韓國數學敎育學會誌 시리즈 E
<數學敎育 論文集>
제 11집, 2001. 1. 145-157.

J. Korea Soc. Math. Ed. Ser. E:
Communications of Mathematical Education
Vol. 11, Jan. 2001. 145-157.

초등학교에서의 알고리즘 지도의 필요성과 지도방법

서 찬 숙 (대구상인초등학교)
남 승 인 (대구교육대학교)

학습자가 수학적 지식이 정말로 가치 있고 유용한 것이라는 실감을 갖게 하기 위해서는 학습자가 학습의 주체로써 능동적인 참여 기회와 환경의 제공해야 할 것이다. 그러나 지금까지의 수학 학습은 주로 교과서에 제시된 내용과 순서에만 의존하여 교사가 자신의 관점에 근거하여 학생들을 가르치기 위해 수업을 설계하고 실행하고 평가함으로 해서 이미 만들어진 수학을 전수 받아 이를 암기하고 반복 연습하는 경우가 많았다.

특히 수학학습에서 가장 기본·기초가 되는 알고리즘 학습의 경우 학생들이 가지고 있는 기존의 경험이나 지식에 근거하여 그들 스스로 알고리즘을 구안·적용해 볼 수 있는 기회를 통해 문제를 해결하는 경험이 중요하다고 보겠다. 이런 맥락에서 본고에서는 인간의 창조적 활동의 산물인 표준화된 알고리즘을 직접적으로 도입·적용하기에 앞서서 학습자의 수준에서 창의적으로 알고리즘을 고안·활용해 볼 수 있도록 하기 위해 초등학교 수학에서 알고리즘을 지도하는 방안에 대해 알아보고자 한다.

Ⅰ. 머리말

과학기술 문명의 발달에 따라 사회적 구조가 산업화 시대에서 정보화 시대로 변모함에 따라 급변하는 사회적 요구에 부응하기 위하여 수학교육도 내용적인 면에서뿐만 아니라 방법적인 면에서도 새로운 변화를 시도하려는 노력이 세계적으로 활발히 이루어지고 있다. 교육의 가장 기본적인 원리 중의 하나는 능동적인 학습을 유발시키는 것이다. 근래 수학교육에서 뿐만 아니라, 발달 심리학, 종족 이론, 성 심리학, 컴퓨터 공학, 등 여러 학문 분야에서 대중적인 위치에 있는 구성주의의 기본 가정은 "모든 지식은 환경으로부터 감각이나 의사 전달을 통해 수동적으로 받아드려지는 것이 아니라 학습자에 의해 능동적으로 구성되어진다"(Kilpatrick, 1987)는 주장이 학교 교육에서 강력하게 지지되고 있다. 이는 능동적인 정신 작용에 의해 획득된 지식이 가장 건전하며, 어떠한 사태나 현상에 대해서도 적용 가능할 뿐만 아니라 새로운 지식을 생성하는 모태가 되기 때문이다. 그러나 지금까지 우리의 수학 학습 실태의 단면을 살펴보면 교과서나 지도서에 제시된 학습 목표 및 내용에 따라 교사가 자신의 관점에 근거하여 학생들을 가르치기 위해 수업을 설계하고 실행하고 평가해 왔다. 그로 인해 요즈음의 어린이들은 탐구하고 사고하는 기회를 통해 스스로 수학적 지식을 터득해 가는 것이 아니라, 이미 만들어진 수학을 교사나 교수 매체를 통해 전수 받아 이를 암기하고 반복 연습하는 교과목으로 인식하고 있다. 그러나 수학적 지식이 정말로 가치 있고 유용한 것이라는 실감을 갖게 하기 위해서는 학습자가 스스로 학습 목표를 인식하고 수업에 직접 참여하여 이끌어 가는 동안에 의지적, 의욕적으로 어려움을 극복하고 지속적으로 학습할 수 있는 기회와 환경이 만들어져야 할 것이다.

특히 수학 학습의 가장 기본·기초가 되는 알고리즘 학습의 경우 학생들이 갖고 있는 기존의 경험이나 지식에 근거하여 그들 스스로 알고리즘을 구안·적용해 볼 수 있는 기회는 박탈당한 채, 오랜 세월을 거쳐 변형·발전된 산물인 교과서의 제시된 표준화된 알고리즘을 안내·강요받음으로 해서 학생들은 잘못된 수학관과 학습관을 갖고 있다. 스켐프는 수학학습에서 진정한 학습은 관계적 이해와 관계적 스키마의 구성이라고 하였다. 이는 학습자가 처음 대하게 되는 개념을 기존에 가지고 있던 개념들과 적절한 관계적 스키마를 연결시키는 것을 의미하는데, 학생들이 이미 가지고 있던 경험과 지식을 존중하는 학습을 강조하는 것이다.

그러나 전통적인 수업은 학생이 스스로 문제를 해결하기보다는 표준화된 방법을 안내하여 익숙해지도록 하는데 중점을 두었다. 그래서 학생들은 문제를 스스로 탐구하고 사고할 수 있는 기회를 가지지 못하고 수학은 오로지 암기하고 반복 연습하는 과목으로만 인식하고 있다. 이런 학습 상황에서 학생 자신이 고안한 알고리즘을 적용하여 문제를 해결하는 경험을 한다는 것은 지극히 어렵다. 유의미한 학습이 이루어지기 위해서는 학습자 자신이 학습의 주체가 되어 스스로 문제 해결 방법을 고안·적용해 보는 자기주도적 학습 경험이 필요하다. 이와 같이 학습자의 능동적인 정신작용에 의하여 획득된 지식이 구조화되어 어떠한 현상에도 적용 가능할 뿐 아니라 또 다른 지식 구성의 모태가 되기 때문이다.

