韓國數學教育學會誌 시리즈 E
<數學敎育 論文集>
제 13집, 2002. 2. 19-38.

J. Korea Soc. Math. Ed. Ser. E:
Communications of Mathematical Education
Vol. 13, Feb. 2002. 19-38.

연산능력을 기르기 위한 대안적 알고리즘 지도 방안
—사칙연산을 중심으로 —

남 승 인 (대구교육대학교)
강 영 란 (경북하양초등학교)
박 인 묵 (경북금락초등학교)

알고리즘이란 '유한한 단계를 거쳐 일련의 문제를 해결하기 위한 명확하고 체계적인 방법'으로써 수량에 관련된 문제를 보다 신속·정확하게 처리하기 위하여 역사적으로 다양한 알고리즘이 존재·변천해 왔다. 계산기가 발명되기 전까지는 지필 알고리즘이 매우 강조되어 왔으나 계산기가 상용화되면서 지필 알고리즘에 대한 효용성과 활용도가 점차 줄어들고 있으나 지필 알고리즘은 수학학습의 기초·기본인 동시에 뼈대로써 그 가치와 역할은 여전히 중요하다. 그러나 표준화된 지필 알고리즘에 대한 지나친 강조로 인해 학생들은 대수적 구조나 계산 원리를 바르게 이해하지 못한 채 반복 연습을 통해 익힌 표준 알고리즘을 기계적으로 적용하여 답을 구하는 경우가 많으며, 이로 인해 학생들은 수학학습에 대한 불안감과 기피현상이 보이고 있다. 또 인간의 창조적 사고활동의 최종적인 산물인 표준 알고리즘은 대안적인 알고리즘에 비해 효율성에서 앞서지만 학생들의 사고 수준에서는 그 원리를 이해하기 힘든 경우가 있을 것이다. 따라서 수학교육의 목적 중의 하나인 문제 해결력을 기르기 위해, 그리고 표준 알고리즘의 가치와 효율성을 인식시키고, 수학학습에 대한 불안감을 줄이기 위해 표준 알고리즘뿐만 아니라 대안적인 알고리즘을 병행하여 지도할 필요가 있다.

1. 서 론

인류 역사와 더불어 발생한 수학에서 가장 먼저 다루기 시작한 영역은 세고 셈하는데 관련된 활동이며, 사람들이 실세계의 물리적 대상이나 현상을 접촉하면서 가장 먼저 이루어지는 활동 역시 인식한 대상에 대해 세고 셈하는 활동, 즉 수와 연산과 관련된 활동일 것이다. 또한 어린이들이 의사소통을 하기 위해 언어와 문자를 배우는 것과 함께 유아·유치원 시절부터 수를 세고, 숫자를 쓰고, 셈하는 것(알고리즘)을 거의 동시에 배운다. 그리고 수학학습의 입문기인 초등학교 1학년에서 처음 경험하는 수학학습의 대상이 수와 연산이다. 이와 같이 수와 연산은 실생활에서 양을 표현하고 그들 사이의 관계를 다루는 도구인 동시에 수단이며, 수학학습의 기초·기본인 동시에 수학의 뼈대로써 수학의 다른 영역과 타교과 학습을 위한 필수적인 도구이다.

알고리즘(algorithm)이란 유한한 단계를 거쳐 일련의 문제를 해결하기 위한 명확하고 체계적인 방법(Maurer, 1998)이라고 볼 때, 알고리즘은 유일한 방법만 존재한다고 볼 수 없다. 이는 문제를 해결하는 방법이나 전략이 다양한 것과 마찬가지이다. Morrow(1998)는 학생 스스로 알고리즘을 발견해

낼 수 있는 기회를 제공해야 한다는 학자가 있는 반면 학생들은 표준 알고리즘을 배워야 한다고 주장하는 수학 교육자들도 있음을 지적하면서, 표준 알고리즘이 가장 효과적이지만 요즈음의 사회는 과거에 비해 지필 계산의 가치나 효용성이 뒤떨어지며 알고리즘이 단순한 발견이 아니라 학생들의 창조적인 산물임을 보여주기 위해 그리고 알고리즘은 고정된 것이 아니라 인간의 창조적인 노력에 의해 변화·발전됨으로 "학생이 만들어낸 알고리즘과 표준 알고리즘 사이에 조화를 이루어야 한다." 고 주장하고 있다.

20세기 초 계산기가 발명되기 전까지는 정확·신속한 계산을 하기 위하여 지필 알고리즘이 매우 강조되어 왔다. 그러나 계산의 신속성이나 정확성에서 뛰어난 성능을 가진 계산기가 발명됨에 따라 지필 알고리즘에 대한 효용성과 활용도가 점차 줄어들고 있으며, 이는 학교 수학에도 커다란 영향을 미치고 있다. 앞으로의 알고리즘 지도는 계산의 도구로써 한정지울 것이 아니라 알고리즘 학습을 통해서 논리적·창의적인 사고력과 학습 집중력을 기르는 방향으로 변해야 할 것이다. 이에 따라 알고리즘 학습의 가치나 중요성에 대해 교사들은 자신의 신념을 되짚어 보아야 할 것이다. 즉 과거에 그들이 배웠던 알고리즘이나 교과서에 제시된 표준 알고리즘 지도에 국한·집착할 것이 아니라 역사적으로 사용되었던 전통적인 알고리즘이나 학생들이 만들어 낸 비형식적인 알고리즘 지도를 통해 알고리즘의 역할 및 변천 과정을 살펴보는 학습 기회와 환경을 제공할 필요가 있다. 이러한 기회를 통해 학생들은 교과서에 제시된 표준 알고리즘 학습의 필요성과 가치를 인식할 것이며, 자신의 창의력을 발휘하여 알고리즘을 분석·고안해 봄으로써 논리적·창의적인 사고력과 추론능력을 기를 수 있을 것이다.

고대에서 현재에 이르기까지 수학에 대한 선각자들에 의해 창안·활용된 수 표현 체계와 계산 알고리즘은 매우 다양하며 인류 문명과 함께 발전해 왔다. 알고리즘은 특별한 상황에서 답에 이르는 단순한 절차이기 때문에 모든 사람이 똑같은 알고리즘을 사용해야 한다는 주장을 뒷받침할 근거는 없다(Morrow, 1998). 계산의 도구로서 계산기가 일반화·보편화되기 이전까지 알고리즘 지도의 중요한 목표가 주어진 문제를 정확하고 신속하게 해결하는 데 있다고 볼 때, 새로운 알고리즘을 창안하고 지도하는 주된 목표는 계산의 정확성과 신속성을 증진시키는데 있었을 것이다. 이렇게 볼 때, 교과서에 제시된 표준 알고리즘은 이전에 활용되었던 전통적인 알고리즘들보다 정확성과 신속성에서 가장 앞섰다고 볼 수 있으며 이에 근거하여 학교 수학에서 주된 지도 내용에 포함시키고 있으므로 표준 알고리즘을 지도하는 것이 바람직하다고 보겠다. 그러나 표준 알고리즘은 인간의 고차적인 사고활동에 의해 가장 세련되게 다듬어진 최종의 산물이라고 볼 때, 학생들의 사고 수준에서는 그 원리를 이해하기 힘든 경우가 있을 것이다. 이는 결국 문제 해결의 도구로 표준 알고리즘을 활용하지 못하거나 잘못 활용함으로 해서 문제 해결을 실패할 가능성이 매우 높다. 따라서 수학교육의 궁극적인 목적 중의 하나인 문제 해결력을 기르기 위해서는 표준 알고리즘을 포함하여 이전에 사용되었던 다양한 유형의 전통적인 알고리즘 및 학생 스스로 개발한 비형식적인 알고리즘, 즉 대안적인 알고리즘(Carroll & Porter, 1998)을 병행하여 지도할 필요가 있다.

Kamii와 Dominick(1998)는 표준 알고리즘에 대한 지나친 강조는 많은 학생들에게 수학 학습에 대한 가치나 학습의 의미를 제대로 전달하지 못할 수 있으며 논리적인 사고를 방해할 수 있다고 주장하고 있으며, NCTM(2000)에서도 지필 계산 및 표준 알고리즘에 대한 지나친 의존을 경고하면서 교사는 학생들에게 정답을 찾아내는데 사용할 수 있는 여러 가지 유형의 대안적인 알고리즘을 소개하고 이를 활용해 보도록 하는 학습 경험은 표준 알고리즘 지도의 효과를 높일 수 있으며, 특히 저학년의 경우 여러 가지 유형의 알고리즘의 특성에 대한 비교·분석을 통하여 가장 간결하고 효율적인 알고리즘을 선택·활용할 수 있는 기회와 환경을 제공함으로써 학생들은 의미있는 수학을 위한 토대를 만들 수 있다고 주장하고 있다.

