현을 경시하는 문제출제
3. 경시내용	각 분야별 수준에 맞는 문제해결 및 프로그램 작성 능력 평가
4. 사용언어	비주얼베이직, 비주얼 C++

한국정보올림피아드(KOI: Korea Olympiad in

Informatics)는 행정안전부에서 주최하고 한국정보문화진흥원에서 주관하는 대회이다. 국내 최고의 IT 영재들이 참가하여 실력을 겨루는 대회로써 국내 최고의 권위를 가진 대회이며 지역(시도)에서 선발된 학생들이 주어진 문제해결 능력을 겨루는 경시대회와 학생이 스스로 개발한 소프트웨어의 작품성을 평가하는 공모대회로 진행이 된다. 경시부분 우수 입상자에게는 국제정보올림피아드 참가 후보자격을 부여하고 있다[29].

3. 교재개발

본 교재의 학습 목표는 알고리즘의 학습이며 프로그래밍은 학습된 내용을 실제 구현하는 단계이므로 알고리즘 학습 후에 제시된다.

3.1 교재의 순서

교재의 순서는 <표 3>과 같다.

<표 3> 교재의 목차

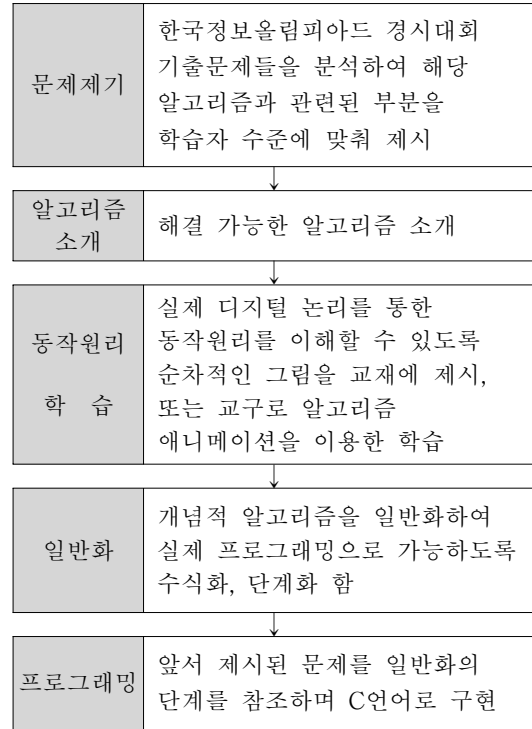
알고리즘으로 컴퓨터 과학 배워요.	
1.	알고리즘이란?
2.	배열, 스택과 큐
3.	욕심쟁이 알고리즘
4.	재귀
5.	분할 정복
6.	동적계획법
7.	백트래킹

3.2 대상 학습자

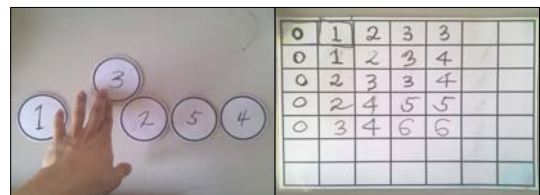
본 교재에서의 프로그래밍 언어는 C를 사용하였다. 본 교재의 학습자는 기초적인 C 프로그래밍의 학습이 이루어진 대상자로 한정한다. 즉 C 언어의 각 예약어의 기능을 알고 있으며 기본적인 입출력 코딩이 가능한 학습자를 돕는 교재이다.

3.3 교재 내용 구성의 단계

본 교재에서는 앞서 제시된 다양한 이론적 배경을 근거로 다음과 같은 순서로 학습내용을 제시한다.



(그림 1) 교재 내용 구성의 단계



(그림 2) 주로 사용되는 교구(알판, 2차원배열판)

3.4 교재 내용의 실제

다음은 교재 내용의 실제 부분으로 6단원 동적계획법의 내용의 일부분을 소개한다.

6 동적계획법

문제제기 치즈를 가장 많이 먹어보자.

