

# 개념적 알고리즘에 기반 한 컴퓨터 알고리즘 교육의 방향

문교식

대구교육대학교 컴퓨터교육과

## 요 약

최근 컴퓨터 교육에서 프로그래밍에 대한 관심이 높아지고 있다. 학습에서 논리력과 창의력의 요구가 늘어나고 있는 추세에 힘입어 컴퓨터 교육 분야에서도 프로그래밍 학습을 통한 논리력과 창의력을 함양하려는 다양한 시도가 이루어지고 있다. 프로그램은 알고리즘을 컴퓨터 언어로 구현한 것이므로 문제해결의 핵심은 논리적 해결 방법의 고안 즉, 알고리즘의 고안에 있다. 알고리즘의 중요성을 인식한다면 이제 알고리즘 교육에 대한 방향 정립을 위한 체계적 연구가 필요한 시점에 와 있다고 볼 수 있다. 지금까지 주로 정렬, 탐색과 같은 특정 문제를 중심으로 한 문제 종속적이고 개별적인 알고리즘 학습을 강조해 왔다. 본 논문에서는 문제 독립적, 집단적 성질을 고려하는 문제 해결의 개념적 유형이라는 관점에서 개념적 알고리즘 (conceptual algorithm)을 정의한다. 개념적 알고리즘의 유형적 특성을 바탕으로 개별적, 문제 종속적인 기존의 알고리즘 교육과 구분되는 체계적, 문제 독립적, 알고리즘적인 문제 해결력을 지향하는 알고리즘 교육의 새로운 방향을 제안한다.

## On the Direction of the Computer Algorithm Education Based on Conceptual Algorithms

Gyo Sik Moon

Dept. of Computer Education, Daegu National University of Education

### Abstract

Computer programming in computer education draws much attention recently. Encouraged by the increased tendency of acquiring logical ability and creativity through learning, various attempts have been made to develop them through learning computer programming in the area of computer education as well. The fact that a computer program is the representation of a computer algorithm expressed in a computer language makes us realize that the devise of a logical method for a solution - i.e., the design of an algorithm - is the key to the solution of a problem. Recognizing the importance of computer algorithm would lead us to such a point that systematic investigations for directional establishment for algorithm education are necessary. We observe that researches on teaching computer algorithm have concentrated mostly on specific problems such as sorting and searching, which can be characterized as problem-dependent and individual. In this paper, the idea of conceptual algorithm is stated from the standpoint of conceptual types of problem-solving methods which are considered as problem-independent and collective. A novice approach to algorithm education based on the characteristics of types of conceptual algorithms is proposed for the purpose of developing systematic, problem-independent, algorithmic problem-solving capabilities of learners, which is widely different from the current methods of individual and problem-dependent algorithm education.

## 1. 서론

컴퓨터 교육에서 전산학의 핵심 분야인 프로그래밍과 알고리즘 교육에 대한 요구가 최근 늘고 있다 [2,7]. 컴퓨터 프로그래밍 학습을 통하여 컴퓨터의 기본 원리를 배우고 문제해결의 과정에서 논리력과 창의력을 개발하려는 다양한 교육적 노력이 시도되고 있다. 그 중 컴퓨터 프로그래밍은 문제 해결의 과정에서 분석력, 논리력, 창의력을 기르는데 우수한 효과가 있다는 연구보고가 많다[1,4,6,16]. 컴퓨터 프로그램은 컴퓨터 언어로 표현된 문제해결을 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 문제해결에는 알고리즘의 고안이 가장 중요하다고 볼 수 있다. [2]에서 강신천(2005)은 “컴퓨터 과학 교육은 컴퓨터와 알고리즘적인 프로세스를 배우는 학문으로 이들의 원리와 하드웨어, 소프트웨어 설계, 이들의 응용 및 이들이 미치는 사회적인 영향 등을 연구하는 학문 분야라고 할 수 있다”라고 알고리즘 분야가 컴퓨터 과학 교육의 핵심임을 주장하였으며 다른 많은 교육전문가들도 알고리즘 교육의 중요성을 인식하고 있다. 최근 초등 컴퓨터 교육에서도 프로그래밍 학습의 중요성이 부각됨에 따라 알고리즘 교육의 필요성이 주목을 받고 있다[5,11,14,15,17,18].

이와 같이 알고리즘을 왜 가르쳐야 하는지에 대한 논의는 많이 있어 왔다. 이제는 무슨 내용을 가르칠 것인가에 대한 체계를 세워야 할 때라고 본다. 그러나 현재 논의되고 있는 알고리즘 학습 내용은 거의 전부가 정렬과 탐색 등 전산학에서 기초가 되는 주제를 중심으로 논의 되고 있다. 정렬 문제나 탐색 문제가 전산학에서 기초적이고 핵심적인 내용임에 틀림없지만 수많은 전산학의 문제들 중의 특정한 문제임을 부인 할 수 없다. 특정한 문제를 해결하는 과정에서 알고리즘의 학습이 될 수 있음을 인정하지만 개별적 문제에 대한 알고리즘 학습 이전에 알고리즘의 원리에 입각한 보다 일반적인 개념을 체계적으로 학습하는 것이 더 효과적이라 생각된다. 과연 알고리

즘의 개념에 입각한 전체적이고 체계적 학습이라는 것이 존재하는지, 있다면 어떠한 순서로 학습해야 하는지, 왜 그러한지, 등에 대한 논의를 본 논문에서 다루고자 한다.

논문의 구성은 제2장 관련 연구에서 컴퓨터 알고리즘의 특성 및 교육의 중요성을 살펴보고 알고리즘 교육의 내용과 관련된 기존의 연구 결과를 검토하여 현재의 연구 방향에 대한 전반적 이해를 돕고자 한다. 제3장에서는 개념적 알고리즘과 그 유형을 식별할 목적으로 개별적 알고리즘과 개념적 알고리즘을 정의하고, 개념적 알고리즘의 유형과 특징을 제시한다. 제4장에서는 개념적 알고리즘의 학습을 주제로 하여 개념적 알고리즘의 유형별 학습 이유와 학습 목표, 각 유형에 적합한 학습 내용을 제안하고 학습의 순서와 단계를 제시한다. 끝으로 제5장에서 결론을 맺는다.