또 강시중(1993)은 표준 알고리즘은 본래 주어진 것이 아니고 인간의 창조적인 노력에 의하여 만들어진 형식 또는 규칙이므로 계산 지도에 있어서는 형식뿐만 아니라 형식을 낳게 하는 비형식적 장면으로부터 알고리즘의 형식을 학습할 것을 권고하고 있으며, 이러한 여러 주장을 종합해 볼 때, 계산 알고리즘 지도는 다양한 유형의 대안적인 알고리즘에 대한 소개와 이를 활용해 보도록 하는 학습 경험, 그리고 다양한 유형의 알고리즘의 특성에 대한 비교와 분석을 통하여 가장 간결하고 효율적인 알고리즘을 선택·활용할 수 있는 기회와 환경을 제공해야 할 것이다. 그러나 아직도 대부분의 학부모와 교사들은 교과서에 제시된 표준 알고리즘을 지나치게 강요하고 있으며, 이를 능숙하게 사용하도록 하기 위해 지필을 이용한 반복 연습을 강조함으로 해서 학생들은 의미없는 수학학습을 하고 있는 실정이며, 이로 인해 학생들은 수학학습에 대한 흥미를 잃거나 불안감 내지는 수학에 대한 기피 현상을 보이고 있다. 이러한 현상을 줄이기 위해서는 우선 교사부터 연산에 있어서 다양한 유형의 전통적·대안적인 알고리즘에 대한 풍부한 지식이 있어야 한다.

본고에서는 표준 알고리즘에 비해 속도나 정확성에서는 다소 뒤떨어질지라도 논리적으로 모순이 없는 비형식적이거나 역사적으로 이전에 사용했던 전통적인 알고리즘, 즉 대안적인 알고리즘 지도의 필요성과 대안적인 알고리즘의 교육적 효과, 그리고 사칙연산에서 사용될 수 있는 대안적 알고리즘의 예들을 소개하고 대안적인 알고리즘을 효과적으로 지도하기 위해 교사들이 유의해야 할 점에 대해서 살펴보고자 한다.

2. 알고리즘의 정의

알고리즘이란 말의 기원은 Kitab al-jam Wal tafrig bi hisab al-Hind(인도 사람들이 방법에 따른 덧셈과 뺄셈)라는 저서에서 십진기수법으로 수를 표기하고 연산 방법을 제시하며 방정식 풀이에서 이항하는 것을 'al-jabr', 동류항을 간단히 정리하는 것을 'almugabala'라고 정의한 al-khwarizmi라는 수학자의 이름에서 유래되었다(Barnett, 1998). 알고리즘에 대해 Barnett(1998)는 알고리즘을 주어진 문제에 대한 절차적인 과정이 있어 그 과정을 정확하게 실행했을 때 정확한 답을 얻을 수 있게 하는 절차로 정의하였고, Maurer(1998)는 알고리즘을 '일련의 문제를 해결할 수 있는 분명하고 체계적인 절차'로서 먼저 '투입하고, 확정된 일련의 절차에 따라 유한회의 절차를 수행하여 결정적인 해를 유

도한다.'는 것으로 정의하고 있다. 그러나 Barnett와 Maurer이 알고리즘을 해석하는 입장은 다소 차이가 있다. Barnett는 전통적인 의미에서 알고리즘을 해석하여 주어진 절차를 정확히 실행하기만 하면 주어진 문제에 대한 옳은 답을 보장한다고 설명하고 있으나, Maurer은 현대적인 의미에서 옳은 답인지의 여부에 초점을 두기보다는 하나의 문제를 해결할 수 있는 다양한 알고리즘 중 가장 적합한 알고리즘을 선택하는 것에 더 초점을 두고 있다. 이처럼 알고리즘을 전통적인 의미로 해석하건 현대적인 의미로 해석하건 알고리즘을 문제를 해결하기 위한 일련의 과정 또는 방법이라는 점에서는 공통적이다.

학교수학에서 알고리즘을 가르치고 배울 때 주로 지필 알고리즘을 이용하고 있으나 알고리즘은 문제에 대한 해를 얻기 위한 절차이므로 현대와 같은 정보화 시대에는 지필 알고리즘뿐만 아니라 계산기와 컴퓨터를 이용한 알고리즘도 알고리즘을 학습하기 위한 하나의 도구로써 한 부분을 차지해야 한다. Usiskin(1998)도 알고리즘의 종류로 지필 알고리즘과 계산기 · 컴퓨터 알고리즘으로 분류하고 있다.

이상을 정리해보면 산 정상(답)을 오르는 길(과정)이 여러 가지 있듯이 답에 이르는 과정의 다양성을 인식하고 교사는 한 가지 알고리즘만을 고집하여 가르치기보다는 다양한 절차를 소개해주고 학생들이 자기 나름으로 해결한 과정에 대해서 인정을 해 주며 이 때 사용하는 학습 도구도 지필 알고리즘에만 국한할 것이 아니라 학습 내용에 따라 적절한 것을 선택하여 제공해야 한다.

3. 알고리즘의 교육적 가치

수백년동안 사용해 온 절차인 알고리즘은 학생들의 수학적 성취의 필수적인 요소로 보는 것이 주된 관점이었고 최근에도 계산 지도는 학교수학에서 주된 관심거리이다. 이와 같이 여전히 알고리즘이 수학교과의 중심이 되는 것은 알고리즘은 다음과 같은 교육적인 가치를 지니고 있기 때문이다 (Usiskin, 1998).

· 알고리즘은 효과적하다. : 알고리즘은 문제해결 과정의 일반화를 의미하므로 주어진 문제에 대해 해답을 보증하는 알고리즘을 알고 있다면, 그와 유사한 종류의 모든 과제를 해결할 수 있다.

· 알고리즘은 신뢰할 수 있다. : 어떤 알고리즘이 정확하다면 그 알고리즘을 몇 번이고 반복해도 정확한 답을 얻을 수 있다.

· 알고리즘은 정확하다. : 주어진 문제를 해결하기 위해 알고리즘을 적용했을 때 알고리즘이 정확하다면 정확한 답을 얻을 수 있다. 즉 알고리즘의 정확성은 답의 정확성과 관련이 있다.

· 알고리즘은 신속하다. : 잘 고안된 알고리즘은 답에 이르는 길을 직접적으로 제시해주므로 시간과 노력을 절약해 준다.

· 알고리즘은 지필 기록을 제공한다. : 답을 얻기 위해 남겨둔 과정의 기록은 학생들이 자신의 활동 과정을 반성하고 세련되게 하며 서로의 알고리즘을 공유하는데 유용하다.

· 알고리즘은 정신적 표상을 증진한다. : 지필로 기록된 알고리즘을 통해서, 연필과 종이없이 결과를 얻을 수 있다.

· 알고리즘은 유익하다. : 문제에서 주어진 정보와 답 사이의 관계에 대한 통찰을 제공한다.

· 알고리즘은 다른 알고리즘에 적용할 수 있다. : 하나의 상황에서 사용된 알고리즘이 다른 상황에서 그 알고리즘의 부분으로 사용된다.

· 알고리즘은 학습의 주제가 될 수 있다. : 알고리즘은 문제의 해에 이르는 과정일 수도 있지만 한편으로 학습의 대상이 된다.

이처럼 알고리즘이 많은 문제를 정확하고 신속하게 해결할 수 있도록 돕고 정신적 표상을 증진시킨다면 학생들은 그 알고리즘을 배워야 한다. 그러나 이 때 그 절차가 외부에서 주어지기보다는 학생이 절차를 만드는 과정에 직접 참여할 수 있어야 한다. 현재 우리의 수학 교육은 알고리즘의 교육적 가치가 지나치게 강조되어 학생들은 알고리즘을 기억하고 정확하게 문제를 해결하기 위해 알고리즘을 반복·연습하는 데 많은 시간을 보내고 있다. 결국 학생이 알고리즘을 익숙하게 사용하기 위해 많은 연습을 하는 동안 기계적인 절차에만 몰두하여 우리가 수학을 통해 얻고자 하는 수감각이나 추론 등은 놓치고 있다.

NCTM(1998)에서는 지필 계산에 대한 지나친 의존을 경고하면서 문제 해결, 수학적 개념, 수감각을 강조하고 알고리즘 학습이 학생들이 의미있는 활동을 통하여 이루어질 것을 권고하고 있다. 즉 알고리즘의 지도는 그 필요성을 제기하는 문제 상황으로부터 시작하여 일방적인 암기와 적용이 아닌 문제상황으로부터의 필요성을 통한 알고리즘 학습을 강조하며 일상 생활에서 알고리즘의 유용성을 이해할 수 있도록 지도해야 한다. 이처럼 교과서에 제시된 유형과 같이 인간의 창조적이고 끊임없는 노력에 의하여 가장 정확하고 신속함에 초점을 둔 표준 알고리즘에 대비하여 Carroll & Porter(1998)은 수학사적으로 과거에 사용했거나 학생들이 생활 경험 및 기존의 지식과 조작활동을 통하여 학생 스스로 고안·활용하는 비형식적인 알고리즘으로써 논리적인 모순이 없는 알고리즘을 대안적 알고리즘이라 하였다.