[그림6_2]는 5×5 크기의 방에 치즈(Ⓢ)가 있는 곳을 표현했다. 실험용 쥐는 시작점에서 끝점까지 치즈를 찾아 먹으며 이동하게 되는데 오른쪽 또는 아래로만 이동할 수 있다. 실험용 쥐가 가장 많은 치즈를 먹고 끝점에 도달할 때의 치즈의 수를 구해보자.

시작	Ⓢ	Ⓢ	Ⓢ	
				Ⓢ
	Ⓢ	Ⓢ		
		Ⓢ	Ⓢ	
	Ⓢ		Ⓢ	끝

[그림 6_2] 치즈방

동적계획법 동적계획법을 알아보자.

동적계획법이란 큰 문제를 작은 문제로 쪼개어 해결한다는 점에서 분할정복법과 비슷한 점이 많다. 문제를 해결할 때 분할정복법은 제일 큰 문제에서 시작하여 작은 문제로 해결방법을 선택하고 동적계획법은 제일 작은 문제를 먼저 해결하고 이를 활용하여 점점 큰 문제를 해결하는 방법에서 차이점을 찾을 수 있겠다.

시작	Ⓢ	Ⓢ	Ⓢ	
				Ⓢ
	Ⓢ	Ⓢ		
		Ⓢ	Ⓢ	
	Ⓢ		Ⓢ	끝

[그림6_3] 욕심쟁이 알고리즘으로 풀이 한 가지 문제에 대한 해결 전략은 다양할 수 있다. 위 문제를 욕심쟁이 알고리즘(아래쪽 우선 이동)으로 풀어보면 [그림6_3]과 같이 총 4개의 치즈를 먹을 수 있다. 과연 이것이 이 문제의 해답일까?

동작원리 컴퓨터는 어떻게 해답을 찾을까?

이 문제를 직접 공책에서 풀어보면 어렵지 않게 해답을 찾을 수 있을 것이다.

시작	Ⓢ	Ⓢ	Ⓢ	
				Ⓢ
	Ⓢ	Ⓢ		
		Ⓢ	Ⓢ	
	Ⓢ		Ⓢ	끝

[그림6_4] 치즈방 문제의 해답

해답은 [그림6_4]와 같이 2가지 경우로 실험용 쥐는 총 6개의 치즈를 먹을 수 있다. 5×5 크기의 방은 사람의 힘으로 한눈에 계산이 가능하지만 더 큰 크기의 방은 컴퓨터의 힘을 빌려야 할 것이다. 컴퓨터는 과연 이 문제를 어떻게 해결해 나갈 수 있을까? 문제의 핵심은 이것이다. [그림6_2]에서 치즈방[X][Y] 지점에 도달하기 위해서는 치즈방[X-1][Y] 또는 치즈방[X][Y-1]을 반드시 지나고 와야 하기 때문에 치즈방[X-1][Y]와 치즈방[X][Y-1]에서의 치즈를 가장 많이 먹고 오는 값을 계산하면 되는 것이다.

		1	2	3	4	5
1	시작	Ⓢ	Ⓢ	Ⓢ		
2						Ⓢ
3			Ⓢ	Ⓢ		
4				Ⓢ	Ⓢ	
5			Ⓢ		Ⓢ	끝

[그림6_5] 치즈방[X][Y]에서 X가 3, Y가 2일 경우 분할정복법에서 큰 문제를 작게 나누었던 기법 그대로 이 문제도 그런 과정이 필요하다.

		1	2	3	4	5
1	시작	Ⓢ	Ⓢ	Ⓢ		
2						Ⓢ
3			Ⓢ	Ⓢ		
4				Ⓢ	Ⓢ	
5			Ⓢ		Ⓢ	끝

[그림6_6] 치즈방[3][1], 치즈방[2][2]에서의 도착 경로 찾기

예를 들자면 [그림6_6]에서 시작점을 치즈방 [1][1]이라고 할 때 치즈방[3][2]에 도달하기 위해서는 실험용 생쥐는 반드시 치즈방[3][1] 또는 치즈방[2][2]를 지나야 한다.