## 2. 관련 연구

### 2.1 컴퓨터 알고리즘의 특성과 알고리즘 교육의 중요성

컴퓨터 알고리즘의 전통적인 정의는 [21,22,23] 등에서 찾아 볼 수 있다. 컴퓨터 알고리즘이란 문제해결을 위한 정확한 방법으로 컴퓨터에 의해 사용 가능해야 한다. 컴퓨터 프로그램이란 문제해결을 위해 고안한 알고리즘의 구현 도구라는 사실로부터 알고리즘과 프로그램의 관계를 찾을 수 있다. 알고리즘은 수리 논리에 기반 한 문제해결의 개념적 설계이므로 프로그래밍 단계에 선행하며 프로그램의 언어적 특성과는 독립적 성질을 갖는다. 알고리즘의 특성을 아래와 같이 정리할 수 있다.

(㉠) 효과성; 하나의 알고리즘은 동일 유형의 문제 군에 집단적 해결을 보장하는 일반적 특성을 갖는다.

(㉡) 정확성; 정확한 알고리즘은 모든 입력에 대하여 항상 올바른 답을 보장한다.

(㉢) 효율성; 알고리즘은 정확한 답을 보장할 뿐 아니라 문제해결의 최선의 방법을 지향한다.

(㉣) 전이성; 하나의 알고리즘은 다른 알고리즘으로의 변형이나 이전이 가능한 경우가 많다.

알고리즘 교육의 중요성에 대한 최근의 연구 결과를 정리하면 아래와 같다[14,18].

(㉠) 알고리즘의 이해와 개발의 과정을 통하여 논리력, 사고력, 창의력의 함양에 도움이 된다.

(㉡) 알고리즘은 프로그래밍과 컴퓨터에 대한 기본 개념이다.

(㉢) 알고리즘 교육은 컴퓨터 프로그래밍의 가장 핵심이 되는 과목이다.

(㉣) 컴퓨터를 사용하는 문제해결력을 기른다.

이와 같이 알고리즘 교육은 알고리즘이 컴퓨터과학의 핵심이라는 위치적 중요성 뿐 아니라 문제해결력의 신장과 논리적 사고력, 창의력의 함양이라는 관점에서 그 중요성을 찾을 수 있다.

## 2.2 알고리즘 교육과 관련된 연구 결과

알고리즘 교육의 중요성이 부각됨에 따라 기능 위주의 프로그래밍 교육에서 탈피하여 알고리즘 교육을 도입하려는 시도가 늘어나고 있다. 최근의 알고리즘 교육과 관련된 연구 결과를 간추리면 다음과 같다.

[3]; 자료 구조적 알고리즘을 사례 중심의 애니메이션으로 시각화하였다.

[5]; 초등학생을 위한 알고리즘 교육용 웹 기반 시뮬레이션의 모형을 제시하였으며 알고리즘의 교육내용으로 정렬과 탐색 알고리즘을 선정하였다. 선정의 이유로 자료처리의 가장 기본이 됨을 들었다.

[9]; 정보블록 정렬 알고리즘의 성능을 측정할 수 있는 시뮬레이터의 기능을 갖춘 시각화 학습시스템을 제시하였다.

[10]; 정보영재 교육 프로그램의 알고리즘 부분에서는 순서도와 자료구조의 여러 형태(스택, 큐, 리스트, 배열, 트리, 그래프), 정렬과 탐색, 이산수학 기초에 대한 교육으로 편성되어 있다.

[11]; 정렬 알고리즘을 학습자가 자기 주도적으로 학습할 수 있도록 하며 정렬 알고리즘의 개념형성을 위해 정렬 알고리즘의 종류별로 그 과정을 보여준다. 정렬 알고리즘을 알고리즘 학습 내용으로 선택한 이유로 프로그램 개발에서 사용도가 많고 중요한 위치를 차지하며 다양한 방법에 따라 성능의 차이를 비교할 수 있어 알고리즘에 대한 다양한 이해가 가능

하다고 했다. 학습 내용에 포함된 정렬 알고리즘의 종류는 버블 정렬, 선택 정렬, 삽입 정렬, 합병 정렬, 빠른 정렬, 힙 정렬이다.

[12]; 객체지향 설계를 통하여 객체 및 클래스와 상속 개념에 기반 한 교육과정을 구성하여 사고의 추상화와 확장성 및 재사용성을 높이는 방안으로 초등학생 자료구조 교육과정을 제시하였다.

[14]; 구체적 조작기 초등학생들의 인지수준에 적합한 교수학습 방안을 제시하였으며 알고리즘 학습 내용으로 정렬(선택, 삽입, 버블정렬)을 선정하였다. 정렬 알고리즘의 선정 이유로 프로그램에서 기초적인 도구라는 중요성과 함께 구체적 조작기 아동의 인지수준인 서열화, 유목화, 가역적 사고에 적합하다는 이유이었다.

[15]; 초등 4학년 수학문제를 활용하여 순차, 조건, 반복적인 특징을 발견하고 그 내용을 순서도로 표현하기 위한 학습내용을 갖는다.

[17]; 학생들의 프로그램을 시각화할 수 있는 탐색 알고리즘 교육을 위한 소프트웨어 컴포넌트를 개발하여 현장에 적용하였다. 탐색 알고리즘을 교육내용으로 선정한 이유는 알고리즘에서 기본이 되며 학생들의 학습을 돕기 위한 방법론의 개발이 필요하기 때문이라고 하였다.