이런 맥락에서, "문제를 해결하기 위한 일련의 과정 또는 방법"을 의미하는 알고리즘의 학습은, 인간의 창조적 활동의 산물인 표준화된 알고리즘을 직접적으로 도입·적용하기에 앞서서 학습자의 수준에서 창의적인 알고리즘을 고안·활용해 볼 수 있는 기회의 제공으로써 필요하다고 생각된다. 일반적으로 알고리즘이라고 하면 교과서에 제시된 표준화된 알고리즘만을 떠올리기 쉽다. 그러나, 알고리즘이 문제 상황에서 해에 이르는 절차에 지나지 않는다는 것을 고려할 때, 하나의 문제를 해결하기 위한 알고리즘이 오직 하나만 존재하는 것은 아니다. 따라서, 학생 모두가 동일한 알고리즘 즉, 표준화된 알고리즘만을 기계적으로 배운다는 것은 큰 의미가 없다. 오히려 학생들이 다양한 알고리즘을 탐구하며 문제를 해결함으로써 수 감각을 기르고 문제를 다양한 관점에서 접근할 수 있게 된다. 또, 여러 가지 알고리즘을 학생들끼리 의사소통하는 과정에서 자신이 만든 알고리즘의 논리적 타당성을 검증할 수 있으며, 더 나아가 표준화된 알고리즘과 자신이 만든 알고리즘을 비교하여 표준화된 알고리즘이 가장 세련되고 신속·정확하다는 것을 깨닫는 기회를 가질 수 있다. 그러나 많은 교사들이 교과서에 제시된 표준화된 알고리즘만을 암암리에 학생들에게 강요하고 있는 실정이다.

이에 본고에서는 NCTM 규준과 알고리즘 학습, 알고리즘의 의미와 종류, 알고리즘의 교육적 가치, 학생이 고안한 알고리즘의 장점과 학생이 알고리즘을 고안하도록 돕기 위한 교사의 지도 방법, 마지막으로 학생이 고안한 알고리즘에 대한 수학적인 평가 기준에 대해 생각해 보겠다.

II. NCTM 규준과 알고리즘 학습

지금까지의 전통적인 수업이 알고리즘의 암기와 적용을 지나치게 강조 즉, 지필에 의한 반복 연습

만을 생각해 왔다. 이로 인해 수학 학습에 부정적인 인식을 심어 준 것도 사실이다. 그러나, 모든 수학학습의 기본·기초 기능에 대한 알고리즘의 지도는 필수 불가결한 것이다. 지금까지 알고리즘 지도의 문제점 중의 하나는 의미없는 계산력 강조에 따른 지루함에서 비롯된 것이었다. 이에 NCTM(1989)은 알고리즘의 지도는 그 필요성을 제기하는 문제 상황으로부터 발생되어야 한다고 이야기하고 있다. 이는 일방적인 암기와 적용이 아닌 문제 상황으로부터의 필요성을 통한 알고리즘의 학습을 강조하며 학생들이 기본적인 알고리즘을 숙달하고 일상생활에서 알고리즘의 유용성을 이해해야 함을 의미한다. 또 5-8학년의 수학 내용과 수업에 있어서 약화되어야 할 것은 '규칙과 알고리즘을 암기하는 것', '지필 알고리즘 연습'이며 '알고리즘과 처리 절차를 만들어 내는 것'은 강화되어야 한다고 하였다.

개인에 따라서 알고리즘을 보는 시각에 다소 차이가 있겠지만, Barnett(1998)는 알고리즘이란 주어진 문제에 대한 절차적인 과정이 있어 그 과정을 정확하게 실행했을 때 정확한 답을 얻을 수 있게 하는 절차를 말한다고 하였다. Carroll과 Porter(1998)는 절차를 상황에 따라 알고리즘과 동의어로 사용하기도 하였다. 이런 관점에서 NCTM (1998)이 제시한 알고리즘 관련 목표를 살펴보았다.

기본적인 계산 지식과 알고리즘을 모델링하고, 설명하고, 효율적으로 운용할 수 있어야 한다.
해와 전략을 새로운 문제상황에 일반화할 수 있어야 한다.
계산 과정과 어림 기법을 개발하고, 분석하고, 설명할 수 있어야 한다.
알고리즘을 개발하고 분석할 수 있어야 한다.
알고리즘의 응용과 컴퓨터 타당화 과정과 관련하여 생기는 문제상황을 탐구할 수 있어야 한다.

따라서 초등학교보다는 중등학교에서 알고리즘에 대한 내용을 많이 다루게 되나, 알고리즘의 학습은 전학년을 통해서 강조해야 한다. 특히 학생들이 알고리즘을 고안하고 분석하며 의사소통하는 것은 수학적인 아이디어를 명료하고 분명하게 이해하게 하며 논리적인 추론을 향상시키는 상황을 제공해 준다.

Ⅲ. 알고리즘의 의미와 종류

알고리즘은 al-khwarizmi라는 수학자의 이름에서 유래되었다. 그는 Kitab al-jam wal tafrig bi hisab al-Hind(인도 사람들의 방법에 따른 덧셈과 뺄셈)라는 저서에서 십진기수법으로 수(number)를 표기하는 방법과 연산방법을 제시하였다. 인도의 십진법 이전에 서양에는 그리스 수 체계(서로 다른 알파벳 문자가 다른 수를 상징함)와 로마 수 체계(IX=10-1을 표현하는 등 grouping 체계 사용)가 있었지만, al-khwarizmi가 인도의 십진법을 소개하여 서구의 수(number)체계가 비로소 현대적인 십진법을 택하게 되었다. 그의 이름이 라틴어로 번역될 때 algorismus라고 번역되었고 이 라틴어에서 algorithm(알고리즘)이라는 용어가 생겨났다.

