12th International Congress on Mathematical Education
Program Name XX-YY-zz (pp. abcde-fghij)
8 July – 15 July, 2012, COEX, Seoul, Korea (This part is for LOC use only. Please do not change this part.)

# LEARNING OF ALGORITHMS: A THEORETICAL MODEL WITH FOCUS ON COGNITIVE DEVELOPMENT

Lianghuo Fan

Southampton Education School, The University of Southampton

L.Fan@southampton.ac.uk

*Taking a broad perspective on algorithm in mathematics, the author presents a theoretical model about the learning and teaching of algorithm with focus on students' cognitive development. The model consists of three cognitive levels: 1. Knowledge and Skills, 2. Understanding and Comprehension, and 3. Evaluation and Construction. The model suggests that teaching and learning of algorithm does not simply mean routine learning, memorization, or lead to a low level of cognition. The paper also discusses different teaching strategies and activities that can be used to support students' cognitive development at different cognitive levels.*

*Key Words: Algorithms, Cognition, Cognitive Development, Mathematics Learning, Mathematics Teaching.*

## INTRODUCTION: NATURE OF ALGORITHMS

Algorithm is a most commonly used word in modern mathematics and computer science, although there have been some slight differences in defining the concept of algorithm. In general, algorithm can be regarded as a certain sort of general method for solving a family of related questions (Taylor, 1998). The Encarta World English Dictionary defines algorithm in two uses: (1) as a problem-solving procedure, which is a logical step-by-step procedure for solving a mathematical problem in a finite number of steps, often involving repetition of the same basic operation, and (2) as a computing program, which is defined as a logical sequence of steps for solving a problem, often written out as a flow chart, that can be translated into a computer program (Rooney, 1999).

In this article, an algorithm is defined as a fixed set of step-by-step procedures for solving a particular kind of mathematics problems. An algorithm must have the following three essential features: certainty, reliability, and generalizability.

- **Certainty** – The procedures in the algorithm to follow in solving a problem are fixed and unambiguous.
- **Reliability** – The correct answer can always be obtained if one follows the algorithm step-by-step correctly (even if he/she does not really understand the steps involved).
- **Generalizability** – Any algorithm can be applied to solve a family of problems.

From a broad perspective as mentioned above, we can see that most, if not all, mathematics theorems (e.g., the theorem for finding the area of a triangle), formulas (e.g., the quadratic formula for solving general quadratic equations), and methods (e.g., the synthetic division method for polynomial division) are essentially algorithms. In fact, if we view mathematics as a science of patterns, we can see that, to a large degree, algorithm is mathematics itself.

In contrast to the fact that algorithms are viewed by researchers in computational mathematics and computer sciences to be of central importance, the teaching and learning of algorithm in modern school mathematics are often given little importance or de-emphasized. Unlike terms such as open-ended questions, contextualized problems, and real-life applications, which are generally linked to the development of students' understanding, creative thinking, and higher-order cognitive skills, the term of algorithm in school mathematics is often related, with negative connotation, to routine procedures, rote learning and memorization, and lower level cognitive skills.

The negative perceptions about teaching of algorithm are found in researchers, curriculum developers, as well as practitioners in mathematics education. For example, in analyzing classroom teaching using TIMSS 1990 Video Study, negative views about the mathematics lessons were described as "very algorithmic", "rule-orientated", and "at a low level and too focused on procedures and rules, with not enough attention to mathematical concepts and reasoning" (Givvin, Jacobs, Hollingsworth, & Hiebert, 2009; also see Leong, Dindyal, Toh, Quek, Tay, & Lou, 2011). In another study with grade 2 to 4 classes about multi-digit addition and multiplication algorithms, the researcher concluded that "algorithms are harmful to children's development of numerical reasoning for two reasons: (a) they "unteach" place value and discourage children from developing number sense, and (b) they force children to give up on their own thinking." (Kamii & Dominick, 1997). Another similar case but with stronger view against teaching algorithm is, as it is claimed, "focusing on teaching algorithms can be very destructive while, to the contrary, focusing on solving problems through common sense means can be very empowering" (Davis, 2011). Because of the widely held negative perceptions about algorithm for students' learning in mathematics, it is not surprising to see that algorithm has been de-emphasized in mathematics curriculum and instruction in different countries. One extreme case recently reported online is that a new mathematics curriculum in Canada excludes the standard algorithms for the four basic arithmetic operations, and moreover, the curriculum is against the teaching of these algorithms (Craigen, 2011).