[18]; 알고리즘 교육의 중요성을 강조하면서 초등 학교 컴퓨터 교육과정에 알고리즘을 포함하여야 함을 주장하였다. 그 교육내용은 중요성을 근거로 정렬과 검색 알고리즘을 선정하였으며 순서도 작성법을 아울러 교육내용에 포함하였다.

그 밖에도 알고리즘 학습에 대한 연구가 다수 있으며 대부분 정렬 알고리즘이나 탐색 알고리즘을 알고리즘 학습의 내용으로 채택하고 있음을 볼 수 있다. 그러한 선정의 이유는 정렬/탐색 알고리즘이 알고리즘 분야에서 차지하는 중요도에 주로 기인하고 있다.

## 3. 개념적 알고리즘과 그 유형

### 3.1 개별적 알고리즘과 개념적 알고리즘

특정한 문제를 해결하기 위한 구체적이고 문제 종속적(problem-dependent)인 알고리즘을 개별적 알고

리즘(individual algorithm)이라 하자. 예를 들면, 자료를 일정한 기준에 의해 순서대로 나열하는 정렬(sorting) 문제는 일상생활 뿐 아니라 정보 처리에서 중요하고 기본적 작업인데 정렬 문제를 해결하는 많은 알고리즘들이 개발되어 있다. 정렬이라는 특정한 문제에 대한 구체적이고 문제 종속적인 알고리즘들의 예를 들면 병합 정렬(Mergesort), 퀵 정렬(Quicksort) 등이 있다[23]. 병합 정렬은 동일한 크기(size)의 두 자료를 차례로 비교하면서 순서에 맞게 합치는(merge) 알고리즘이며, 퀵 정렬은 적당한 기준값을 선택하여 그 기준값을 중심으로 원래의 자료를 둘로 분할하고 분할된 각각의 자료를 다시 동일한 과정을 거치면서 더 이상 분할 할 수 없을 때까지 계속하는 알고리즘이다. 이 두 알고리즘은 동일한 문제를 다루지만 서로 다른 해결 방법을 갖는 개별적 알고리즘으로 볼 수 있다. 그러나 이 두 알고리즘은 하나의 공통적인 개념이 핵심을 이루고 있다. 즉, 원래의 문제를 작은 문제들로 나누고 나누어진 각각의 문제를 해결함으로써 원래의 문제를 해결한다는 공통적 개념이 존재한다.

이제 정렬이 아닌 다른 문제를 생각해 보자. 주어진 자료에서 특정 값에 대한 정보를 찾는 탐색 문제(search problem)를 생각해 보자. 많은 탐색 알고리즘들 중에 효율 좋은 이진 탐색(Binary Search) 알고리즘은 정렬된 원래의 자료에서 그 중앙에 위치한 값과 찾고자 하는 값을 비교하고, 그 두 값이 다르면 중앙값에 의해 분할된 두 부분 중에서 어느 한 부분만 찾는다. 그렇게 나눈 그 한 쪽 부분에서 다시 중앙값을 찾고 동일한 과정을 계속하는 알고리즘이다. 이 알고리즘에서 문제를 두 부분으로 분할해 나간다는 점이 핵심적인 개념이다. 즉, 이진 탐색 알고리즘은 문제는 다르지만 병합 정렬, 퀵 정렬과 같이 문제를 분할하여 처리한다는 동일한 개념을 사용하고 있다. 이러한 개념의 알고리즘을 특별히 분할 정복(Divide-and-Conquer)이라 부르고 있다. 이러한 개념은 포괄적이며 문제 독립적(problem-independent)이라는 성질을 갖는다는 것을 알 수 있다. 이와 같이 개념적이고 문제 독립적인 알고리즘을 개념적 알고리즘(conceptual algorithm)이라 하자.

개념적 알고리즘의 특징을 다음과 같이 정리해 볼

수 있다.

첫째, 개념적 알고리즘은 문제 해결의 집단적, 포괄적 설계에 해당한다. 개별적이고 구체적인 알고리즘은 각 개별 문제에 맞게 개별적 알고리즘을 고안해야 한다. 예를 들어 분할 정복 알고리즘의 경우, 문제를 분할하는데 언제, 어떻게 분할 할 것인지, 분할해서 처리한 결과를 어떻게 할 것인지 등과 같은 문제 종속적 사항은 각 문제의 요구에 맞추어 개별적으로 알고리즘을 고안해야 한다.

둘째, 동일한 문제에 대해 동일한 개념적 알고리즘을 사용하더라도 다수의 서로 다른 특성을 갖는 개별적 알고리즘들이 가능하다. 위의 예에서 병합 정렬과 퀵 정렬은 같은 정렬 문제에 대하여 같은 ‘분할 정복’이라는 개념적 알고리즘을 사용하지만 서로 다른 개별적 알고리즘이다.

셋째, 동일한 문제에 대해 서로 다른 개념적 알고리즘을 사용할 수 있다. 정렬의 예를 들면, 선택 정렬(Selection Sort)은 탐욕(Greedy) 알고리즘이지만 병합 정렬은 분할 정복 알고리즘이다.

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

## 3.2 개념적 알고리즘의 유형과 특징

### 3.2.1 개념적 알고리즘의 유형

활용도가 높은 개념적 알고리즘들에 대하여 각각의 개념과 활용 예를 든다. 활용 예로 사용한 알고리즘들은 알고리즘 분야에서 잘 알려진 기초적인 알고리즘들이다[8,13,19,20,21,23,24].

#### ① 반복(Iteration)

(ㄱ) 개념; 대상 자료의 각각에 일련의 동일한 연산을 동일하게 실행한다. 반복 개념이 사용 되지 않는 알고리즘은 거의 없을 정도로 가장 기초적인 개념이다.

(ㄴ) 활용 예; N개의 수를 합산하기, N개의 수에서 최대값 구하기, 두 행렬을 곱하기, 등.