알고리즘은 컴퓨터를 이용한 문제해결에서 프로그래밍의 핵심이 되기 때문에 최근 더욱 강조되고 있다. 수학은 실용적인 '계산술'로써 출발하였으며, 계산력은 수학의 바탕으로 오늘날에도 변함없이 중요하다. 알고리즘은 수학에 강력한 힘을 주는 동시에 개념적 깊이를 더해주는 기술적 수단이기도 하다. 오늘날 algebra라고 불리는 수학의 분야는 계산이 중심이며 아라비아어에 그 어원이 있다는 것은 잘 알려진 사실이다. 12세기에 이슬람 문화의 산물인 인도-아라비아식 기수법을 사용하는 놀라운 계산법이 서구에 전해지면서 수학의 연구는 도형의 성질을 연구하는 기하에서 기호를 조작하는 계산법, 곧 대수에로 그 중심이 옮겨졌다. 16세기에 프랑스의 수학자 Viete가 대수에 문자를 도입하면서 계산법의 힘을 더욱 강력한 위력을 발휘하게 되었으며, 17세기 초반에 Descartes는 방정식을 이용한 계산법에 의해 도형의 성질을 연구하는 방법 곧, 해석기하학을 탄생시켜 기하를 대수화함으로써 현대 수학과 과학의 발전을 위한 기틀을 마련하였다.

앞에서 언급했듯이 알고리즘은 수학적인 문제에 대한 해라고 일컬어질 정도로 수학에서 중요한 위치를 차지해 왔지만, 그 전통적인 의미와 현대적인 의미는 다소의 차이가 있다.

Barnett(1998)는 전통적인 의미에서 알고리즘이란, 주어진 절차를 정확히 실행하기만 하면 주어진 문제에 대해 옳은 답을 보장하는 점진적(step-by-step)인 과정이라고 정의한 반면, Maurer(1998)는 현대적인 입장에서 알고리즘을 정의하고자 하였다. 그에 의하면 알고리즘이란 문제를 해결하기 위한 구체적이고 체계적인 방법이다. 알고리즘은 투입을 하면 정해진(determine) 규칙의 집합을 따라 결정적인(conclusive) 답을 제공하는 산출물(output)이 유한 개가 나온다. "정해진(determine)" 규칙이란, 유한 개의 규칙이 있어 처음의 규칙이 무엇이며, 다음에 적용될 규칙이 무엇인지 순서대로 정해져 있다는 의미이다. "결정적인(determine)" 답이란 알고리즘을 정확히 수행하면 항상 참이거나 또는 항상 거짓임이 결정되어 있다는 뜻이다. 따라서, 그 알고리즘이 문제를 옳게 해결한다거나 해결할 수 없음을 의미하는 것은 아니라고 하였다.

Barnett과 Maurer의 의견을 종합해 보면, 알고리즘이 문제를 해결하기 위한 일련의 과정 또는 방법이라는 점에서는 공통적이다. 그러나 전통적인 알고리즘은 문제에 대한 옳은 답을 보증하는 반면, 현대적인 의미의 알고리즘은 옳은 답을 보증하지는 않는다. 알고리즘은 옳은 답, 또는 틀린 답을 얻게 한다. 즉, 현대적인 의미에서 볼 때, 학생들이 고안한 해결의 과정이 "알고리즘인가, 아닌가"가 보다는, "좋은 알고리즘인가 좋지 못한 알고리즘인가"라는 질문이 제기된다.

학교 수학에서는 사용되는 대부분의 알고리즘은 지필 알고리즘이며, 그 종류도 다양하기 때문에 알고리즘이라고 하면 흔히 지필 알고리즘만을 떠올리기가 쉽다. 그러나 문제에 대한 해를 얻기 위한 절차를 알고리즘이라고 하면, 현대와 같은 정보화 시대에는 지필 알고리즘 뿐만 아니라 계산기와 컴퓨터를 이용한 알고리즘도 한 부분을 차지한다. 그래서 알고리즘의 종류는 크게 지필 알고리즘과 계산기 · 컴퓨터 알고리즘, 이 두 가지를 드는 데 Usiskin(1998)은 지필 알고리즘을 산술 알고리즘, 대수와 미적분 알고리즘, 그림 알고리즘 세 가지로 분류하고 있다. 이를 간략히 살펴보면 다음과 같다.

① 산술(Arithmetic) 알고리즘에는 나눗셈의 세로셈, 곱셈의 세로셈, 덧셈의 세로셈, 자리수가 많은

뺄셈, 분수의 나눗셈과 곱셈, 평균과 표준 편차의 계산, 제곱근 찾기 등이 있으며 ② 부분 분수의 조작, 제한된 정수 계산, 1차 방정식과 부등식의 해결, 최적의 직선과 곡선 결정하기 등에 사용되는 알고리즘은 대수와 미적분 알고리즘이고, ③ 그림 알고리즘에는 막대 그래프, 원 그래프, 비교 그래프, 함수 그래프, 관계 그래프 그리기와 기하학에서 자와 컴퍼스를 사용하여 작도하기, 그리고 도형의 변환찾기 알고리즘 등이 속한다.

Ⅳ. 알고리즘의 교육적 가치

앞에서 언급했듯이 알고리즘은 문제 해결의 일련의 절차, 과정인데 Usiskin(1998)은 그 교육적인 가치를 다음과 같이 9가지를 제시하였다.

첫째, 알고리즘은 강력하다.

알고리즘은 한편으로 문제해결 과정의 일반화를 의미한다. 우리가 한 문제에 대해 해답을 보증하는 알고리즘을 알고 있다면, 그와 유사한 종류의 모든 과제를 해결할 수 있다. 그러므로, 알고리즘의 힘은 그 적용 범위에 따라 결정된다.

예를 들어, 자연수의 덧셈에 대한 훌륭한 알고리즘은 소수의 덧셈에도 적용이 될 만큼 강력하다. 그러나, 계산기로 하는 나눗셈의 알고리즘은 계산기의 기능에 따라 제약을 받기 때문에 나눗셈의 지필 알고리즘보다 강력하지 못하다.

둘째, 알고리즘은 신뢰할 만하다.

어떤 알고리즘이 정확한 답을 보증해 주는 것이라면 몇 번을 반복해도 정확한 답을 얻을 수 있어야 한다. 하나의 알고리즘을 실행하는 데 있어 실수할 가능성이 크다면, 그 알고리즘은 유용하지 못하며, 신뢰도가 떨어진다. 그래서 일반적으로 지필 알고리즘보다 컴퓨터나 계산기 알고리즘을 선호한다.

셋째, 알고리즘은 정확하다.