일반화 치즈방 문제를 동적계획법으로!

이러한 과정을 염두에 두고 아래와 같은 식을 만들어낼 수 있을 것이다.

$$\text{치즈방}[X][Y] = \text{최대값} \{ \text{치즈방}[X-1][Y], \text{치즈방}[X][Y-1] \}$$

	1	2	3	4	5
1	0	1	1	1	0
2	0	0	0	0	1
3	0	1	1	0	0
4	0	0	1	1	0
5	0	1	0	1	0

	1	2	3	4	5
1	0	1	2	3	3
2	0	1	2	3	4
3	0	2	3	3	4
4	0	2	4	5	5
5	0	3	4	6	6

[그림6_7] 2차배열 [그림6_8] 동적계획법
 실제 프로그래밍 하기에 앞서 [그림6_2]의 치즈방을 5×5인 2차원 배열로 구성하고 치즈가 있는 곳에 숫자 1을, 없는 곳에는 0으로 가득 채운다[그림6_7]. 또한 위에서 이끌어낸 식을 이용하여 [그림6_7]을 [그림6_8]로 변형시킬 수 있을 것이다.
 이 때 [그림6_8]의 치즈방[5][5]에 쓰여진 숫자가 실험용 생쥐가 먹을 수 있는 치즈의 최대값이 된다.

프로그래밍 C언어로 구현하기

위의 문제 해결방법을 구현하면 아래와 같다. INPUT.TXT 파일의 값이 [소스6_1]과 같다고 할 때 소스는 [소스6_2]와 같다.

[소스6_1] INPUT.TXT

5
0 1 1 1 0
0 0 0 0 1
0 1 1 0 0
0 0 1 1 0
0 1 0 1 0

[소스6_2] Cheese.C

```

1 #include <stdio.h>
2 #define X 101
3 #define Y 101
4 int cheese[X][Y]={0,};
5 int MAX_way(int x, int y){
6     if(cheese[x-1][y]>cheese[x][y-1])
7         return cheese[x-1][y];
8     else return cheese[x][y-1];
9 }
10 int main(){
11     int i,j,n;
12     FILE *in=fopen("INPUT.TXT","r");
13     FILE *out=fopen("OUTPUT.TXT","w");
14     fscanf(in,"%d",&n);
15     for(i=1;i<=n;i++){
16         for(j=1;j<=n;j++){
17             fscanf(in,"%d",&cheese[i][j]);
18         }
19     }
20     for(i=1;i<=n;i++){
21         for(j=1;j<=n;j++){
22             cheese[i][j]+=MAX_way(i,j);
23         }
24     }
25     fprintf(out,"%d ",cheese[n][n]);
26     fclose(in);
27     fclose(out);
28     return 0;
29 }
    
```

- 4 최대 cheese[101][101]에 치즈방 2차 배열을 입력받는다.
- 5 ~ 9 MAX_way라는 함수는 현재 치즈방의 위치 x, y를 받아 치즈방[x-1][y]의 값과 치즈방[x][y-1]의 값 중 최대값을 반환한다.
- 20 ~ 24 MAX_way함수에서 반환된 값을 현재 치즈방 [i][j]에 있는 수와 합하여 저장한다.
- 25 치즈방[끝][끝]의 값을 OUTPUT.TXT 파일에 저장한다.

4. 적용 및 연구결과의 해석

본 연구에 의한 학습이 어느 정도의 효과가 있는지를 검증해보고자 다음과 같이 교육 및 검사를 진행하였다.

4.1 연구대상 및 연구방법

본 연구는 <표 4>와 같이 J대학교 주관의 초등 정보 영재 A반 9명을 실험집단 대상으로 정하고 2009년 4월 4일부터 6월 20일까지 약 3개월 보름동안 7회의 집체교육으로 학습이 이루어졌다. 1회의 교육은 총 3시간이며 본 연구에서 개발한 교재로 교수·학습이 이루어졌다.