#### ② 탐욕(Greedy)

(ㄱ) 개념; 문제가 원하는 최적해(optimal solution)를 구하기 위해 현재의 상황에서 가장 우수한 국소

적 최적해(globally optimal solution)를 차례로 구해 나아감으로써 결국 전역적 최적해(globally optimal solution)를 얻고자 하는 개념의 알고리즘이다. 국소적 최적해가 반드시 전역적 최적해를 보장하지는 않지만 많은 경우에 있어서 쉽게 효율적인 알고리즘을 산출할 수 있는 중요한 개념적 알고리즘이다.

(ㄴ) 활용 예; 선택 정렬(Selection Sort), 최소 비용 신장 나무(Minimum Cost Spanning Tree), 최단 경로 문제(Shortest Path Problem), 등.

### ③ 재귀(Recursion)

(ㄱ) 개념; 풀려는 문제를 직접 또는 간접으로 그 문제를 이용하여 해석하는 방법을 말한다. 다만, 무한 루프(loop)를 방지하기 위해 피정의(被定意) 부분의 자료의 수가 단조 감소(monotonic decrease)하거나 적절한 종료 조건이 부가 되어야 한다. 재귀의 개념이 알고리즘에서 중요한 이유는 첫째, 재귀적 정의에 의한 문제의 표현이 흔히 존재하고, 둘째, 문제의 성격상 재귀적 표현이 자연스러운 경우들이 많이 있으며, 셋째, 분할 정복 알고리즘에서 일반적으로 재귀적 정의를 하위 개념으로 사용하고 있다.

(ㄴ) 활용 예; 수치해석 분야의 피보나치(Fibonacci) 수열, 액커만(Ackermann) 함수, 르 장드르(Le Gendre) 다항식, 유클리드(Euclid)의 GCD 알고리즘 등과, 컴파일러 분야의 재귀적 하향 분석기(Recursive Descent Parser), 등.

### ④ 분할 정복(Divide-and-Conquer)

(ㄱ) 개념; 원래의 문제를 둘 이상의 작은 문제로 분할하는데 분할된 각각의 문제는 원래의 문제와 성격이 같다. 따라서 분할된 문제의 개수는 증가하더라도 처리하는 알고리즘은 동일(즉, 재귀적)하다. 분할된 문제는 다시 전 단계와 동일한 방법으로 복수개의 문제로 분할 가능하며 해를 구할 수 있을 때까지 분할한다. 문제의 분할 방법과 분할된 문제의 크기, 개수, 해의 병합 유무와 방법 등은 개별적 알고리즘에 종속적이다.

(ㄴ) 활용 예; 병합 정렬, 퀵 정렬, 이진 탐색, 스트라센(Strassen)의 행렬 곱, 하노이 탑 문제, 등.

⑤ 동적(Dynamic) 알고리즘 (또는, 동적 프로그래밍)

(ㄱ) 개념; 문제를 해결하는 과정에서 많은 부분해

가 중복 발생하고 부분해로부터 문제의 해가 도출될 때 한 번 계산한 부분해를 저장해 두고 필요할 때마다 그 결과를 참조함으로써 부분해의 중복 계산을 방지하는 기법이다. 이 방법은 중복이 많은 부분해를 생성하는 문제에서 중복 계산을 제거함으로써 계산 시간을 현저히 줄일 수 있다는 장점이 있다. 그러나 부분해의 발생 빈도에 비례적으로 기억 공간의 사용이 증가한다는 단점이 있다. 최근 컴퓨터 기억용량의 증가로 대부분의 문제에서 기억용량의 절약보다 계산 시간의 단축이 더 중요하게 간주되고 있으므로 동적 프로그래밍은 많은 부분해를 중복 발생하는 문제들을 효율적으로 해결하는 요긴한 알고리즘이다.

(ㄴ) 활용 예; 피보나치 수열, 행렬의 연쇄 곱(Matrix-chain Multiplication), 삼각형 분할 문제, 단계 그래프 문제, 배낭 문제(Knapsack Problem), 작업 배치 문제, 최단 거리 문제, 최적 이진 탐색 문제, 판매원 문제(TSP; Traveling Salesperson Problem), 등.

### ⑥ 백트래킹(Backtracking)

(ㄱ) 개념; 어떤 제약 조건을 만족하는 최적해를 구하기 위해 모든 가능성을 찾아야 하는 문제를 풀기 위한 개념이다. 해를 찾을 수 없다고 판단되면 그 시점에서 탐색과정을 되 돌이켜 다른 가능성을 찾을 수 있어야 한다. 효율성을 높이기 위해 불필요한 계산을 제거하는 가지치기(pruning) 기법을 함께 사용할 수 있다.

(ㄴ) 활용 예; 바둑, 장기 등 게임에서 사용하는 게임 나무(game tree)의 연산, N-여왕 문제, 부분집합의 합(Sum of Subsets), 그래프 색칠 문제(Graph Coloring), 배낭 문제(Knapsack Problem), 등.

## 3.2.2 개념적 알고리즘의 특징

각 개념적 알고리즘에는 고유한 특징이 존재한다. 상호 위계성, 논리적 난이도, 설계의 용이성, 활용성 등의 특징들은 학습의 순서를 결정 할 때 중요한 고려 사항이 된다.

(ㄱ) 반복 알고리즘; 반복 개념은 모든 알고리즘에서 활용되는 가장 기초적인 요소이다.

(ㄴ) 탐욕 알고리즘; 탐욕의 개념은 현재의 상태에서 문제가 요구하는 최적해를 찾는 것이다. 따라서

이 개념은 논리적으로 단순한 경우가 일반적이다. 이러한 단순성 때문에 알고리즘을 쉽게 이해하고 쉽게 설계할 수 있다는 장점이 있다.

(c) 재귀 알고리즘; 문제의 재귀적 표현은 알고리즘의 설계에서 빈번하게 요구 된다. 문제의 재귀적 정의는 문제를 간결하게 표현 할 수 있어 알고리즘의 고안과 이해에 도움을 준다. 분할 정복 알고리즘에서는 일반적으로 재귀적 정의를 포함하고 있으므로 재귀 알고리즘은 분할 정복 알고리즘의 선행 개념이라 볼 수 있다. 탐색, 동적 알고리즘, 백트래킹에서도 재귀 개념을 포함하여 사용하는 경우가 많다.