좋은 알고리즘일수록 더 정확하다. 이는 곧 답의 정확도와 관계가 있다. 그래서 대부분의 학생들은 정확한 답을 얻을 수 있는 알고리즘을 알고 있을 때 어림에 만족하지 않고 정확한 계산을 한다. 예를 들어, 60×28을 상황을 제시하지 않고 어림을 하게 하면 학생들은 계산을 한다. 이는 정확한 답을 얻을 수 있는 알고리즘을 이미 알고 있기 때문이다.

넷째, 알고리즘은 빠르다.

잘 고안된 알고리즘은 답에 이르는 길을 직접적으로 제시하고, 더 직접적인 알고리즘은 각각 떨어진 단계를 실행하는 것보다 훨씬 시간을 절약해 준다. 가장 좋은 것은 빠르게 적용할 수 있고 외우기 쉬운 알고리즘이지만, 외우기가 어려워서 잊어버린 뒤 필요할 때마다 알고리즘을 책에서 찾거나 유도하면 속도가 떨어진다.

다섯째, 알고리즘은 문자 기록을 남긴다.

문제의 답을 얻기 위해 이용된 알고리즘의 기록은 지필 활동, 계산과정을 남기는 컴퓨터 프로그램

과 계산기 등에서 쉽게 볼 수 있다. 이러한 문자의 기록은 학생들이 자신의 활동 과정을 반성하고 세련되게 하며 서로의 알고리즘을 공유하는 데 유용하다.

여섯째, 알고리즘은 정신적인 상(mental image)을 만든다.

지필로 기록된 알고리즘을 통해서, 연필과 종이 없이 결과를 얻을 수도 있다. 예를 들어 481÷13을 계산할 때, 13×3=39이므로 481에서 8 위에 3을 쓰고, 481의 아래에 39(390)를 쓰면 나머지가 91이 된다. 13x7=91이므로 481의 1위에 7을 쓰고, 91 아래에 91을 쓴다. 따라서 몫은 37이 된다.

그러나, 위와 같은 인지적인 상을 만들기까지는 아주 많은 시간이 필요하다.

일곱째, 알고리즘은 유익(instructive)하다.

문제에서 주어진 정보와 답 사이의 관계에 대한 통찰을 제공하는 알고리즘도 있다. 예를 들어, 자연수의 덧셈에서 받아올림의 알고리즘은 자리값의 개념을 적용하는 방법을 보여주기 때문에 유익하다. 또, $(a-b)(a+b)=a^2-ba+ab-b^2=a^2-b^2$ 은 중간 과정이 분배 법칙의 이용방법을 나타낸다.

그러나, 유의해야 할 것은 각 단계 사이의 이동이 명백히 알 수 있는 것임에도 불구하고 추가적인 단계를 필요이상으로 넣으면, 학생들은 본래의 문제와 멀어지게 된다는 점이다.

여덟째, 하나의 알고리즘이 다른 알고리즘에 사용된다.

하나의 상황에서 사용된 알고리즘이 다른 상황에서 그 알고리즘의 부분으로 사용될 수 있다. 따라서, A알고리즘이 B알고리즘보다 중요할 수도 있다. 초등학교에서 배운 나눗셈의 세로셈 알고리즘은 나중에 다항식의 나눗셈에 사용되고, 평행사변형을 그리는 알고리즘이 직육면체를 그릴 때 사용된다.

아홉째, 알고리즘은 학습의 대상이 된다.

알고리즘은 문제의 해에 이르는 과정일 수도 있지만, 한편으로 학습의 대상이 된다. 말하자면, 알고리즘의 효율성, 수학적 특성 등을 비교해 볼 수 있다.

예를 들어, 분수의 나눗셈을 $\frac{a}{b} \div \frac{c}{d} = \frac{\frac{a}{b}}{\frac{c}{d}} = \frac{\frac{a}{b} \cdot bd}{\frac{c}{d} \cdot bd} = \frac{ad}{bc}$ 와 $\frac{a}{b} \div \frac{c}{d} = \frac{a}{b} \cdot \frac{d}{c} = \frac{ad}{bc}$ 의 두 가지 방법으로 하였다. 앞의 식이 유익하고 기억하기 쉬울 수도 있지만, 뒤의 식이 쓰기에는 경제적이다.

V. 알고리즘의 지도방안

과거의 수학교육은 학생들이 표준 알고리즘을 학습하여 연습하는 것을 강조했지만, 최근에 와서는 추론, 문제해결, 개념적인 이해에 관심을 쏟고 있다. 특히, 수와 연산에 대한 학생의 이해를 향상시키는 방법 가운데 한 가지는 학생들에게 의미 있는 계산절차(알고리즘)를 스스로 고안해 내도록 하는 것이다. Carroll과 Porter(1997)에 의하면 학생들은 간단한 곱셈이나 나눗셈에서 자신에게 의미 있고 정확한 절차를 만들어 냈다고 한다. 그들은 학생들이 그 절차를 융통성 있게 다룰 수 있을 때, 그것

을 표준 알고리즘과 구별하여 "고안된 알고리즘"이라고 했다.

그러나 알고리즘을 만들어 내는 것이 이해를 발달시키는 유일한 방법은 아니며, 학생들이 표준 알고리즘을 선택했을 때조차도 그것을 사용하지 못하도록 해서는 안 된다. Carroll과 Porter는 고안된 알고리즘이 이해를 촉진시키는 이유를 다음의 세 가지를 들었다.

첫째, 고안된 절차(알고리즘)는 의미 있는 활동으로써 수학적인 아이디어를 자극한다.