4. 알고리즘 강조에 따른 위험 요소

알고리즘이란 주어진 문제에 대한 절차적인 과정이 있어 그 과정을 정확하게 실행했을 때 정확한 답을 얻을 수 있게 하는 절차(Barnett, 1998)라고 할 때, 앞서 알고리즘의 교육적인 가치에서 언급한 바와 같이 특정한 문제를 정확하고 신속하게 해결할 수 있는 가장 효과적인 수단이다. 그러나 Usiskin(1998)은 지필 알고리즘을 지나치게 강조할 경우, 다음과 같은 몇 가지 위험성을 내포하고 있은 지적하고 있다. 따라서 알고리즘에도 과정에서 교사들은 각별히 유념할 필요가 있을 것이다.

1. 지필 알고리즘을 수행한 결과를 맹목적인 수용할 가능성이 있다.

대부분의 학생들은 문제를 풀어서 답을 구했을 때, 그 답의 정당성에 대해서 검토해 보는 과정을 매우 소홀히 하거나 아예 검토할 필요성을 느끼지 않는 경향이 있다. 이는 문제 해결 과정에서 중요한 일부분을 생략하는 것과 마찬가지이다. 특히 알고리즘은 믿을만하기 때문에 알고리즘이 올바르게 수행되었다면 답을 맹목적으로 받아들이는 경우가 종종 있다. 문제 해결 과정에 대한 반성과 검토는

해결 과정에서 발생할 수 있는 오류를 발견하는데 도움을 줄뿐만 아니라 구한 답의 정당성을 뒷받침해 준다. 문제 해결과정에 대한 반성은 바른 답을 확인하는 것도 중요하지만 문제 해결 과정에서 작용한 사고를 확장시켜 주는 기회를 갖게 하는 것이 필요하다.

2. 지필 알고리즘의 지나친 적용은 효과적인 문제 해결 접근 방법을 방해할 수 있다.

다음 예에서 보는 바와 같이 학생들은 암산으로 해결할 수 있는 과제에 대해서도 지필 알고리즘을 지나치게 의존할 경우 지필 알고리즘을 적용하는 것 그 자체는 잘못된 것은 아니지만 현명하지 못한 생각이며 비효율적이다. 또한 암산을 지나치게 적용할 경우 지필이나 계산기의 남용만큼이나 위험한 일이다. 암산으로 계산을 할 경우는 투입에 대한 기록을 할 수 없기 때문에 만약 오류가 발생한다면 알고리즘 수행과정상의 오류인지 아니면 투입상의 오류인지 말할 수 없는 경우가 종종 있다.

A.

$$
\begin{array}{r}
345 \\
\times\ 10000 \\
\hline
000 \\
000 \\
000 \\
000 \\
345 \\
\hline
3450000
\end{array}
$$

B.

$$
\begin{array}{r}
18.5 \\
2\overline{)37.0} \\
2 \\
\hline
17 \\
16 \\
\hline
10 \\
10 \\
\hline
0
\end{array}
$$

C. $8\times\frac{3}{4} = \frac{8}{1}\times\frac{3}{4} = \frac{2}{1}\times\frac{3}{1} = \frac{6}{1}$

(학생들은 답이 6임을 깨닫지 못할지도 모름)

D.

$$
\begin{aligned}
4x &= 32 \\
\frac{x}{4} &= \frac{32}{4} \\
x &= 8
\end{aligned}
$$

3. 지필 알고리즘은 수학에 관련된 지적 능력을 단련시킨다고 과신할 수 있다.

어떤 사람들은 지필 알고리즘이 순서와 주의를 가르쳐주는 절차를 경험하는 과정에서 지적 능력을 강화시킨다 고 주장한다. 그러나 이러한 주장을 지지해주는 어떤 증거도 아직까지는 없다. 알고리듬은 정신적 표상을 제공하는 데 유익하지만 정신적 표상은 특별한 과제에 의도적으로 적용되며, 또 이러한 표상들이 공간지각이나 문제해결과 같은 보다 고차적인 능력으로 일반화시켜준다는 증거는 없다. 오히려 어려운 알고리즘은 고차적인 능력으로 일반화시켜주기보다는 어려운 알고리즘은 가르치는 과정에서 보다 중요한 수학적 능력을 억제시키거나 학생들을 수학으로부터 멀어지게 할 수도 있다.

4. 지필 알고리즘은 계산기나 컴퓨터 등 교육공학적 도구의 사용을 억제할 수 있다.

지필 알고리즘과 계산기 또는 컴퓨터 알고리즘을 사용하기 위해서는 일정 수준의 기본적인 소양과 기능을 갖추어야 한다. 지필 알고리즘을 옹호하는 사람들은 지필 알고리즘이 계산기나 컴퓨터 알고리즘보다 더 유용하다고 주장하고 있으나 오늘날은 계산기나 컴퓨터 알고리즘이 지필 알고리즘보다 더 유용할 때가 많으며 계산의 도구로써 이들은 일반화·보편화되어 있다. 특히 정보화의 시대라고 일컫는

요즈음의 사회에서는 다루어야 할 수량관계가 크고 복잡하다. 지필 알고리즘에 대한 지나친 강조는 교통사고를 핑계로 자동차를 이용하지 않는 것과 같은 어리석음을 자행하는 일과 같다. 따라서 계산 도구로써 지필 알고리즘 계산기나 컴퓨터가 갖고 있는 특성을 잘 분석·판단하여 문제 상황에 따라 가장 적절한 도구를 선택·활용하도록 하는 것이 계산 교육의 목적이라고 할 수 있다.

4. 대안적 알고리즘 지도의 필요성

과거의 수학교육은 학생들이 표준 알고리즘을 학습하는 것을 강조했지만 최근에 와서는 수 감각, 추론, 개념적인 이해에 관심을 쏟고 있다. 이에 따라 학생들이 스스로 고안해 보는 의미있는 계산 절차인 대안적인 알고리즘의 중요성이 부각되고 있다. 대안적인 알고리즘 지도의 필요성을 네 가지로 정리한다면 다음과 같다.

① 대안적인 알고리즘은 의미 있는 활동으로써 수학적인 아이디어를 자극한다(Carroll & Porter,1998).

'121-78의 경우, 12에서 1을 빌려와서 11에서 8을 뺀다'와 같은 표준적인 절차를 우리가 학습하는 이유는 정확한 답을 얻을 수 있기 때문이다. 그러나 이러한 알고리즘을 학생이 수동적으로 받아들이고 반복·연습만 한다면 절차에 대한 아무런 의미도 알지 못한 채 활동이 끝이 난다. 오히려 121-78을 78, (+2)80, (+10)90, (+10)100, (+10)110, (+10)120, (+1)121로 더해서 '+2+10+10+10+10+1=43'이라는 값을 얻던지 혹은 수를 분리시켜서 '121-78=(100-78)+21
=22+21=43'의 방법으로 뺄셈을 하는 등 학생 자신이 스스로 문제를 다양한 방법으로 해결하고자 시도하는 동안 단순한 계산 기능과 수 감각이 요구되므로 이러한 과정에서 창의적인 사고력을 신장시킬 수 있다.

② 대안적인 알고리즘은 다양한 방법으로 문제를 해결하기 쉽도록 만들어준다(Carroll & Porter,1998).

표준 알고리즘의 최대 강점이 문제에 적용하면 답을 쉽고 빠르게 얻어낼 수 있다는 점이다. 그러나 표준 알고리즘이 기억나지 않을 때는 해결해보려는 노력도 없이 포기해버리는 경우가 많다. 이를 극복하기 위해서는 학생들이 저학년 때부터 한가지 알고리즘을 고집하여 지도하기보다는 다양한 알고리즘과 전략을 고안하고 탐구할 수 있는 기회를 부여하여 문제 해결 방법을 선택하는데 융통성을 발휘할 수 있도록 해야 한다.

③ 대안적인 알고리즘은 학생들의 내적인 특성에 부합된다(Carroll & Porter,1998).