(d) 분할 정복 알고리즘; 원 문제가 분할 가능한지, 어떻게 분할 할 것인지, 분할된 부분을 어떻게 처리 할 것인지, 분할된 각 부분의 해를 어떻게 전체의 해로 해석할 것인지 등과 같은 많은 요소를 고려하여야 하므로 알고리즘 설계의 난이도가 일반적으로 높다고 할 수 있다.

(e) 동적 알고리즘; 부분해를 적절한 자료구조에 저장하고 필요할 때 참조하는 과정이 이 알고리즘의 핵심이다. 이 개념은 다른 유형의 알고리즘들과 충돌 없이 사용될 수 있다.

(f) 백트래킹 알고리즘; 모든 경우를 고려하여 문제가 원하는 최적해를 찾아 가는 탐색의 문제에 적합한 유형의 알고리즘이다. 다른 알고리즘들과 비교하면 범용성이 떨어지지만 처리할 수 있는 문제의 영역이 분명히 존재하는 필수적인 개념의 알고리즘이다.

## 4. 개념적 알고리즘의 학습

### 4.1 개념적 알고리즘의 학습 이유

알고리즘의 궁극적 목적은 주어진 문제의 효율적 해결에 있다. 문제 해결력은 효율적 알고리즘의 설계 능력을 요구한다. 이를 위해 개념적 알고리즘의 학습이 필요한 이유를 제시하고자 한다. 학습 내용에 대한 논의에 앞서 제3장에서 제시한 여섯 가지의 개념적 알고리즘에 대한 학습이 필요한 이유를 아래와 같이 세 가지로 제시한다.

(1) 알고리즘에 대한 종합적이고 포괄적인 개념의

학습을 위하여; 현재까지 알고리즘 학습에 대한 논의는 특정 문제에 대한 개별적 알고리즘들을 주로 다루어 왔다. 그 중 대부분은 정렬과 탐색이라는 특정 문제에 초점을 맞추어 왔다. 정렬과 탐색 문제는 전산학의 전통적인 핵심 문제로 간주 되고 있지만 그 문제 자체가 알고리즘의 원리와 개념을 드러내는 것은 아니다. 알고리즘의 원리가 그러한 문제들에 대해 어떻게 적용되는지에 대한 학습은 가능하지만 알고리즘의 개념과 원리를 포괄적이고 종합적으로 파악하기에는 미흡하다. 알고리즘학(Algorithmics)은 문제의 집단적 성질(또는 유형)을 규명함으로써 문제의 본질을 밝혀 주는 중요한 역할을 한다. 문제에 대한 분석은 세부적, 개별적인 알고리즘의 고안에 앞서 그 문제에 내재한 집단적 성질의 파악이 선행 되어야 한다. 따라서 문제 독립적이고, 포괄적, 종합적인 알고리즘의 개념이 알고리즘 학습에 필수적이다.

(2) 문제 해결의 방향 제시를 위하여; 개별적 알고리즘을 설계하는데 있어서 문제 해결의 전체적 방향을 올바르게 파악하는 것이 무엇보다 중요하다. 개념적 알고리즘은 문제의 집단적 성격에 근거한 알고리즘 설계의 원리를 규명하므로 개념적 알고리즘의 학습을 통해 개별 문제에 대한 집단적 성격을 규명하여 논리적이고 체계적인 알고리즘 설계의 방향을 확보할 수 있다. 예를 들면, 어떤 특정 문제에 대해 탐색 알고리즘으로 접근할 것인지, 아니면 동적 알고리즘으로 접근할 것인지를 판단할 수 있게 된다. 이와 같이 개념적 알고리즘은 문제 해결의 방향을 제시해주는 중요한 이정표와 같은 역할을 한다.

(3) 문제에 대한 분석력의 함양을 위하여; 개념적 알고리즘을 통한 문제 해결 방식은 문제 해결을 위한 적합한 개념 원리의 발견을 요구하므로 학습자로서 하여금 문제의 성격을 규명하기 위해 문제를 분석적으로 관찰하게 한다. 이러한 분석 과정을 통해 문제를 이해하고 분석하는 힘을 기르게 된다.

### 4.2 개념적 알고리즘의 학습 목표

각 개념적 알고리즘에 대한 학습 목표를 아래와 같이 세 가지로 요약할 수 있다.

(1) 각 개념적 알고리즘의 원리를 이해한다.

(2) 문제에서 개념적 알고리즘의 요소를 발견 할

수 있다.

(c) 개념적 알고리즘(들)을 구사하여 문제를 효율적으로 해결 할 수 있다.

### 4.3 개념적 알고리즘의 학습 내용

학습 내용을 선정하는 방향은 아래와 같다.

첫째, 본 논문에서는 학습자를 초등학교 고학년 이상의 학생으로 한다. 따라서 초등학교 고학년 학생의 인지능력에 적합한 내용을 선정한다.

둘째, 학습자 간의 수준 차이를 고려하여 수준별 학습이 될 수 있도록 학습 내용을 구성한다.

셋째, 흥미를 유도할 수 있는 내용을 선정한다.

넷째, 다양하고 창의적인 사고를 유도하도록 학습 내용을 마련한다.

각 개념적 알고리즘에 대한 학습 내용의 개요는 아래와 같다.

#### ① 반복 알고리즘

(1) 반복 알고리즘의 기본 개념을 학습 한다; 반복의 정의.

(2) 반복 요소의 발견과 알고리즘적 표현; 문제에서 반복의 요소를 발견하고, 반복 개념을 사용하여 문제를 알고리즘으로 표현한다.