대부분의 교사 또는 일반인들이 학습한 계산을 위한 표준 알고리즘은 1000년경까지도 논쟁의 대상이었으며 15세기까지도 받아들여지지 않았었다. 계산의 표준 알고리즘이 효과적인 기술이기는 하지만, 그것이 채택된 이유는 종종 애매 모호할 때가 많다. "121-78의 경우, 12에서 1을 빌려와서 11에서 8을 뺀다." 이러한 절차를 우리가 학습한 이유는 그것이 우리 자신에게 의미롭기 때문이 아니라 단지 정확한 답을 얻을 수 있기 때문이었다. 121-78에서 78, (+2)80, (+10)90, (+10)100, (+10)110, (+10)120, (+1)121로 더해서 +2+10+10+10+10+1=43이기 때문에 121과 78의 차가 43이 됨을 알 수 있다. 이 경우 두 수의 관계뿐만 아니라 거리까지도 알 수 있다. 또 adding-up 전략은 뺄셈보다 쉽고 실수할 확률도 낮다.

또 다른 방법은 수를 분리시켜서 121-78=(100-78)+21=22+21=43 의 방법으로 뺄셈을 하는 것이다. 이 경우에는 문제를 풀기가 훨씬 수월하여 쉬운 문제가 되며, 학생들의 단순한 계산 기능과 아울러 수감각도 요구된다.

어떤 경우에든, 학생들이 그들만의 알고리즘(절차)을 고안하도록 하기 위해서, 교사는 계산을 수행하는 것을 강조하기보다 계산 전략을 고안하는 데 관심을 두어야 한다.

둘째, 서로 다른 문제는 다른 방법으로 해결하기가 쉽다.(문제가 다르면 최선의 해결 방법도 달라진다) 표준 지필 알고리즘의 최대 강점은 모든 문제에 적용된다는 점이다. 그러나, 그 알고리즘이 항상 최선의 방법은 아니다. 예를 들어, 37×28의 문제가 주어졌을 때, 표준 알고리즘으로 해결하면 표 1과 같다. 그러나, 40×3의 경우 위의 방법보다는 40+40+40=120이 더 훌륭한 방법일 수 있다. 극단적인 예로 700×400의 경우 표 2의 방법으로 해결했다고 하면, 표준 알고리즘을 과신한 결과라고 할 수 있다. 이러한 문제를 극복하기 위해서는, 학생들이 저학년때부터 다양한 알고리즘과 전략을 고안하고 탐구할 수 있는 기회를 부여하여 문제 해결방법을 선택하는 데 융통성을 발휘할 수 있도록 해야 한다. 즉, 하나의 문제에 최선의 해결방법 하나를 선택하도록 해야 한다.

<표 1>

$$\begin{array}{r} 37 \\ \underline{\times\ 28} \\ 296 \\ \underline{74\ \ } \\ 1036 \end{array}$$

셋째, 학생 스스로 고안한 알고리즘은 학생들의 내적인 특성에 부합된다.

많은 연구결과들을 보면 덧셈과 뺄셈 문제에서 학생들은 오른쪽에서 왼쪽으로 계산하기보다는 왼쪽에서 오른쪽으로 계산하는 경향이 있다. 24+37에서 표준 알고리즘을 학습하기 전, 대부분의 학생들

은 20+30=50, 4+7=11이어서 50+11=61이 된다고 계산한다. 이 방법은 묶기를 배운 학생들에게조차도 의미가 있다. 뿐만 아니라, 2+3보다 20+30이라고 하면 자리값을 상기시키는 데도 도움이 된다. 학생들은 20+30을 하면서 일의 자리보다는 답에 큰 영향을 주는 십의 자리에 더 관심을 갖게 된다. 실제로 초등학교 때 표준 알고리즘을 배운 학생보다 위와 같은 방법으로 수를 학습한 학생이 머릿속으로 수를 조작하는 능력이 뛰어나다고 한다.

<표 2>

700
× 400
000
000
2800
280000

학생들이 직접 알고리즘을 고안하면 의미있는 수학 활동이 되며, 하나의 문제에는 그 문제에 최적인 알고리즘이 존재하고, 학생이 고안한 알고리즘은 그들의 내적인 특성에 부합하기 때문에 학생들이 나름대로 알고리즘을 고안하는 것이 좋다. 그렇다면 어떻게 학생들이 알고리즘을 고안하도록 할 것인가?

1) 학생들이 자신만의 방법을 탐구할 시간을 제공한다.

특히, 초등학교 학생들에게는 시간을 주어 문제를 탐구하며, 부담없는 상황에서 해결 방법을 찾도록 하는 것이 중요하다. 소집단을 활용하면 시간이 많이 걸리는 과제에서도 성공을 거두기가 용이하다. 성공적인 수업이 되려면 ① 학생들에게 문제를 제시하고, ② 개별적으로 또는 소집단으로 문제 해결방법을 고안해 낸 다음, ③ 마지막으로, 각자가 사용한 다양한 방법을 토론한다. 그 토론을 통해서, 학생들은 대안적인 접근방법을 제시하고 다른 사람의 방법으로부터 배운다. 이 때, 5분만에 20문제를 풀기보다 시간은 많이 걸리겠지만 상황이 자세하고 풍부한 문제 1-2개를 푸는 것이 더 효과적이다.

유의해야 할 것은 두 자리 이상의 뺄셈이나 곱셈의 경우, 모든 학생들이 자신만의 알고리즘을 고안해 내지는 못한다는 점이다. 이 때 교사는 짝을 짓거나 소집단을 구성할 수도 있고, 교사가 생각하기에 성공적인 방법을 제안할 수도 있다. 모든 학생들이 같은 수준에 있는 것이 아니므로 어떤 학생은 건너뛰어 세기와 같은 아주 세련된 알고리즘을 고안할 것이고 또 어떤 학생들은 시간이 많이 걸리는, 하나씩 세는 방법을 고안할 수도 있다.

2) 학생들의 사고작용을 뒷받침해 줄 수 있는 구체적인 조작을 한다.

학생들은 기본구구를 외우거나 지필 기호를 사용하기 전에 구체물로 상황을 모델링하여 수에 대한 문제 해결을 학습한다. 모델링을 촉진하면 학생들이 상황에 대한 이해를 높이고, 실수를 줄일 수 있을 뿐만 아니라 적절한 해를 구하는 데 도움이 된다. 구체물을 사용하지 않을 경우에, 저학년 학생들은 "연필이 19cm이고 지우개가 8cm일 때, 연필이 지우개보다 얼마나 클까요?"라는 문제를 어려워한다. 유치원 학생들도 19와 8을 정육면체나 다른 구체물로 표현할 수는 있지만, 두 수를 비교하는 데에는 별 관심이 없다. 학생들이 겪는 어려움의 대부분은 산술계산의 문제라기 보다는 문제 상황을 표현하는 데 있다. 따라서 구체적인 조작은 문제 상황을 구체화하고 해답에 대한 힌트가 될 수 있다.