Given the widely held negative perceptions about the teaching and learning of algorithm, I think it is important and timely for the mathematics educators to re-look at the meaning of algorithm in modern mathematics and sciences, and re-think the values and ways of teaching and learning algorithm in schools (Fan, 2004). In this connection, I shall below present a theoretical (conceptual) model about teaching and learning of algorithm with focus on students' cognitive development.

## LEARNING OF ALGORITHM AND COGNITIVE LEVELS

Learning is a process of acquiring knowledge, skills, understanding, or values through different experiences or activities. From learners' perspective, Säljö (1979) once identified the following five categories of leaning, which are seen to be hierarchical (Ramsden, 1992, p. 26; also see Bailey, 2002).

1. Learning as a quantitative increase in knowledge. Learning is acquiring information.
2. Learning as memorising. Learning is storing information that can be reproduced.
3. Learning as acquiring facts, skills, and methods that can be retained and used as necessary.
4. Learning as making sense or abstracting meaning. Learning involves relating parts of the subject matter to each other and to the real world.
5. Learning as interpreting and understanding reality in a different way. Learning involves comprehending the world by reinterpreting knowledge.

Similarly, just like learning of many other things, learning of algorithm can also take place at different cognitive levels. Below I shall use examples of algorithm to describe three different cognitive levels in relation to students' learning of algorithm in schools.

**Cognitive Level 1: Knowledge and Skills**

At this level, the product of learning is to recognize and remember what it is and how it is carried out in a basically straightforward situation. For example, after students have learned the standard algorithm for multiplication of two multi-digit integers, they should remember it and be able to use it to find the product of two given integers, and after students have learned the theorem that the area of a triangle is half the base times the height, they should remember it and be able to use it to find the area of a triangle when the base and its height are given.

The learning activities at this level is mainly acquiring information (e.g., by listening to teachers or by reading the book), storing information (e.g., by remembering) so it can be reproduced, and retrieving information when it is used in essentially straightforward situations.

Although this level appears to be the lowest or basic level in learning of algorithm, it is still essential, particularly for fundamental algorithms or those commonly used in further learning of mathematics or in daily life. In fact, a large number of studies have revealed that many students lack knowledge and skills about some basic mathematics algorithms. A particular example is about the addition, subtraction and division of fractions (e.g., see Hasemann, 1981; Streefland, 1982). The TIMSS studies have also shown that students' mastery of basic algorithms in many countries is far from adequate. It remains a fundamental challenge to mathematics educators and teachers.

In a sense, this level is largely corresponding to Level 1, 2, and 3 in the classification of learning by Säljö (1979) or Level 1 (knowledge) in Bloom's taxonomy (Bloom, Engelhart, Furst, Hill, & Krathwohl, 1956).

**Cognitive Level 2: Understanding and Comprehension**

At this level, the product of learning is to know why an algorithm works and how it can be used in a relatively complex situation. For example, in learning of the formula for find the sum of the first $n$ terms of an arithmetic progression,

$$\sum_{i=1}^{n} a_i = \frac{n}{2}(a_1 + a_n)$$

at the knowledge and skills level, the students should know the formula itself and be able to use it in simple situations (e.g., $n$, $a_1$, and $a_n$ are given or can be easily found), but at the understanding and comprehension level, they should also understand why the formula is true or how it is derived, and be able to use it in a more complex situations, for example, using the following equivalent formula with $n$, $a_1$, and $d$, the common difference:

$$S_n = \frac{n}{2}[2a_1 + (n-1)d]$$

Learning of an algorithm at Level 1 and Level 2 can take place at almost the same time or within a short time interval in many cases. In fact, they can often reinforce each other, in other words, learning at Level 1 is more effective when learning at Level 2 also takes place. However, in other cases, it might not be feasible for students to achieve a thorough understanding at Level 2. For example, students at the lower primary school level can relatively easily know, remember and use that if the sum of all the digits of an integer is divisible by 3, then the integer is divisible by 3, but it is much more challenging for them to understand why it is mathematically true, which can be deduced from either of the following algebraic expressions (note: $a_n$, $a_{n-1}$, ..., $a_2$, and $a_1$ are from 0 to 9 in difference places of an integer; $a_n \neq 0$)