(c) 오류의 발견과 처치

(d) 반복 알고리즘의 연습 문제; 반복 합산 문제, 최대값(최소값) 계산 문제, 2차원 배열에서 카운팅(counting) 문제, 등.

(e) 수준별 응용문제와 창의력 개발 문제

#### ② 탐욕 알고리즘

(1) 탐욕 알고리즘의 기본 개념을 학습 한다; 최적해, 국소적 최적해, 전역적 최적해, 탐욕 알고리즘의 장단점에 대한 이해.

(2) 탐욕 요소의 발견과 알고리즘적 표현; 문제에서 탐욕의 요소를 발견하고, 탐욕 개념을 사용하여 문제를 알고리즘으로 표현한다.

(c) 오류의 발견과 처치

(d) 탐욕 알고리즘의 연습 문제; 선택 정렬, 최소 비용 신장 나무, 최단 경로 문제, 등.

(e) 수준별 응용문제와 창의력 개발 문제

#### ③ 재귀 알고리즘

(1) 재귀 알고리즘의 기본 개념을 학습 한다; 재귀

적 정의, 재귀 함수의 실행에 대한 이해.

(2) 재귀 요소의 발견과 알고리즘적 표현; 문제에서 재귀의 요소를 발견하고, 재귀 개념을 사용하여 문제를 알고리즘으로 표현한다.

(c) 오류의 발견과 처치

(d) 재귀 알고리즘의 연습 문제; 피보나치 수열, 유클리드의 GCD 알고리즘, 사칙 연산 계산기, 등.

(e) 수준별 응용문제와 창의력 개발 문제

#### ④ 분할 정복 알고리즘

(1) 분할 정복 알고리즘의 기본 개념을 학습 한다; 분할의 의미, 병합의 유무 판단, 분할 방법, 병합 방법에 대한 이해.

(2) 분할 정복 요소의 발견과 알고리즘적 표현; 문제에서 분할 정복의 요소를 발견하고, 분할 정복 개념을 사용하여 문제를 알고리즘으로 표현한다.

(c) 오류의 발견과 처치

(d) 분할 정복 알고리즘의 연습 문제; 이진 탐색, 퀵 정렬, 병합 정렬, 등.

(e) 수준별 응용문제와 창의력 개발 문제

#### ⑤ 동적 알고리즘

(1) 동적 알고리즘의 기본 개념을 학습 한다; 중복 계산의 요인 발견, 부분해의 저장과 참조 방법.

(2) 동적 요소의 발견과 알고리즘적 표현; 문제에서 동적 요소를 발견하고, 동적 개념을 사용하여 문제를 알고리즘으로 표현한다.

(c) 오류의 발견과 처치

(d) 동적 알고리즘의 연습 문제; 배낭 문제, 최대 이윤 생산 문제, 삼각형 분할 문제, 등.

(e) 수준별 응용문제와 창의력 개발 문제

#### ⑥ 백트래킹 알고리즘

(1) 백트래킹 알고리즘의 기본 개념을 학습 한다; 제약 조건을 만족하는 경우를 열거하는 논리적 개념, 탐색 개념, 가지치기 개념의 이해.

(2) 백트래킹 요소의 발견과 알고리즘적 표현; 문제에서 백트래킹의 요소를 발견하고, 백트래킹 개념을 사용하여 문제를 알고리즘으로 표현한다.

(c) 오류의 발견과 처치

(d) 백트래킹 알고리즘의 연습 문제; N-여왕 문제, 부분 집합의 합 문제, 그래프 색칠 문제, 등.

(e) 수준별 응용문제와 창의력 개발 문제

#### 4.4 개념적 알고리즘의 학습 순서

개념적 알고리즘들 상호간의 위계성과 학습 난이도, 활용도를 고려하여 아래의 순서와 같이 학습 순서를 제시한다.

(㉠) 반복 알고리즘; 반복 알고리즘은 가장 기초적이고 쉬운 개념이므로 제일 먼저 학습한다.

(㉡) 탐욕 알고리즘; 탐욕 알고리즘은 국소적 최적해만을 추구하는 논리적 단순성으로 인해 그 개념을 습득하기 쉽고 적용 범위가 넓기 때문에 반복을 제외한 다른 알고리즘들 보다 먼저 학습한다.

(㉢) 재귀 알고리즘; 재귀 알고리즘은 분할 정복 알고리즘의 하위 개념으로 사용 되는 것이 일반적이므로 재귀 알고리즘을 분할 정복 보다 먼저 학습한다. 재귀 알고리즘은 분할 정복을 포함한 다른 개념적 알고리즘들에서도 자주 활용되는 기본적인 개념이므로 탐욕 알고리즘 다음으로 학습하는 것이 바람직하다.

(㉣) 분할 정복 알고리즘; 분할 정복 알고리즘은 동적 알고리즘이나 백트래킹 알고리즘 보다는 그 활용 범위가 넓고 논리적으로도 쉽게 학습할 수 있다.

(㉤) 동적 알고리즘과 백트래킹 알고리즘; 동적 알고리즘은 기본적으로 최적화 문제를 대상으로 나온 개념이고 백트래킹은 모든 경우를 고려해야 하는 탐색의 문제를 다룬다. 둘 사이에는 순서에 관계없이 학습이 가능하다. 그러나 순서를 가린다면 활용도가 더 높은 동적 알고리즘을 먼저 학습하는 것이 바람직하다.

#### 4.5 개념적 알고리즘의 학습 단계

본 절에서는 개념적 알고리즘을 어떠한 단계로 교육할 것인지에 대해 논의하고자 한다. 각 개념적 알고리즘에 대해 아래와 같은 네 단계의 학습 과정을 거친다.

##### ① 개념 도입 단계

(㉠) 개념의 일상적 의미의 발견; 학습하려는 개념의 사전적 뜻을 찾아보고 일상에서 그 예를 찾는다.

‘반복’ 알고리즘의 예; ‘반복’의 사전적 뜻을 찾아보고 일상에서 ‘반복’의 예를 찾는다.