또 다른 이유는 구체물 조작이, 정확한 절차(알고리즘)를 선택하는 데 도움이 되기 때문이다. 산가

지를 이용한다면 34-18=24라는 실수를 범할 가능성이 훨씬 줄어들 것이다. 그러나, 주의해야 할 것은 산가지나 계산기와 같은 구체물은 도구의 하나일 뿐이기 때문에 어떤 학생들은 그 도구를 사용하는데 도움이 필요할 수도 있고, 반면에 뛰어난 학생들은 구체적인 조작 대신에 암산으로 계산할 수도 있다는 사실이다.

3) 기본구구단만이 아니라 구구단을 사용할 수 있는 전략을 지도한다.

학생들이 두 자리 수 이상의 연산 문제를 정확하게 풀려고 할 때는 기본 구구를 이해할 필요가 있다. 학생들이 세어서 해결하기와 파생된 전략을 동시에 사용하면 더 효과적이다. 실제로, 성인들도 9×12=9×10+18과 같은 전략을 기본구구와 함께 사용한다. 특히 10에 가까운 수에서 파생된 연산에 통달하면, 대안적인 곱셈절차(알고리즘)를 고안하는 데 도움이 된다.

예를 들어, 16-8=16-(6+2)=(16-6)-2로 계산하면 훨씬 쉽고 빨리 해결할 수 있다. 곱셈에서도 같은 방법을 적용할 수 있다. 구구단을 외우는 것도 중요하지만 구구단을 의미있는 절차로 이해하기 시작할 때에도, 파생된 곱셈방법을 탐구하는 것이 좋다.

4) 의미있는 상황에서 문제를 제시한다.

의미있는 상황에서 제시된 문제는 학생들에게 동기를 부여하고, 수학을 기호의 조작이라기보다는 학생들이 실제 상황에 적용하는 문제 해결로 생각하도록 도와준다. 교사가 만들어 문제를 제시할 수도 있지만 학생이 스스로 문장제 문제를 만들어 해결할 수도 있다. 해결한 문제를 함께 공유하면서 다른 학생이 대안이 될 만한 방법을 제안할 수 있다. 이러한 토론은 학생들이 실수한 것을 바로 잡아줄 뿐 아니라, 다른 방법도 접할 수 있는 기회를 제공한다. 학교에서는 도입 부분에서 기호가 포함된 문제를 제시하고 문장제 문제는 정리 부분에서 접하는 경우가 대부분인데 그 순서를 바꾸는 것이 더 효과적인 경우가 많다. 구체물 조작과 함께 상황에 의미를 부여하면, 실수할 가능성이 낮아진다. 또, 학생들이 만들어 낸 문장제 문제는 학생의 이해 정도를 교사가 알 수 있는 정보를 제공한다.

5) 학생들이 전략을 공유하도록 한다.

학생들이 서로의 전략을 공유하기 위해서는 모험을 할만큼 편안한 분위기를 조성해야 한다. 학생들이 부모나 친구들에게 편지를 쓰도록 하여 전시하거나, OHP를 이용해 학급 전체에게 설명을 하여 토론을 할 수도 있다. 또는 학급 게시판에 "전략용" 코너를 마련해 학생들의 다양한 전략을 소개함으로써 절차(알고리즘)를 고안하지 못한 학생이나 잊어버린 학생들에게 도움을 줄 수 있다.

VI. 알고리즘 지도의 실제

일반 학급의 계산 원리 지도에 있어서도 '표준화된 알고리즘'을 지도할 경우, 우선적으로 학생들 스스로 주어진 장면의 문제를 해결하기 위해 자신의 방법을 이용하여 해결해 볼 수 있는 기회와 환경을 제공한 후, 이것과 교과서나 참고서, 교사의 풀이 방법,…, 등을 비교・검토하도록 하여 어느 방법이 가장 효과적인 판단하도록 하고 이를 활용하게 할 필요가 있다. 다음 표에서 세 학생이 주어진

문제를 해결하기 위해 사용한 알고리즘을 살펴보자.

<table>
<tr><td colspan="4">• 산가지 156개를 한 묶음에 4개씩 묶으면 모두 몇 묶음이나 되겠는가?</td></tr>
<tr>
<td><학생 A>의 계산
156÷4 = 10   25
-40   10
116÷4 = 4   + 4
-16   39
100÷4 = 25
-100
0
위 학생은 '156÷4'를 '156개에서 4개짜리 묶음을 몇 개 만들 수 있는가?'로 이해하고, 먼저 쉬운 것부터 이용한 것 같다. 즉 4×4, 4×10와 4×25=100이라는 사실을 이용하기 위하여 156에서 100이 될 때까지 4씩 몇 묶음을 뺀 후, 4개씩 묶음의 수를 합하여 39 묶음이라는 사실을 찾아냈다.</td>
<td><학생 B>의 계산
156
- 4 … 1
152
- 4 … 1
148
- 4 … 1
144
- 4 … 1
140
- 4 … 1
136
………
위 학생은 156으로부터
4씩 연감산을 통해 156에는 4가 39번 들어있음을 알고, 156을 4묶음으로 나누면 모두 39묶음임을 알아냈을 것이다.</td>
<td><학생 C의 계산>
1 2 3 4
20 20 20 20
5 5 5 5
10 10 10 10
1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1
위 학생은 먼저 4개의 열을 만든 후 자기가 암산하기 쉬운 20개씩 먼저 분배한 후, 남은 것을 5개, 10개 차례로 분배하고, 나머지가 없을 때까지 1개씩 분배하여 각 열에 분배된 개수의 합이 39임을 알고, 한 묶음에 39개라는 사실을 알아냈다.</td>
<td><표준화 된 방법>
39
4) 156
12
36
36
0
왼쪽 두 방법에 비해 가장 간단, 명료, 신속하게 구할 수 있는 방법으로 교과서에 제시된 방법의 하나이다.</td>
</tr>
</table>