<표 4> 연구대상

구 분	아 동 수				계
	남		여		
학 년	5	6	5	6	
학생수	2	3	·	4	9

4.2 연구가설

본 연구에서는 다음과 같이 가설을 설정하였다.

영가설: 본 교재의 학습 전후 학생들의 한국정보 올림피아드 문제 해결능력에는 차이가 없다.
대립가설: 본 교재의 학습 전후 학생들의 한국정보 올림피아드 문제 해결능력에는 차이가 있다.

4.3 검사도구 및 연구설계

검사도구는 1996년 한국정보올림피아드 초등부 경시대회 전국본선에서 기출된 3문제로 하였고 사전·사후 검사 모두 동형인 검사지를 사용하였다.

한국정보올림피아드 경시대회에서의 문제들은 모두 답이 정해져 있지만 답에 근접한 값이 정답이 되는 경우도 있을 수 있으며 답을 도출해 내는 방법이 다양할 수 있다.

또한 중요한 것은 문제에서 제시되는 조건에 맞

춰 수험자는 프로그램을 작성해야 한다는 것이다. 주로 문제들은 실행시간의 제한을 두게 되는데 이 시간 안에 출제자가 제시한 답을 구한다면 점수를 부여받을 수 있고, 반대의 경우에는 점수가 아예 없다. 알고리즘의 효율성에 따라 제한된 시간 내에 결과를 낼 수도 있고, 제한 시간에 답을 못 낼 수도 있게 된다. 문제에서 1초안에 결과가 나올 것이라고 했는데 프로그램의 효율성이 낮은 관계로 5초에 답이 나왔다고 하면 점수를 얻을 수가 없다[28].

본 연구에서는 위와 같은 기준으로는 학습 효과에 대한 결과를 도출하기 힘들기 때문에 <표 5>와 같이 채점기준표를 재구성하였다.

<표 5> 채점기준표

구분	내 용	배 점
1	문제1, 문제2	각 30점
2	문제3	40점
계	총점	100점

구분	감 점 내 용	감 점
1	모든 조건을 만족하며 모든 테스트 데이터의 결과값이 올바르다.	0점
2	결과값은 올바르나 시간이 초과된다.	-10점
3	일부의 결과값은 올바르지 못하다.	-10점
4	일부의 결과값은 올바르지 못하며 때로는 시간이 초과된다.	-20점
5	알고리즘 설계는 올바르나 결과값이 모두 올바르지 못하다. (프로그래밍 과정상의 실수)	-25점

본 연구에서의 사전·사후 평가에 대한 해석은 단일표본 t검정을 통해 대상 집단의 유의미한 평균의 차이가 있는지를 알아보고자 한다.

본 연구에서 개발한 교재를 투입하여 교육하기 시작하기 전인 2009년 4월 4일에 검사지를 통해 사전평가가 이루어졌고 교육을 끝낸 2009년 6월 20일에 사후평가가 이루어졌다. 채점기준표에 의해 검사 실시 결과로 얻어진 점수를 종속변인으로 설정하였다. 연구에서 α 레벨은 $p < .05$ 로 검증하였다.

4.4 연구결과 및 해석

본 연구에서 개발한 교재의 적용 결과 다음과 같은 결과를 도출해낼 수 있었다.

<표 6> 대응표본 상관계수

		N	상관계수	유의확률
대응1	사전검사 & 사후검사	9	.965	.000

<표 7> 대응표본 검정

사전검사		평균	사후검사		평균
		4.4444			26.6666
		대응차			
		평균	표준편차	평균의 표준오차	
대응1	사전검사-사후검사	-8.333333	10.00000	3.33333	
대응차		t	자유도	유의 확률 (양쪽)	
차이의 95% 신뢰구간					
하 한	상 한				
-16.02001	-.64665	-2.500	8	.037	

두 변수의 대응표본 상관계수는 0.965로 강한 상관관계를 보이고 있으며 유의 확률(양쪽) p=0.037이므로 유의수준 0.05에서 사전검사와 사후검사의 평균차이는 유의한 것으로 나타나 영가설은 기각된다.