기대 효과; 생활과 관련 있는 문제를 제기함으로써 학습자의 흥미와 관심을 높인다.

(㉡) 알고리즘적 의미의 파악; 용어의 알고리즘적 의미를 설명하고 토의를 통하여 일상적 의미와 비교하고 관련성을 발견한다.

‘반복’ 알고리즘의 예; ‘반복’의 알고리즘적 의미를 배우고 일상적 의미와 비교한다.

기대 효과; 일상과의 관련성을 발견함으로써 새로운 학습에 대한 흥미를 유발한다. 토의식으로 운영하여 학생들의 참여를 높인다.

(㉢) 활용 예를 통한 개념 요소의 발견; 전 단계에서 학습한 개념의 의미를 되새겨 누구나 쉽게 이해할 수 있는 활용 예들을 제시하여 새로운 개념이 알고리즘에서 어떻게 활용되는지 학생들이 스스로 발견하도록 유도한다. 교수자는 학생들이 스스로 발견할 수 있도록 도와주는 보조자 또는 조언자의 역할을 한다.

‘반복’ 알고리즘의 예;

[활용 예1]; N개의 수를 합산하기 (반복 요소; 덧셈)

[활용 예2]; N개의 수에서 최대값 구하기 (반복 요소; 비교)

[활용 예3]; 0/1 2차원 배열에서 ‘1’의 개수 구하기 (반복 요소; 비교와 합산)

기대 효과; 학습자는 실제의 문제에서 학습한 새로운 개념을 발견해 봄으로써 흥미를 유지한다. 학생들이 스스로의 발견을 통하여 자신감을 길러준다.

##### ② 기초 연마 단계

전 단계의 학습 내용을 조금 응용하여 학생들이 충분히 해결 가능한 문제들을 학생들이 스스로 풀도록 한다. 교수자는 학습자들이 문제를 해결할 수 있도록 보조자 또는 조언자의 역할을 한다.

‘반복’ 알고리즘의 예;

[기초 문제1]; N개의 수에서 짝수만 합산하기

[기초 문제2]; N개의 수에서 최빈수 구하기

[기초 문제3]; 0/1 2차원 배열에서 ‘1’로 연결된 최대의 수평 길이 구하기

기대 효과; 문제의 해결을 통해 알고리즘의 개념을 사용하는 문제 분석력을 기른다. 학습자는 기초문제를 통해 습득한 지식을 내면화하고 성취감과 자신



감을 기른다.

### ③ 응용 확산 단계

습득한 지식의 단순한 적용으로는 풀 수 없는 응용력과 사고를 요하는 문제를 제시한다. 교수자는 문제에 대한 설명, 예시 등을 통해 학습자들이 문제를 충분히 이해하도록 하되 학생들이 스스로 해결하도록 문제의 풀이에 직접 영향을 주는 도움은 피한다. 학습자가 충분한 시간을 갖고 생각하도록 학습자를 도와주며 학습자의 다양한 시도와 성과를 격려한다. 모범이 되는 결과물에 대하여 토의를 통하여 좋은 아이디어를 공유하며 더 좋은 아이디어를 모색하도록 학습자들의 참여를 적극 유도한다.

‘반복’ 알고리즘의 예; 과일 상자에 사과 10개, 배 11개, 감 7개, 귤 12개가 들어있다. 이 상자로부터 아무 과일이든 합해서 10개가 되는 모든 경우를 나열하는 알고리즘을 작성하시오. 그러한 경우가 모두 몇 가지인지를 계산하는 프로그램을 작성하시오.

기대 효과; 지식의 단순한 적용으로는 풀 수 없고 사고력을 요하는 문제를 통하여 알고리즘적 문제 분석력, 확산적 사고력, 과제에 대한 집착력을 배양한다.

### ④ 창의적 사고 단계

문제의 해결을 위하여 창의적 사고가 필요한 문제를 제시한다. 이 단계의 문제가 갖는 특징은 창의적인 다양한 풀이가 가능하며 동일한 답을 산출하더라도 알고리즘에 따라 풀이의 효율성이 상당히 달라지고 소수의 재능 있는 학생들만 풀 수 있다는 점들이다. 따라서 교수자는 모든 학습자들에게 동일하게 요구하지 말고 자원하거나 학습 성취도가 탁월한 학습자들을 선별하여 제시한다.

‘반복’ 알고리즘의 예; N개의 직사각형이 네 모서리의 좌표로 주어질 때 직사각형들이 차지하는 면적을 구하는 알고리즘을 작성하시오. 이 때 직사각형들은 서로 부분적으로 겹칠 수도 있으며, 포함 될 수도 있고, 변이나 꼭짓점이 겹칠 수도 있다. 단, 직사각형의 밑변은 가로축에 평행하다.

기대 효과; 창의적 사고의 개발, 수준별 교육, 영재의 발굴 및 훈련 등의 효과를 기대할 수 있다.

## 5. 결론

최근 프로그래밍을 통한 문제 해결력 배양에 많은 관심을 모으고 있으며 그에 따라 문제 해결의 수리 논리적 방법과 절차에 대한 분야인 알고리즘에 대한 관심이 높아지고 있는 것은 자연스러운 흐름이라 하겠다. 알고리즘 교육 관련 연구들을 분석하여 보면 대부분 특정 문제를 해결하기 위한 알고리즘들, 특히 정렬과 탐색 문제에 집중되어 있음을 볼 수 있다. 이러한 문제 종속적이고 개별적인 알고리즘의 학습으로는 체계적이고 종합적인 시각을 갖춘 알고리즘적 문제 분석력을 함양하기에 미흡하다. 본 논문에서는 개념 중심의 문제 독립적, 집단적 성질을 바탕으로 하는 개념적 알고리즘의 학습을 제안하였다.