교과서의 표준 알고리즘은 오른쪽과 같이 오른쪽에서 왼쪽으로 진행하는 미산법(未算法)하고 강조하고 있으나 Campbell & Rowan & Suarez(1998)등의 연구 결과들을 보면 덧셈과 뺄셈 문제에서 학생들은 왼쪽에서 오른쪽으로 계산하는 두산법(頭算法)을 더 많이 이용하는 경향이 있다고 주장하고 있다. 따라서 교과서에서 요구하는 표준 알고리즘은 학생의 방법과 충돌할 수 밖에 없어 결국 자신

의 사고를 포기하고 이해 없이 알고리즘을 암기해야 한다. 결국 이는 학생들이 점점 수학을 어려워하고 싫어하는 등 수학에 대해 부정적인 태도만 길러준다. 따라서 학생들의 이해를 바탕으로 한 알고리즘 수업이 되기 위해서는 학생들의 내적인 특성에 맞는 알고리즘으로 학습할 수 있는 환경을 제공해야 한다. 즉 그들이 스스로 알고리즘을 고안·이용하고 다양한 전통적인 알고리즘과 표준 알고리즘을 비교해보고 그 차이를 토론해 보는 과정에서 알고리즘의 가치와 가장 효율적인 알고리즘을 선택·활용할 수 있는 능력과 자신감을 갖게 할 것이다.

$$\begin{array}{r} 3\,6\,5 \\ +\,5\,7\,4 \\ \hline 8\,0\,0 \\ 1\,3\,0 \\ 9 \\ \hline 9\,3\,9 \end{array} \qquad \begin{array}{r} 3\,6\,5 \\ +\,5\,7\,4 \\ \hline 9 \\ 1\,3\,0 \\ 8\,0\,0 \\ \hline 9\,3\,9 \end{array}$$

(두산법) (미산법)

④ 대안적인 알고리즘은 문제해결을 위한 기본적 소양과 능력을 갖게 한다.

대안적인 알고리즘은 학생들의 내적인 특성에 부합하며 수학적인 아이디어를 자극하므로 수학적으로는 가치있는 도구이다. 그러나 지나치게 표준 알고리즘만 강요할 경우 표준 알고리즘의 원리를 이용하여 문제를 해결하기 어려운 학생은 문제해결을 포기할 가능성이 있다. 따라서 그들에게 효율성면에서는 다소 뒤떨어진다고 하더라도 대안적인 알고리즘을 통하여 당면한 문제를 해결할 수 있는 능력을 길러줌으로써 후속하는 학습을 가능하게 해 줄 것이다.

5. 대안적 알고리즘 지도의 효과

Carroll과 Porter(1998)에 따르면 수·연산에 대한 학생의 이해를 향상시키는 방법 중의 하나가 학생들이 의미 또는 계산 절차를 스스로 고안해내는 것이고, 이러한 활동은 학생들에게 다음과 같은 효과를 준다.

① 연산 규칙이나 절차의 의미를 이해할 수 있다.

대부분 계산으로 구성된 교육과정을 사용하여 학생들이 계산 규칙을 교과서에 제시된 순서대로 이해되든 안 되든 교육 받는 경험을 통하여 학생은 교사의 설명에 의존하고, 교사의 지시에 따라 맹목적으로 이해하며 연습을 하는 동안 아주 자연스럽게 수동적인 학생으로 변하게 된다. 그러나 학생들이 자기 나름의 알고리즘을 만들어가는 기회를 갖게 된다면 계산 규칙이나 계산 절차가 왜 필요하며, 어떤 과정을 통하여 그렇게 만들어졌는지, 또 우리 생활에서 어떻게 이용되는지를 스스로 이해하게 된다.

② 정확한 연산 결과를 얻을 수 있는 길이 여러 가지 있음을 알게 된다.

수학은 필요에 따라 만들어지고 이것이 일반화되어 공식이나 절차가 된 것이므로 문제를 해결하는 방법이 오직 한가지 방법만이 있는 것이 아니다. 자신의 필요에 따라 새로운 규칙이나 절차를 만들어 낼 수 있다는 생각을 가지고 여러 가지를 시도할 수 있다. 예를 들면 '121-78의 경우, 12에서 1을 빌려와서 11에서 8을 뺀다.'와 같은 표준 알고리즘의 절차를 수를 분리시켜 121-78 = (100-78)+21 =

22+21=43의 방법으로 계산하거나 수직선을 그려 두 수 사이의 거리를 통해 43이라는 값을 얻을 수도 있다. 이와 같이 문제를 해결할 수 있는 방법이 여러 가지 있다는 것을 깨달을 때 수학에 대해 생각하는 방법도 바뀌게 되어 고착화된 사고에서 벗어나 유연하게 사고를 할 수 있다.

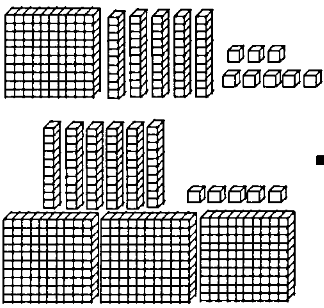
③ 수감각과 연산감각을 향상시킬 수 있다.

문제가 주어지면 해결하기 위한 과정이 복잡한지 혹은 단순한지에 따라 암산을 하거나 어림을 통해 대략적인 답을 추측해보는 등 문제를 해결하기 위한 가장 적합한 알고리즘을 찾는 동안 수 감각과 연산 감각은 길러진다. Sowder(1992)와 많은 연구자들이 수업 현장을 관찰을 하고 난 결과 보고서에 따르면 수업 현장에서 학생들이 문제를 해결할 때 정답을 찾는 데만 집중하지 않고 가장 적합한 방법은 무엇인지 그러한 과정을 통해 얻은 결과는 어느 정도 합리적인지에 더 초점을 두는 학생이 추론과 의사소통 및 수감각과 연산감각이 더 발달한다고 언급하고 있다.

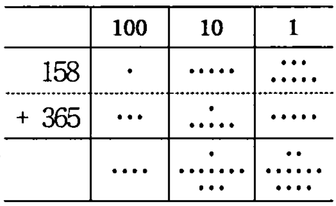
6. 대안적인 알고리즘의 예

(1) 표준 알고리즘 지도 절차

① 구체물을 이용한 계산



② 수도(數圖)를 이용한 계산



③ 중간알고리즘 1

```
   1  5  8
 + 3  6  5
 ---------
      1  3   ←일의 자리의 합
   1  1  0   ←십의 자리의 합
   4  0  0   ←백의 자리의 합
 ---------
   5  2  3
```

④ 중간알고리즘 2

```
   1  5  8
 + 3  6  5
 ---------
      1  3
   1  1
   4
 ---------
   5  2  3
```

⑤ 표준알고리즘 1

```
   1  1
   1  5  8
 + 3  6  5
 ---------
   5  2  3
```

⑥ 표준알고리즘 2

```
   1  5  8
 + 3  6  5
 ---------
   5  2  3
```

표준 알고리즘의 지도는 위 그림과 같이 구체물 조작단계부터 받아올림한 수를 기억하여 처리하는 단계까지 즉 ①~⑥단계를 거치면서 좀더 간편하고 단순화시키면서 지도하는 것이 일반적인 방법이다. 그러나 대안적인 알고리즘은 이와는 달리 여러 가지 방법이 있다.

(2) 대안 알고리즘의 예

알고리즘이 문제 상황에서 답에 이르는 절차에 지나지 않는다는 것을 고려할 때, 하나의 문제를 해결하기 위한 알고리즘이 오직 하나만 존재하는 것은 아니다. 따라서 모든 학생이 똑같은 알고리즘

만을 배운다는 것은 큰 의미가 없다. 지금부터 학생들이 문제를 다양한 관점에서 알고리즘을 탐구할 수 있도록 도움을 주는 자연수 연산에서 역사적으로 흥미있는 전통적인 알고리즘을 소개하고자 한다.

가. 덧셈

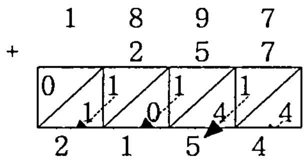
① 부분합을 이용한 알고리즘

덧셈에서 교사와 학생들이 유용하게 사용하고 있는 대안적인 알고리즘 중의 하나가 부분합이다. 이 방법은 부분합의 과정이 진행되는 동안 계산 방향을 (1)과 같이 왼쪽에서 오른쪽으로 하거나, (2)와 같이 오른쪽에서 왼쪽으로 하는 연산 순서에 관계 없이 받아올림이 행해지지 않아 머릿속에 받아올림을 한 값을 기억해 둘 필요가 없기 때문에 표준 알고리즘, 즉 세로셈의 과정을 안내하는 중간 단계의 역할을 하고 있다.

(1)	(2)
1 8 9 7	1 8 9 7
+ 2 5 7	+ 2 5 7
1 0 0 0	1 4
1 0 0 0	1 4 0
1 4 0	1 0 0 0
1 4	1 0 0 0
2 1 5 4	2 1 5 4

② 격자를 이용한 알고리즘

부분합은 표준 알고리즘보다 더 쉽게 이해할 수 있지만 시간과 공간을 더 요구하여 비효율적이라 할 수 있다. 부분합과 같이 받아올림에 대한 개념은 그리 중시하지 않으나 시·공간적으로 효율적인 방법 중의 하나가 바로 덧셈에 대한 격자셈법이라 하겠다. 아래와 같이 먼저 더하고자 하는 두 수를 격자 위에 쓴 후 각 자리마다 더한 값을 자릿값에 맞추어 적은 후 대각선으로 일직선 상에 있는 수들을 더하면 된다.