즉 본 연구에서 개발한 교재로 효과적인 학습이 이루어졌음을 알 수 있었다.

5. 결론 및 제언

알고리즘과 프로그래밍 학습이 컴퓨터 과학의 교육의 핵심 요소임은 누구나 인정하고 있지만 무엇을 어떻게 어떤 순서대로 가르쳐야 할지는 아직도 연구가 충분치 못한 실정이다.

본 연구에서는 개념적 알고리즘을 학습을 위해 프로그램의 시각화(알고리즘 애니메이션)의 교수방법과 개발된 교재를 이용하여 최대한 학습자가 어려운 알고리즘의 내용을 쉽게 이해할 수 있도록 순차적인 단계를 통해 학습을 유도하였다.

이 교재의 문제들은 개념적 알고리즘별로 한국정보올림피아드 경시대회 전국본선 기출문제들을 변형하여 제시하였다. 이러한 문제들은 단순한 프로그래밍 기법의 훈련만으로는 해결할 수 없는 논리력과 창의력이 필요한 종류의 것이다[19].

본 교재의 모든 내용이 초등학교 전학년을 대상으로 가르칠 수 있는 수준이 아님은 명백하다. 하지만 사교육에 의존한 정보올림피아드 대회의 전면에 대한 문제점을 생각해 볼 때 더 이상 알고리즘 교육은 영재교육이 아니며, 어려운 분야의 것도 아님을 하루 빨리 인지할 필요가 있다. 즉, 이러한 대회가 갖는 취지를 이해하고 학생들을 교육시켜야 하는 역할도 공교육에서 충분히 할 수 있으며 해야 할 것이다. 또한 초등학교 모든 학생들이 배울 수 있는 알고리즘 교육과 교재의 개발도 병행되어야 할 필요가 있다고 본다.

향후 연구에서는 본 연구가 기반이 되어 더 좋은 교재가 개발되어 공교육에서의 컴퓨터 과학 교육의 역할을 다하기를 바라며 각 지역교육청 또는 관련 연구회와 공동연구를 통해 교재가 개발되기를 기대한다.

참고문헌

- [1] 강성원, 이애정, 이재호 (2003). 초등 정보과학 영재용 프로그래밍 교육, 한국정보교육학회 논문지 7-3, 363-371.
- [2] 강신천 (2005). 초중등 컴퓨터 교육과정과 대학 컴퓨터 교육과정 연계 연구, IT 인력 양성과 컴퓨터 교육 워크샵, 한국교육과정평가원.
- [3] 김미량 (2002). 컴퓨터 프로그래밍 교육에 적용 가능한 효과적 교수방법의 탐색적 대안, 한국컴퓨터교육학회 논문지 5-3, 1-9.
- [4] 김영기, 한선관 (2007). 정보교육방법의 실제, 한국학술정보출판사.
- [5] 김정우, 이미화 (2006). 알고리즘 교육을 위한 시뮬레이션 설계 방안 연구, 한국정보교육학회 학술발표논문집 11-1, 243-248.
- [6] 김태완, 유인환 (2004). LEGO Dacta를 이용한 프로그래밍 학습모형 개발을 위한 이론적 고찰, 한국

정보교육학회 학술발표논문집 9-2, 216-225.

[7] 김현철 (2005). 외국의 컴퓨터 교육과정 사례를 통한 지식정보 교육과정 분석, IT 인력 양성과 컴퓨터 교육 워크샵, 한국교육과정평가원.

[8] 목영해 (2001). 디지털 문화와 교육, 문음사.

[9] 문교식 (2007). 개념적 알고리즘에 기반한 컴퓨터 알고리즘 교육의 방향, 정보교육학회논문지 11-1, 29-38.