물론 위 <학생 A, B, C의 방법>은 효율적인 방법이 아니다. 만약 계산 속도의 효율성이 목적이라면 이들은 더 공부를 해야 한다. 그러나 두 학생 모두 그들 자신이 의미있는 계산 개념을 분명히 구조화했으며, 그들 스스로 문제를 해결해 낼 수 있다는 자신감과 믿음과 이해력을 보여준 것으로 해석할 수 있다. 우리는 이와 같이 학생 스스로 문제를 해결해보는 기회나 환경을 제공하는 데 인색하며, 불필요하다거나 귀찮다고 생각하고 일반적으로 <표준화된 방법>을 학생들에게 강조·강요한다. 또 그것이 수학을 가르키는 바른 지도법(正導)인양 생각하지만 오스벨의 선행조직자의 원리나 스켐프의 관계적 이해를 강조한 학습 지도 및 구성주의자들이 권고하는 학습지도 원리와는 거리가 있다.

그러나 구성주의적 관점에서의 수업에서 무엇보다 중요한 것 중의 하나는 '문제의 해에 이르는 길은 다양하다'는 인식을 갖도록 하는 일이다. 따라서, 학생들이 나름의 다양한 알고리즘을 고안해 서로 의사소통함으로써, 또다른 방법으로 찾아보게 할 수 있으며, 그 결과를 이용하여 형식화된 수학을 이끌어 낸다면 학생들에게 더욱 의미가 있을 것이다. 그러므로 학생들이 스스로 만든 문제해결 전략이나 원리·법칙이 가장 세련된 현대의 방법과는 다르더라도 그것이 수학적으로 논리적인 모순이 없다면 매우 존중할 가치가 있는 것이며, 이는 더욱 권장해야 할 창의적인 사고력 신장의 원천이다.

Ⅶ. 학생들이 고안한 알고리즘에 대한 수학적인 평가 기준

교사는 학생들이 완벽한 알고리즘을 고안하기를 기대할 수도 없고 해서도 안 된다. 허용적인 분위기 속에서 학생들이 자신의 생각을 나름대로 정리하여 알고리즘을 만드는 것은 수학적인 사고력과 문제해결력에 긍정적인 영향을 끼친다. 그러나 그들이 만든 모든 결과(알고리즘)가 수학적으로 허용될 만한 것이라고 할 수는 없기 때문에, 교사는 학생들이 고안한 알고리즘을 통해서, 학생들의 생각을 이해하고 발문을 통해 수학적으로 타당한 것으로 이끌어야 한다. 그러므로 교사는 학생의 알고리즘을 발전적으로 안내하기 위해 그것을 판단할 수 있는 수학적인 기준을 명확히 할 필요가 있다. 여기서는 Campbell등(1998)이 제시한 3가지 기준을 이용하여 '훌륭한 알고리즘이란 무엇인가'에 대해 이야기하고자 한다 : (1) 효율적인가? ; (2) 수학적으로 타당한가? ; (3) 일반화가 가능한가?

(1) 효율적인가?

효율적인 알고리즘이란 학생이 고안한 알고리즘을 자신이 수행했을 때 답에 도달하기가 불가능하거나 아주 복잡한 것은 지양되어야 한다는 뜻이다. 이 기준은 적용하기가 비교적 쉽다. 다음과 같은 문제가 주어졌다고 가정하자.

'복도를 쓸고 닦는 데 걸리는 시간은, 영은이 혼자는 10분이 걸리고, 철우는 20분, 기영이는 30분이 걸린다고 한다. 영은이, 철우, 기영이가 모두 함께 복도를 쓸고 닦으면 시간이 얼마나 걸릴까?'

어떤 학생이 이 문제를 해결하는 데 다음과 같은 알고리즘을 고안했다.

	영은이가 한 일	철우가 한 일	기영이가 한 일	세 명이 한 일
1분 후	$\frac{1}{10}$	$\frac{1}{20}$	$\frac{1}{30}$	$\frac{1}{10}+\frac{1}{20}+\frac{1}{30}$
2분 후	$\frac{2}{10}$	$\frac{2}{20}$	$\frac{2}{30}$	$\frac{2}{10}+\frac{2}{20}+\frac{2}{30}$
·	·	·	·	·
·	·	·	·	·
·	·	·	·	·

이 학생은 한 사람이 1분 동안 한 일의 양을 구해내기는 했지만, 그 양과 시간의 관계를 일반화시키지 못하고 있다. 이 학생이 고안한 알고리즘은 논리적인 오류는 없지만, 실제로 실행하여 정답에 도달하기는 거의 불가능하므로 효율적이지 못하다.

(2) 수학적으로 타당한가?

수학적인 타당성은 판단하기가 간단하지 않다. 수학적인 타당성이란 과정을 나타낸 알고리즘에 오

류가 있는가 없는가를 살펴보는 것으로 수학적 논리성과 유사하다. 예를 들어, 어떤 학생이 156÷4를 6÷4를 먼저 계산하여 $1\frac{1}{2}$, 두 번째 50÷4가 $12\frac{1}{2}$, 마지막으로 100÷4가 25므로 몫을 모두 더하면, $1\frac{1}{2}+12\frac{1}{2}+25=39$라는 답을 얻었다. 꽤 까다로운 방법이지만, 각각의 몫이 자리값에 바탕을 두고 있기 때문에 수학적으로 타당하다.

(3) 일반화 가능한가?

알고리즘은 문제해결의 과정에서 생겨난다. 이 알고리즘은 모든 비슷한 문제에 적용가능한가를 나타내는 것이 일반화이다. 문제 A의 해결과정에서 만들어진 알고리즘이 문제 A′에도 적용가능한가를 뜻한다.