③ Scratch법을 이용한 알고리즘 Ⅰ : 자리값이 낮은 자리에서 높은 자리의 차례로 계산하기

이 방법은 학생들이 한 자리 수 덧셈을 할 때 오류를 줄일 수 있는 방법으로 합이 10보다 크게 되는 수를 빗금으로 그어 줌으로 받아올림한 값을 기억하지 않아도 된다. 78+56+38을 Scratch 덧셈법으로 계산하면 다음과 같다.

$\begin{array}{r} 7\ 8 \\ 5\ \cancel{6} \\ +\ 3\ 8^{4} \\ \hline \end{array}$	위에서부터 차례로 덧셈을 하되, 8과 6의 합처럼 10이상이 될 때 두 수 중 마지막 수 즉 6에 빗금을 긋고 합한 값의 일의 자리 수 4를 다음에 더해야 하는 숫자 8 옆에 적는다.
$\begin{array}{r} 7\ 8 \\ 5\ \cancel{6} \\ +\ 3\ \cancel{8}^{4} \\ \hline 2 \end{array}$	4와 8의 합에서도 12가 되므로 8에 빗금을 긋고 선 아래에 2를 쓴다.
$\begin{array}{r} {}^{2}\ \ \ \\ 7\ 8 \\ \cancel{5}\ \cancel{6} \\ +\ 3^{4}\ \cancel{8}^{4} \\ \hline 1\ 7\ 2 \end{array}$	일의 자리 합에서 빗금이 2개 있으므로 이 2를 받아올림해 준다. 십의 자리에서도 일의 자리에서의 합과 같은 방법으로 단계를 거쳐 합한다.

④ Scratch법을 이용한 알고리즘 Ⅱ : 자리값이 높은 자리에서 낮은 자리의 차례로 계산하기

이 방법은 구체물 조작이나 암산, 주산에서 이루어지는 절차와 같은 두산법(頭算法)을 이용한 계산 방법으로서 표준 알고리즘을 사용하기 이전에 대안적 알고리즘으로 학생들이 흔히 사용하는 방법이다.

$$\begin{array}{r} 987 \\ +\,356 \\ \hline 12 \end{array} \Rightarrow \begin{array}{r} 987 \\ +\,356 \\ \hline 1\cancel{2}3 \\ 3 \end{array} \Rightarrow \begin{array}{r} 987 \\ +\,356 \\ \hline 1\cancel{2}\cancel{3}3 \\ 34 \end{array} \Rightarrow \begin{array}{r} 987 \\ +\,356 \\ \hline 1343 \end{array}$$

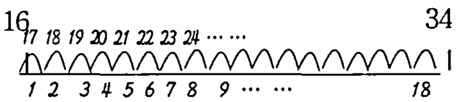
위 그림처럼 자리값이 높은 자리부터 더한다. 그리고 그 아래 자리로 옮기면서 차례로 계산한다.

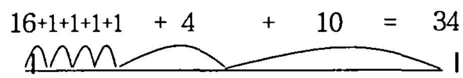
⑤ 전개식을 이용한 알고리즘 : 자리값이 낮은 자리에서 높은 자리의 차례로 계산하기. 이 방법은 표준 알고리즘의 중간 단계에서 이용될 수 있는 미산법(未算法)을 이용한 계산방법이다.

$$\begin{array}{rl} 547 = & 5(100) + 4(10) + 7(1) \\ +\ 296 = & 2(100) + 9(10) + 6(1) \\ \hline & 7(100) + 13(10) + 13(1) \\ & 7(100) + 14(10) + 3(1) \\ & 8(100) + 3(10) + 3(1) \\ \hline & = 833 \end{array}$$

⑥ 학생이 만든 비형식적 알고리즘의 예

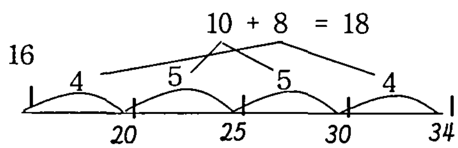
㉮ 시각화를 이용한 가로셈

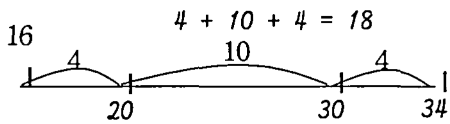




① 16을 기준점으로 보고, 1에서 차례대로 세어 18까지 센 후, 16부터 1씩 18번 뛰어 세면 34이다.

② 16부터 1씩 4번 뛰어 세면 20이다. 남은 14를 (4+10)으로 분해하여 20에서 4를 먼저 뛰어 센 후, 다시 10을 뛰어 센 수가 34이다





③ 18=10+8=(5+5)+(4+4)

$\underline{16+4}+5+5+4 = \underline{20+5}+5+4$

$= \underline{25+5}+4$

$= \underline{30+4} = 34$

④ 18=4+10+4

$\underline{16+4}+10+4$

$= \underline{20+10}+4$

$= \underline{30+4} = 34$

㉯ 암산을 이용한 세로셈

$$\begin{array}{r} 16 \\ +\ 18 \\ \hline \end{array}$$

10 + 10 = 20	10 + 10 = 20	10 + 10 = 20
6 + 8 = 14	6 + 4 = 10	6 + 6 = 12
20 + 10 = 30	20 + 10 = 30	12 + 2 = 14
30 + 4 = 34	30 + 4 = 34	20 + 10 = 30
		30 + 4 = 34

나. 뺄셈

① 더해가기(adding up)를 이용한 알고리즘

$$\begin{array}{rrr} & & +10 \\ & & +5 \\ 4 & 1 & +1 \\ -\ 2 & 5 & \\ \hline 1 & 6 & \end{array}$$

이 방법은 뺄셈의 비교적인 의미에 초점을 두거나 뺄셈이 덧셈의 역이라는 개념에 초점을 둔 것이다. 예를 들어 '41-25'에서 '45와 25는 얼마만큼 멀리 있는가?' 혹은 '25를 41로 만들기 위해서는 내가 얼마나 더 더해야 하는가?'와 같은 생각을 바탕으로 한 알고리즘이다. 25에다 10을 더하면, 35이고, 다시 5을 더하면 40이며, 41이 되기 위해서는 1을 더 더하면 된다. 달고 생각할 수 있다. 이 방법은 여러 단계의 더하는 과정을 통해 해결하는 알고리즘으로 표준화된 뺄셈 알고리즘의 사용 전이나 뺄셈에 익숙하지 않은 학생들에게 상당히 도움을 주지만 다루는 수의 규모가 클 경우 암산하는 것에 대한 부담감을 줄 수 있다.

② 부분차를 이용한 알고리즘

$$\begin{array}{rrr} & (1) & \\ 5 & 4 & 1 \\ -\ 2 & 7 & 9 \\ \hline 3 & 0 & 0 \\ -\ 3 & 0 \\ & -\ 8 \\ \hline 2 & 6 & 2 \end{array} \qquad \begin{array}{rrr} & & (2) \\ 5 & 4 & 1 \\ -\ 2 & 7 & 9 \\ \hline & & -\ 8 \\ & -\ 3 & 0 \\ 3 & 0 & 0 \\ \hline 2 & 6 & 2 \end{array}$$

이 방법은 덧셈에 있어서 부분합 알고리즘과 유사한 개념으로 표준 알고리즘과 달리 연산 순서가 (1)과 같이 왼쪽에서 오른쪽으로 하거나, (2)와 같이 오른쪽에서 왼쪽으로 하거나 관계없이 연산이 이루어진다. '541-279'에서 피감수보다 감수가 큰 경우 즉 십의 자리의 계산에서 40-70의 값을 백의 자리에서 빌려오는 개념보다는 손해적인 표현으로 '-'를 도입한다.

③ 동수등가(同數等加)법을 이용한 알고리즘(equal-addition algorithm)

$$\begin{array}{rr} & 1 \\ 9 & 1 \\ 3 & \\ -\ \cancel{2} & 4 \\ \hline 6 & 7 \end{array} \qquad \begin{array}{l} 11 - 4 = 7 \\ 9 - 3 = 6 \end{array}$$

이 방법은 피감수와 감수에 각각 같은 수를 더한 다음 미산법으로 계산하는 방법으로 표준 알고리즘을 지도하기 이전 단계에서 활용될 수 있다. 왼쪽 계산의 경우, 피감수에 '10'을 더하여 11에서 4를 뺀 차를 일의 자리에 쓰고, 피감수 90에서 감수 20에도 '10'더 더한 30를 뺀 차를 십의 자리에 쓴 것이다.