[10] 박윤성, 한병래 (2009). 초등학교 컴퓨터교육에서 언플러그드 학습 방법을 활용한 정보표현 영역 교수 · 학습에 관한 연구, 정보교육학회논문지 13-4, 479-487.

[11] 배영권, 문교식 (2008). 사칙연산을 이용한 알고리즘 원리 학습 방안, 정보교육학회논문지 12-3, 303-312.

[12] 이길복, 전우천 (2004). 초등학교 정보영재를 위한 알고리즘 학습 지원 시스템 연구, 한국정보교육학회 학술발표논문집 9-1, 296-303.

[13] 이영호 (2004). 월드와 함께하는 정보올림피아드 알고리즘 이론편, yeswoa닷컴.

[14] 이주희, 김갑수 (2006). 구체적 조작기의 초등학생을 위한 정렬 알고리즘 교수 · 학습에 관한 연구, 한국정보교육학회 학술발표논문집 11-2, 95-100.

[15] 이현수 (2002). 제14회 국제정보올림피아드. 정보과학회지 20-3, 66-70.

[16] 임민영, 정상목, 한병래 (2007). 초등학생을 위한 컴퓨터 과학의 언플러그드 학습 방법 연구, 정보교육학회논문지 11-4, 497-504.

[17] 임화경, 전승순 (2006). 초등 수문제를 이용한 컴퓨터 알고리즘 개념에 대한 교수방법, 한국컴퓨터교육학회 논문지 9-3, 109-119.

[18] 장승익, 권보섭 (2003). 프로그래밍 학습과 교과학습간의 상관관계 연구, 한국컴퓨터교육학회 논문지 9-3, 109-119.

[19] 장지현 (2007). 정보올림피아드와 정보과학 교육. 정보과학회지 25-7, 61-66.

[20] 하성욱 (2002). 쉽게 배우는 실전 알고리즘 & 정보올림피아드 도전하기, 영진닷컴.

[21] 한선관, 김경신 (2007). 초등학생을 위한 컴퓨터 과학의 언플러그드 학습 방법 연구, 정보교육학회 논문지 11-4, 497-504.

[22] Gary Haggard, John Schlipf, Sue Whitesides

(2006). Discrete Mathematics for Computer Science, Cengage Learning.

[23] John T, Stasko (1990). Tango: A Framework and System for Algorithm Animation, IEEE Computer 23-9, 27-39.

[24] Marc H. Brown (1985). Techniques for Algorithm Animation, IEEE Computer 2-1, 28-39.

[25] Steven S.Skiena, Miguel A.Revilla (2004). Programming Challenges, Springer.

[26] Tim Bell, Lan H. Witten, Mike Fellows (2006). 놀이로 배우는 컴퓨터 과학, 홍릉과학출판사

[27] Tom Verhoeff, Gyula Horvath, Krzysztof Diks and Gordon Cormack (2006). A proposal for and IOI Syllabus, Teaching Mathematics and Computer Science 4-1, 193-216.

[28] 한국정보화진흥원. 자주묻는질문 보기. [http:// www.nia.or.kr/koi/F03000000000/F03020000000V.asp?BoardIDX=20](http://www.nia.or.kr/koi/F03000000000/F03020000000V.asp?BoardIDX=20)

[29] 한국정보화진흥원. KOI 소개. <http://www.nia.or.kr/KOI/F01000000000/F01020000000.asp>

저 자 소 개

김 병 수



2002 제주교육대학교
컴퓨터교육과(교육학박사)
2010~현재 제주대학교
컴퓨터교육전공 박사과정
관심분야: STEAM 교육, 알고리즘
교육, Computational Thinking
e-mail: pigpotato79@naver.com

김 종 훈



1998 홍익대학교
전자계산학과(이학박사)
1999~현재 제주대학교
초등컴퓨터교육전공 교수
관심분야: 컴퓨터교육
e-mail: jkim0858@jejunu.ac.kr