개념적 알고리즘의 유형으로 여섯 가지(반복, 탐색, 재귀, 분할 정복, 동적 프로그래밍, 백트래킹)를 제시하였고 각각의 특징과 함께 개념적 알고리즘의 학습 목표와 학습 내용을 제시하였다. 개념적 알고리즘들 상호간의 위계성과 학습 난이도, 활용도를 고려하여 학습 순서를 정하였고 학습 단계(개념 도입 단계, 기초 연마 단계, 응용 확산 단계, 창의적 사고 단계)를 제안 하였다.

자신이 설계한 알고리즘을 프로그램으로 직접 구현하여 각종 입력에 대한 출력을 확인해 보는 것은 알고리즘의 동작을 직접 체험할 수 있을 뿐 아니라 성취감과 흥미를 고취하는 중요한 단계이다. 초등학교 고학년 이상의 수준이면 논리적으로 개념적 알고리즘의 학습이 가능하지만 대부분의 초등학생들은 프로그래밍 경험이 없어 프로그래밍에 의한 체험이 어렵다는 점이 문제점이다.

현재 국내의 많은 교육기관에서 시행하고 있는 정보영재 교육에 개념적 알고리즘의 학습을 활용할 수 있다. 국내의 많은 정보영재 교육기관에서 프로그래밍을 위주로 교육하고 있으나 알고리즘에 대한 체계적 교육은 아직 미흡하다. 따라서 기본적인 프로그래밍 기능을 갖춘 학생들이 스스로 설계한 알고리즘을 직접 프로그래밍 하여 본다면 문제해결력의 향상에 많은 도움이 될 것이다.

지금 논의되고 있는 알고리즘 교육은 문제 종속적인 개별적 알고리즘의 학습에 치중하고 있는 추세이다. 이에 대한 새로운 방향의 제시로 본 논문은 문제 독립적인 개념적 알고리즘 학습을 제안하였다. 아직

그 논거가 견고하지 않지만 컴퓨터 알고리즘 교육에 대한 새로운 방향을 제시하는데 그 의미가 있을 것으로 본다.

### 참 고 문 헌

- [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] 권량희, 임진숙, 김성식(2004), "자료구조적 알고리즘 학습을 위한 사례 중심 웹 코스웨어 설계 및 구현", 한국컴퓨터교육학회 학술발표논문집 8(2), 63-70.
- [4] 김미량(2002), "컴퓨터 프로그래밍 교육에 적용 가능한 효과적 교수방법의 탐색적 대안", 한국컴퓨터교육학회 논문지 5(3), 1-9.
- [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", 한국정보교육학회 학술발표논문집 9(2), 216-225.
- [7] 김현철(2005), "외국의 컴퓨터 교육과정 사례를 통한 지식 정보 교육과정 분석", IT 인력 양성과 컴퓨터 교육 워크샵, 한국교육과정평가원.
- [8] 박정호(2005), 컴퓨터 알고리즘, 상조사.
- [9] 송태욱, 한병래, 김태영, 송기상(2000), "정보블록 알고리즘의 학습을 위한 시각화 시스템의 설계 및 구현", 한국컴퓨터교육학회 학술발표논문집 5(2), 469-478.
- [10] 오성훈, 이애정, 이재호(2005), "과학영재교육원 기초반을 위한 초등 정보과학영재 교육과정 개발", 한국정보교육학회 논문지 9(1), 89-97.
- [11] 이길복, 전우천(2004), "초등학교 정보영재를 위한 알고리즘 학습 지원 시스템 연구", 한국정보교육학회 학술발표논문집 9(1), 296-303.
- [12] 이연정, 이철환, 한선관(2004), "객체지향기법을 적용한 초등 자료구조 학습의 지도 연구", 한국정보교육학회 학술발표논문집 9(2), 189-196.
- [13] 이재규(2003), C로 배우는 알고리즘, 세화 출판사.
- [14] 이주희, 김갑수(2006), "구체적 조작기의 초등학생을 위한 정렬 알고리즘 교수-학습에 관한 연구", 한국정보교육학회 학술발표논문집 11(2), 95-100.
- [15] 임화경, 전승순(2006), "초등 수학문제를 이용한 컴퓨터 알고리즘 개념에 대한 교수방법", 한국컴퓨터교육학회 논문지 9(3), 109-119.
- [16] 장승익, 권보섭(2003), "프로그래밍 학습과 교과학습간의 상관관계 연구", 한국컴퓨터교육학회 학술발표논문집 7(2), 328-335.
- [17] 정인기(2002), "탐색 알고리즘 교육을 위한 SW 컴포넌트 개발", 한국정보교육학회 논문지 6(2), 179-187.
- [18] 홍순조, 한선관(2004), "알고리즘 분석을 통한 컴퓨터교육에서의 알고리즘 교육의 방향", 한국정보교육학회, 학술발표논문집 9(2), 236-241.
- [19] 황종선, 정영식(1996), C언어로 설명한 알고리즘, 정익사.
- [20] R. Becker(1989), Data Structures, Prentice Hall.
- [21] T. Cormen, C. Leiserson, R. Rivest(1992), Introduction to Algorithms, McGraw-Hill.
- [22] D. Harel(1987), Algorithmics, The Spirit of Computing, Addison-Wesley.
- [23] E. Horowitz, S. Sahni(1978), Fundamentals of Computer Algorithms, Computer Science Press.
- [24] E. Horowitz, S. Sahni(1984), Fundamentals of Data Structures in PASCAL, Computer Science Press.

### 문 교 식



1982 경북대학교 공과대학 컴퓨터공학과 (공학사)

1982~1986 KIST 시스템공학연구소, 연구원

1989 University of Oklahoma 대학원 전산학과 (이학석사)

1995 University of North Texas 대학원 전산학과 (이학박사)

1996~1997 동명정보대학교 컴퓨터공학과 조교수

1997~현재: 대구교육대학교 컴퓨터교육과 부교수

관심분야: 컴퓨터교육, 알고리즘

E-Mail: gmoon@dnue.ac.kr