'가로의 길이가 4cm, 세로의 길이가 4cm, 높이가 6cm인 직육면체의 겉넓이를 구하시오.'라는 문제가 있다고 생각해 보자. 학생 A는 $\{(4\times6)\times4\}+\{(4\times4)\times2\}=128(cm^2)$라는 답을 얻었고, 학생 B는 $\{(4\times6)\times2\}+\{(4\times6)\times2\}+\{(4\times4)\times2\}=128(cm^2)$라는 답을 얻었다. 두 학생이 얻은 결과는 같지만, 과정에서는 차이가 난다. 학생 A는 주어진 직육면체의 옆면이 모두 합동이라는 점에 주목하여 문제를 해결하였고, 학생 B는 직육면체의 마주 보는 면이 합동이라는 점을 고려하여 문제를 풀었다. 학생 A가 고안한 알고리즘이 학생 B의 것보다 단순, 간결하지만 우연히 이 문제에서 가로와 세로의 길이가 같기 때문에 가능한 것이었다. 만약, 가로의 길이가 3cm, 세로의 길이가 4cm, 높이가 5cm인 직육면체였다면 학생 A의 알고리즘은 더 이상 유용하지 않다. 따라서, 학생 B의 알고리즘은 일반화가 가능하지만 학생 A의 것은 그렇지 못하다. 또, 간단한 알고리즘이라고 해서 반드시 더 좋다고 할 수 없으며 때로는 길이가 긴 알고리즘이 더 나을 수도 있다.

Ⅷ. 맺는 말

얼핏 "알고리즘의 학습"이라 하면, 경직되고 기계적인 학습을 생각하기가 쉽다. 그러나 알고리즘은 문제 해결의 처리절차를 만들어 내는 창의적인 학습이며, 학생 중심의 학습이다. 특히, 컴퓨터와 계산기가 주도하는 정보사회에서의 다양한 상황으로부터 스스로 문제를 알고리즘적인 방법으로 해결하는 것은 중요하다. 또, 알고리즘은 강력하고 신뢰로우며, 신속 · 정확하다. 문자기록으로 남길 수 있을 뿐 아니라, 인지적인 상을 가능하게 한다. 한편으로 학습의 대상이 되기도 한다. 알고리즘을 학생들이 스스로 고안한다는 것은 저절로 되는 것이 아니라, 교사의 역할이 중요하다. 교사는 학생들이 알고리즘을 고안할 수 있는 충분한 시간을 제공하며, 학생들의 사고를 뒷받침해 줄 수 있는 구체적인 조작을 많이 해야 한다. 아울러 적용 가능한 전략을 지도하며 의미있는 상황에서 학생들이 알고리즘을 고안하도록 하는 것이 좋다. 그렇게 고안된 알고리즘을 다른 친구들과 공유할 수 있도록 해야 한다. 이렇듯 알고리즘의 학습은 추론, 문제해결, 의사소통과 통합적으로 이루어지는 것이 바람직하다. 학생이 스스로 고안한 알고리즘은 학생 자신에게 자신감을 심어주어 수학에 대한 긍정적인 태도 형

성에 도움을 준다. 다만 초등수학에서 산술 알고리즘이 큰 부분을 차지하고 많이 다루지 않기 때문에 이 글에서 사례로 든 것이 대부분 산술 알고리즘이라는 제한이 따른다.

앞으로 초등수학에서 대수 미적분 알고리즘과 그림 알고리즘의 학습에 대한 연구와 지도방법에 대한 연구가 필요할 것 같다. 마지막으로 수학 교육과정의 내용 모두를 학생이 고안한 알고리즘으로 지도하는 무리가 있을 것이다. 교사는 학생이 고안한 알고리즘과 표준 알고리즘이 적절히 균형을 이루도록 지도해야 할 것이라고 생각된다.

참 고 문 헌

구광조 · 오병승 · 류희찬 공역 (1998). 수학교육과정과 평가의 새로운 방향, 서울: 경문사

Barnett (1998). A Brief History of Algorithms in Mathematics. In L.J. Morrow & M.J. Kenny (Eds.) (1998). *The Teaching and Learning of Algorithms in school Mathematics : 1998 Yearbook*, pp.69-77, Reston, VA: The National Council of Teachers Mathematics, Inc.

Campbell, Rowan & Suarez (1998) What Criteria for Student-Invented Algorithms? In L.J. Morrow & M.J. Kenny (Eds.) (1998). *The Teaching and Learning of Algorithms in school Mathematics : 1998 Yearbook*, pp.49-55, Reston, VA: The National Council of Teachers Mathematics, Inc.

Carroll, W. M. & Porter, D. (1997 March) *Invented Strategies Can Develop Meaningful Mathematical Procedures*, Teaching Children Mathematics 3, pp.370-374

Frances R. Curcio & Sydney L. Schwartz (1998), There Are No Algorithms for teaching Algorithms : *Teaching Math*, 5 Sep

Hughes (1998). Understanding Algorithms from Their History. In L. J. Morrow & M. J. Kenny (Eds.) (1998). *The Teaching and Learning of Algorithms in school Mathematics : 1998 Yearbook*, pp.78-80, Reston, VA: The National Council of Teachers Mathematics, Inc.

Kilpatrick, (1987). What Constructivism Might Be in Mathematics Education. *In Proceedings of the Eleventh International Conference on the Psychology of Mathematics Education, 3-27.* Montreal, Quebee.

Maurer (1998). What Is an Algorithm? What is an Answer? In L.J. Morrow & M.J. Kenny (Eds.) (1998). *The Teaching and Learning of Algorithms in school Mathematics : 1998 Yearbook*, pp.21-31, Reston, VA: The National Council of Teachers Mathematics, Inc.

Usiskin (1998). Paper-and-Pencil Algorithms in a Calculator-and-Computer Age, In L. J. Morrow & M.J. Kenny (Eds.) (1998). *The Teaching and Learning of Algorithms in school Mathematics : 1998 Yearbook*, pp.7-20, Reston, VA: The National Council of Teachers Mathematics, Inc.