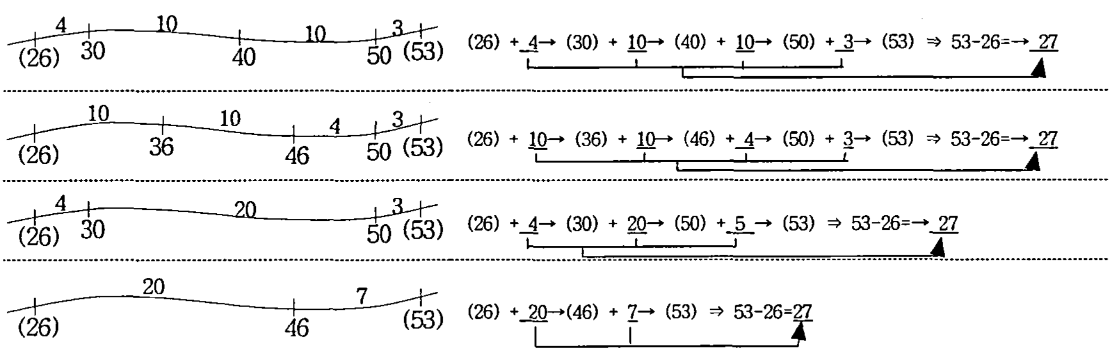
④ 9의 보수를 이용한 알고리즘

이 방법은 감수를 9에 대한 감수의 보수로 대치하고 피감수에 더하는 방법으로 덧셈에 비해 뺄셈에 대해 불안감을 갖는 학생들에게 유용한 방법이다.

$\begin{array}{r} 2\,3\,8\,4 \\ -\ \cancel{1}\,\cancel{6}\,\cancel{9}\,\cancel{5} \\ \hline 8\,3\,0\,4 \end{array}$	· 자리값이 같은 수끼리 세로셈 형식으로 나타낸다. · 감수 대신에 자리값이 다른 각각의 자리에 있는 수를 '9'에 대한 그 수(감수)의 보수를 감수 아래 쓴다.
$\begin{array}{r} 2\,3\,8\,4 \\ +\ 8\,3\,0\,4 \\ \hline 1\,0\,6\,8\,8 \end{array}$	· 피감수에서 각 자리에 있는 '9'에 대한 감수로 나타낸 수끼리 더한다.
$\begin{array}{r} \cancel{1}\,0\,6\,8\,8 \\ +\quad\quad\ 1 \\ \hline 6\,9\,9 \end{array}$	· 합의 첫째 자리에 있는 숫자인 '1'을 지우고, 일의 자리에 '1'을 더한다.

⑤ 수선(數線 : Number line & Number curve)

이 방법은 직선이나 곡선에 수를 표시함므로서 계산과정에서 시각적 직관력과 양감을 이용하는 방법이다. 문제상황을 시각화하면서 간단한 부분계산으로 단순화시켜 계산을 하는 방법으로 계산과정에서 정신적 조작에 따르는 학습불안을 줄일 수 있으며, 계산지도의 초기 단계나 계산학습 부진을 예방하기 위한 방안으로서 활용하는 데 효과적이며, 암산 능력이 발달할수록 그 단계를 축소할 수 있다. 그러나 차가 큰 수를 계산할 경우, 선에서 나타내는 수 사이의 거리를 비율에 따라 표현하기 어려우므로 두 수 사이의 간격을 임의로 나타낼 수 밖에 없어 양감 형성에는 바람직하지 못한 영향을 줄 수도 있다. 뺄셈 '53-26'을 수선를 이용하여 접근해보면 다음과 같다.

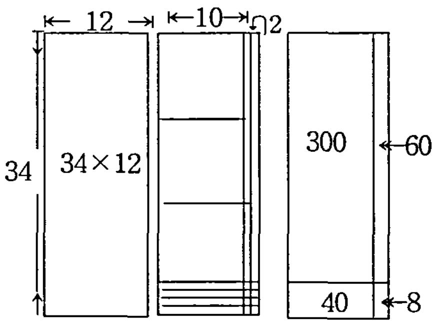


⑥ 학생이 만든 비형식적 알고리즘의 예

44	40 - 10 = 30	40 - 10 = 30	40 - 10 = 30
- 15	4 - 5 = 0보다 1작은	30 - 5 = 25	30 + 4 = 34
	30 - 1 = 29	25 + 4 = 29	34 - 5 = 29

다. 곱셈

① 그림을 이용한 알고리즘



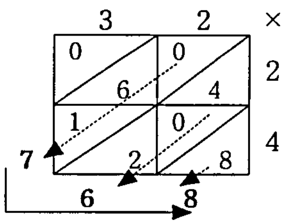
$$\begin{array}{r} 300 \\ 60 \\ 40 \\ +\quad 8 \\ \hline 408 \end{array}$$

② 전개식을 이용한 알고리즘

34×12=□

34×12 = 34×(10+2) ←- 전개
= (34×10)+(34×2) ←- 분배
= (30+4)×10+(30+4)×2 ←-전개
= (30×10)+(4×10)+(30×2)+(4×2)←-분배
= 300+40+60+8 ←- 곱셈
= 408 ←- 덧셈

③ 격자를 이용한 알고리즘



곱셈의 절차에서 고쳐묶기와 자릿값에 대해 그다지 신경쓰지 않아도 되는 방법으로 가로줄 윗쪽과 세로줄 오른쪽 부분에 각각 승수와 피승수를 적는다. 그리고 나서 각각을 계산한 부분값을 대응하는 칸에 기록한다. 예를 들면 '32×24'을 계산할 때 2와 2의 곱의 값인 04은 오른쪽 위칸에 기록하고 3와 2의 곱의 값인 06은 왼쪽 윗칸에 기록한다. 모든 곱셈 과정이 끝나면 격자 속에 적힌 수들을 대각선 방향으로 합하고 그 값을 시계 반대 방향에서 읽으면 32와 24의 곱의 결과가 된다.

④ 러시아 소작농의 곱셈법을 이용한 알고리즘

①	55	26
②	27	52
③	13	104
~~④~~	~~6~~	~~208~~
⑤	3	416
⑥	1	832
		1430

과거 러시아의 소작농들이 사용했다는 이 방법은 곱셈구구에 대한 지식이 없이도 곱셈을 할 수 있다. 오른쪽 그림처럼 곱하고자 하는 두 수를 나란히 옆에 쓰고(피승수는 왼쪽, 승수는 오른쪽), 한 쪽(승수)은 계속 2배를 하고 다른 한 쪽(피승수)은 2로 나누는 과정을 반복한다. 계속 2로 나눈 쪽(피승수)이 1이 되었을 때, ④단계의 '6 206' 처럼 피승수가 2의 배수인 경우는 제외하고, 2배를 한 쪽(승수)의 수들을 합이 두 수의 곱이다.(Sgroi, 1998). 왼쪽의 예는 '26×55'의 계산 과정을 나타낸 것이다.

⑤ 2의 거듭제곱(Duplication)을 이용한 알고리즘

이 방법은 2를 거듭제곱한 수의 성질을 이용하여 곱을 구하는 방법이다. 예컨대 28을 2의 거듭제곱한 수가 피승수, 즉 28보다 커지면 멈춘다. 1, 2, 4, 8, 16 중 28(=16+8+4)이므로 28×36=(16+8+4)×36의 값과 같다. 따라서 28과 36의 곱은 '*' 한 수들의 합, 즉 144+288+576=1008이다.

28×36=□

1×36 = 36
2배 ↓ 2배
2×36 = 72
2배 ↓ 2배
4×36 = 144 *
2배 ↓ 2배
8×36 = 288 *
2배 ↓ 2배
16×36 = 758 *

```
  1 4 4
  2 8 8
+ 7 5 8
-------
1 0 0 8
```

⑥ 학생이 만든 비형식적 알고리즘의 예

```
 135     4×100 = 400          4×100 = 400
× 4      4× 30 = 120          4× 35 = (2+2)×35
         4×  5 = 20                 = 70+70 = 140
         400+120+20 = 540     400 + 140 = 540
```

라. 나눗셈

표준 알고리즘을 이용하여 지필 계산할 경우, 대부분의 학생들은 사칙 연산 알고리즘 중에서 나눗셈을 가장 어려워한다. 그 이유를 네 가지로 요약하면 다음과 같다. ① 덧셈, 뺄셈, 곱셈은 자리값이 낮은 자리부터 높은 자리의 순서로 계산하지만 나눗셈의 경우는 그 반대로 자리값이 높은 자리부터 낮은 자리의 순서로 계산한다. ② 덧셈, 뺄셈, 곱셈은 한 가지 연산자를 이용하여 답을 계산하지만 나눗셈은 곱셈(몫과 젯수)과 뺄셈(피젯수에서 몫과 젯수의 곱)의 2가지 연산자를 동시에 이용해야 한다. ③ 덧셈, 뺄셈, 곱셈은 덧셈·뺄셈 구구를 이용하여 곧바로 합·차 및 곱을 곧바로 정하지만 나눗셈은 피젯수 안에 젯수가 몇 인지를 어림하여 가정몫을 정해보는 과정을 거쳐야 한다. ④ 젯수와 피젯수 사이의 관계를 고려하여 몫의 자리값을 지정하는데 혼란을 일어키기 쉽다.

① Scaffold법을 이용한 알고리즘

앞서서 언급한 것처럼 나눗셈 알고리즘은 뺄셈과 곱셈이 동시에 수반되는 연산으로 특히 가정몫을 어림하는 과정에서 겪는 많은 실패로 다른 연산에 비해 나눗셈을 더욱 어렵게 생각하고 있다. 이처럼 한 번에 몫을 정하는 것을 어려워하거나 또는 한 번 몫을 정하면 수정하지 않으려는 성향을 가진 학생들에게는 Scaffold 방법을 대안적으로 적용해보는 것이 좋다. Scaffold 계산법은 다음 <방법 1>, <방법 2>처럼 나머지가 제수보다 작을 때까지 피제수에 제수가 몇 번 들어있는지를 어림하여 구하는 방법으로 몫의 자리값에 대한 혼란을 줄일 수 있으나 몫을 추측하고 검토·확인하는 과정이 비효율적이다. <방법 1>은 저학년 초기에 나눗셈 알고리즘을 지도 방법으로서는 권고할 수 있으나 기본 원리가 이해된 이후에는 단계 ②와 ③을 합하여 표준 알고리즘 지도의 중간 과정으로 지도하는 것이 바람직하다고 보겠다.

<방법 1> 5 7 3 9 ÷ 3 1 = □

31)5739 ← ① 5739에는 31이 얼마쯤 들어있을까?. 어림 : 100
3100 ← (100×31)
2639 ← ② 2639에는 31이 얼마쯤 들어있을까?. 어림 : 50
1550 ← (50×31)
1089 ← ③ 1089에는 31이 얼마쯤 들어있을까?. 어림 : 30
930 ← (30×31)
159 ← ④ 159에는 31이 얼마쯤 들어있을까?. 어림 : 5
155 ← (5×31)
4 1 8 5 ← ⑤ 4<31이므로, 5739에 들어있는 31의 수를 모두 합하면,
100+50+30+5=185이고, 4가 남는다. 즉 몫은 185이고, 나머지는 4이다.

<방법 2> 5 7 3 9 ÷ 3 1 = □

1 8 5 ※ 답을 젯수 위에 거꾸로 쓰고, 그 합을 구한다.
5 5739에 들어있는 31의 수를 모두 합하면, 100+50+30+5=185이고, 4가 남는다.
30 따라서, 몫은 185이고, 나머지는 4이다.
50
100
31)5739 ← ① 5739에는 31이 얼마쯤 들어있을까?. 어림 : 100
3100
2639 ← ② 2639에는 31이 얼마쯤 들어있을까?. 어림 : 50
1550
1089 ← ③ 1089에는 31이 얼마쯤 들어있을까?. 어림 : 30
930
159 ← ④ 159에는 31이 얼마쯤 들어있을까?. 어림 : 5
155
4 ← ⑤ 4<31이므로, 더 이상 계산하지 않는다.

② 학생이 만든 비형식적 알고리즘의 예

산가지 156개를 한 묶음에 4개씩 묶으면 모두 몇 묶음이나 되겠는가?		
<학생 A>의 계산	<학생 B>의 계산	<학생 C의 계산>
$156 \div 4 = 10$ 25 -40 10 $116 \div 4 = 4$ + 4 -16 39 $100 \div 4 = 25$ -100 0	156 $-40 \cdots 10 \times 4$ 116 $-40 \cdots 10 \times 4$ 76 $-40 \cdots 10 \times 4$ 36 $-36 \cdots 9 \times 4$ 0	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 4×4, 4×10와 4×25=100이라는 사실을 이용하기 위하여 156에서 100이 될 때까지 4씩 몇 묶음을 뺀 후, 4개씩 묶음의 수를 합하여 39묶음이라는 사실을 찾아냈다.	위 학생은 4의 배수에 대한 지식이 넉넉한 학생으로 156으로부터 40씩 연감산을 통해 156에는 4가 39번 들어있음을 알아냈을 것이다.	위 학생은 먼저 4개의 열을 만든 후, 먼저 20개씩 분배한 후, 남은 것을 5개, 10개 차례로 분배하고, 나머지가 없을 때까지 1개씩 분배하여 각 열에 분배된 개수의 합이 39임을 알아냈을 것이다.

6. 대안적 알고리즘 지도상의 유의점

학생들이 직접 알고리즘을 탐구하고 고안해보는 의미있는 수학 활동이 되기 위해서는 교사의 역할이 매우 중요하다. 대안적인 알고리즘을 효과적으로 지도하기 위해 유의할 점은 다음과 같다.

① 학생들에게 탐구할 시간을 충분히 제공해야 한다.

성공적인 알고리즘 수업이 되기 위해서는 (1) 학생들에게 문제를 제시하고, (2) 개별적으로 또는 소집단으로 문제 해결 방법을 고안해 낸 다음, (3) 각자가 사용한 다양한 방법을 토론하며 자신이 만든 절차가 합리적인지를 살펴보거나 혹은 해결 과정에서 어려움을 겪는 부분은 다른 학생들이 제시한 대안적인 접근 방법을 통해 배울 수 있는 기회를 제공해야 한다.(남승인·서찬숙, 2000) 이와 같은 알고리즘 수업이 되기 위해서는 학생들이 여유롭게 탐구하면서 해결 방법을 찾을 수 있도록 충분한 시간을 주어야 할 것이다.

② 가치있는 문제 장면을 제시한다.

실제로 학생들은 교실 밖에서 접하는 문제 상황에 대해서는 무조건 계산하기보다는 직관이나 자신의 경험에 비추어 해결하려고 노력을 하지만 교실 안에서는 실생활과는 동떨어진 의미없는 수들의 연산이나 절차를 기억하거나 반복·연습할 것을 요구받는다. 이처럼 알고리즘의 반복·연습을 할 수 있는 문제만 접하면 학생들의 계산력과 신속성은 높일 수 있으나 생각하는 것을 포기하게 만들고 논리적인 사고를 방해하기 때문에 자신이 배우지 않은 문제라고 여기는 문제에 대해서는 해결하려는 노력도 없이 쉽게 포기해버리고 만다. 따라서 무의미한 기호의 조합만을 학생들에게 제시할 것이 아

니라 해결해보고자 하는 동기를 부여할 수 있는 의미있는 상황의 문제를 제공하여 실제 상황에서 문제를 해결하듯이 적절한 직관과 논리를 통하여 자기 나름의 해결 방법을 찾아 보게 한다.

③ 학생들이 고안한 다양한 알고리즘을 서로 공유한다.

학생들에게 자기 나름의 방식으로 계산할 기회를 주고 가장 적합한 알고리즘을 선택하여 계산할 때 자신이 만든 알고리즘이 정확한 답이 산출하고 있는지 또 다른 사람들도 이해하기 쉽도록 단순해야 하고 적용하는 데에 효율적이며 합리적으로 구성되어 있는지 알아볼 필요가 있다. 따라서 이러한 활동이 이루어질 수 있도록 다양한 알고리즘을 공유할 수 있는 토의 시간을 주어야 한다.

④ 편안한 분위기를 조성해준다.

교사는 연산 과정에서 발생하는 빈번한 오류들에 대해 질책하기보다는 관심을 가져 오류를 수정해주어야 한다. 만약 제 때에 도움을 받지 못한다면 그 학생은 연산의 절차나 개념을 배우는 기회를 놓칠 수 있기 때문에 어려움에 직면한 학생에게는 교사가 도움을 주는 역할을 해야 한다. 따라서 학생들이 마음껏 모험도 하고 오류를 범해도 괜찮으며 언제든지 교사에게 도움을 받을 수 있는 허용적인 분위기를 조성해야 한다. 또 학생들이 필요하다면 사고 과정을 뒷받침해줄 수 있는 구체적 조작물을 사용하는 것도 기꺼이 허락해야 한다.

⑤ 학생들이 고안한 알고리즘의 교육적 가치에 대한 올바른 평가가 이루어져야 한다.

학생이 스스로 알고리즘을 창안・활용하는 것은 교육적으로 매우 가치있는 일이며 권장할 일이다. 그러나 학생이 창안한 알고리즘이라고 하여 모두 가치있는 것은 아니다. 학생들이 창안한 알고리즘에 대해서 적어도 다음 세 가지 관점에서 평가가 이루어져야 할 것이다.

첫째, 알고리즘의 효율성을 따져보아야 한다. 즉 알고리즘을 수행하는데 따른 간편성, 신속성에서 그 가치를 판단해야 할 것이다. 시간이 너무 오래 걸린다든가, 수행과정이 복잡하다든가, 부정확한 답을 산출하는 알고리즘은 그 가치가 상대적으로 낮다고 볼 수 있다.

둘째, 수학적으로 타당한지를 분석・판단해야 한다. 즉 알고리즘의 수행과정에서 논리적인 모순이 발생하지 않는지 살펴보아야 한다. 아무리 신속・간편하다고 하더라도 수학의 대수적 이론과 충돌이 발생한다면 그것은 알고리즘으로서 가치가 없다. 다음의 [예 1]과 [예 2]의 알고리즘은 효율성이나 일반화 가능성에서는 다소 의문이 제기되지만 논리적인 모순은 없으므로 수학적으로는 그 절차가 타당하다고 할 수 있다.

셋째, 일반화가 가능성에 대해서 분석・평가해야 한다. 아무리 정확・신속・간편한 알고리즘이라 하더라도 특수한 문제에 국한된 알고리즘은 가치가 없다. 예컨대 다음 [예 1]의 경우는 우리 눈에는 다소 어색하여 일반화 가능성이 없는 것으로 생각할 수 있다. 그러나 큰 수를 사용할 경우, 계산과정이 길어지고 복잡하여 효율성은 다소 떨어지지만 논리적 모순이 없으므로 일반화가 가능하다. 또 [예 2]의 나눗셈 '126÷17' 의 알고리즘은 논리적 모순은 없으므로 수학적으로 타당하다. 그러나 몫을 계산하는 절차가 복잡하고 효율성이 떨어질 뿐만 아니라, 소수의 나눗셈으로 확장할 경우 장애가 발생하므로 일반화할 가치는 적다.

$$\begin{array}{rrrl} 6 & 4 & 3 & \\ -\;4 & 8 & 6 & 3 \\ \hline 2 & 0 & 0 & \\ \hline 1 & 6 & 0 & \\ \hline 1 & 5 & 7 & \end{array}$$

· 600 - 400 = 200
· 80 - 40 = 20
· 200-40 = 160
· 6 - 3 = 3
· 160 - 300 = 157

[예 1]

$$\begin{array}{r} 126 \\ \div\ 17 \\ \hline \frac{6}{17} \\ 1\frac{3}{17} \\ 5\frac{5}{17} \\ \hline 6\frac{24}{17} = 7\frac{7}{17} \end{array}$$

· 6을 17로 나누면 $\frac{6}{17}$ 이다.
· 20을 17로 나누면 $1\frac{3}{17}$ 이다.
· 100을 17로 나누면 $5\frac{5}{17}$ 이다.
· 몫을 모두 더한다.

[예 2]

8. 맺으며

역사 이래 사용해 온 계산 절차인 지필 알고리즘은 수학적 성취의 필수적인 요소로 보는 것이 주된 관점이었고 최근 계산 학습에서 알고리즘 지도는 학교 수학에서 주된 관심거리이다. 요즈음 대부분의 초등 학생들은 졸업할 무렵이면 지필 알고리즘을 이용하여 기본 연산인 사칙 계산을 할 수 있다. 그러나 그 중 많은 아이들이 자신이 실제로 대수적 구조나 계산 원리를 바르게 이해하지 못한 채 자신이 무엇을 하고 있으며, 왜 그렇게 하는지를 실제로 이해하지 못하고 반복 연습을 통해 익힌 형식화된 알고리즘을 기계적으로 적용하여 답을 구하는 경우가 많다(Whitebread, 1995). 예컨대 곱셈의 의미도 모른 채 곱셈구구를 능숙하게 외우거나 분수의 의미도 제대로 이해하지 못한 채 분수의 사칙 계산을 하기 위해 기계적으로 암기한 알고리즘을 맹목적으로 적용하는 일 등 관계적 이해를 바탕으로 한 알고리즘 학습이 아닌 도구적 이해에 의존하여 학습을 수행함으로써 수학학습을 기피하고 불안해 하거나 혼란스러워하는 모습을 종종 볼 수 있다.

최근 NCTM(1998), Carroll & Porter(1998), Compbell, er al(1998)은 앞으로의 계산학습에서 지필 계산에 대해 지나치게 의존하는 것을 경고하면서 학생 스스로 절차를 고안·활용하는 비형식적인 알고리즘과 이전 세대에서 활용했던 전통적인 알고리즘, 즉 대안적인 알고리즘의 필요성을 주장하였다. 이는 최근 학습 이론의 사조인 구성주의적 관점에 근거한 교수·학습활동을 권고하는 암시이며 정보화의 시대라고 일컫는 요즈음의 사회적 요구, 즉 지필을 대신할 수 있는 계산기나 컴퓨터 등 새로운 계산 도구의 개발·보급 및 상용화로 인해 지필 계산의 교육적 가치와 효율성이 상대적으로 낮아지고 있기 때문으로 생각된다.

대안적 알고리즘은 표준 알고리즘과 구별하기 위하여 사용된 용어로 학생 중심의 알고리즘 학습을 의미한다. 대안적인 알고리즘 지도의 필요성은 학생들의 내적 특성에 부합되는 것으로 자기 나름의 절차를 만들어가는 동안 규칙이나 절차의 의미를 이해하게 되고, 정확한 연산 결과를 얻기 위해 다양한 방법으로 시도해 보는 과정에서 수 감각이나 연산 감각이 함께 향상될 수 있기 때문이다. 그러나 보다 직접적인 이유는 표준 알고리즘의 원리를 이용하여 문제를 해결하기 어려운 학생은 문제해결을 포기할 가능성이 있다. 따라서 그들에게 효율성 면에서는 다소 뒤떨어짐다고 하더라도 대안

적인 알고리즘을 통하여 당면한 문제를 해결할 수 있는 능력을 길러줌으로써 후속하는 학습을 가능하게 해 주는 데 있다. 이를 위하여 교사는 전통적인 알고리즘에 대한 소개와 함께 학생들이 직접 알고리즘을 탐구・고안・이용・토론해 볼 수 있는 환경과 기회를 제공해야 할 것이다.

참 고 문 헌

강시중 (1993). 수학교육론. 서울: 교육출판사

김용국 (1991). 수학의 토픽스, 서울: 전파과학사.

교육부 (1998). 초등학교 교육과정 해설서 IV, 서울: 대한교과서주식회사.

서찬숙・남승인 (2000). 초등학교에서의 알고리즘 지도의 필요성과 지도방법, 대한수학교육학회 논문집 2000년 추계, pp.383-396.

Burger, W.F & Musser, G.L. (1996). *Mathematics for Elementary Teachers.* Prentice-Hall, Inc. Simon & Schuster? A Viacom Company Upper Saddle River,NJ 07458.

Campbell, P.E.; Rowen, T.E. & Suarez, A.R. (1998). What criteria for student-invented algorithm?: In L. J. Morrow & M. J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics*, pp.49-55, Reston, VA: NCTM, INC.

Carroll, W.M & Poter, D. (1998). Alternative Algorithms for Whole-number Operations. In L.J. Morrow & M.J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics*, pp.106-114, Reston, VA: NCTM, INC.

Kamii, C. & Dominick, A. (1998). The Harmful Effects of Algorithms in Grades 1-4. In L.J. Morrow & M.J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics*, pp.130-140. Reston, VA: NCTM, INC.

Morrow(1998,L.J.(1998). Whither Algorithm? Mathematics Educators Express Their Views. *The Teaching and Learning of Algorithms in School Mathematics.* Reston, VA: NCTM, INC.

Maurer, S.B. (1998). What Is an Algorithm? What Is an Answer?. In L.J. Morrow & M.J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics*, pp.21-31, Reston, VA: NCTM, INC.

NCTM(2000). Standards 2000, *Principles and Standards for School Mathematics* : Discussion Draft. Reston, VA: NCTM, INC.

Sowder, J. (1992). Estimation and Number Sense. Douglas A. Grouws, (Eds) *In Handbook of Research on Mathematics Teacning and Learning*, pp.371-389. New York: Macmillan Publishing Co.

Usiskin, Z. (1998). Paper and Pencil Algorithms in a Calculator and Computer Age. In L.J.

Morrow & M.J. Kenney (Eds), *The Teaching and Learning of Algorithms in School Mathematics*, pp.7-20. Reston, VA: NCTM, INC.

Whitebread. David. (1995). Emergent Mathematics or How to Help Young Children Become Confident Mathematicians. *Children's Mathematical Thinking in the Primary Years.* Perspectives on Children's Learning. Julia Anghileri and the Contributors. 215 Park Acenue South. New York.