(1) $a_n a_{n-1} \ldots a_2 a_1 = a_n \times 1000\ldots0 + a_{n-1} \times 100\ldots0 + \ldots + a_3 \times 100 + a_2 \times 10 + a_1$

$= a_n \times (1 + 999\ldots9) + a_{n-1} \times (1 + 99\ldots9) + \ldots + a_3 \times (1 + 99) + a_2 \times (1 + 9) + a_1$

$= (a_n + a_{n-1} + \cdots + a_1) + (999\ldots9 + 99\ldots9 + \ldots + 99 + 9)$

(2) $a_n a_{n-1} \ldots a_2 a_1 = a_n \times 10^{n-1} + a_{n-1} \times 10^{n-2} + \ldots + a_3 \times 10^2 + a_2 \times 10 + a_1$

$= a_n \times (9 + 1)^{n-1} + a_{n-1} \times (9 + 1)^{n-2} + \ldots + a_3 \times (9 + 1)^2 + a_2 \times (9 + 1) + a_1$

$= (a_n + a_{n-1} + \cdots + a_1) + 9 \times m$

where,

$$m = \sum_{r=1}^{n-1} \sum_{k=0}^{n-r-1} \binom{n-r}{k} 9^{n-r-k}$$

Although the first method above is easier to understand as it basically requires the knowledge of place value while the second method requires knowledge of the binomial expansion, the second method is more general and powerful as it can be further applied to obtain the method for divisibility of an integer by other divisors (e.g., by 11)[1]. From the perspective that understanding is connected knowledge (Waddington, 1995), we can see that learning of algorithm at this level requires students not only to have more knowledge as compared to Level 1, but also to make the connection of different knowledge they have learned.

---

[1] Over the last few years, I have intentionally asked a large number (>100) of pre-service teachers and graduate students, and surprisingly, most of them were not able to explain why the method holds true, although they did remember the method. To some extent, it reflects the way they were taught when they were students in schools.

Level 2 is largely corresponding to Level 4 in the classification of learning by Säljö (1979) and, to a varying degree, Level 2 (comprehension), Level 3 (application), and Level 4 (analysis) in Bloom's taxonomy (Bloom, et. al, 1956).

**Cognitive Level 3: Evaluation and Construction**

At this level, the product of learning is to be able to judge the value or worth of an algorithm and to re-construct an already existing algorithm or construct a new algorithm.

Learning of algorithm at this level goes beyond simply knowing, memorising, applying, justifying or connecting an algorithm with other knowledge. It requires critical and creative thinking. For example, after students learned the following solution

$$\frac{1}{1\times 2}+\frac{1}{2\times 3}+\cdots+\frac{1}{n\times(n+1)}=\left[\left(1-\frac{1}{2}\right)+\left(\frac{1}{2}-\frac{1}{3}\right)+\cdots+\left(\frac{1}{n}-\frac{1}{n+1}\right)\right]=\frac{n}{n+1},$$

at the evaluation and construction level, they should not only know why it is true, how it can be used given a specific value of *n*, but also be able to judge if there is a need and possibility that the method can be generalized to solve a family of similar questions:

$$\frac{1}{n\times(n+1)}+\frac{1}{(n+1)\times(n+2)}+\cdots+\frac{1}{(n+k)\times(n+k+1)},$$

and more generally, to the following

$$\frac{1}{n\times(n+a)}+\frac{1}{(n+a)\times(n+2a)}+\cdots+\frac{1}{(n+ka)\times(n+ka+a)}$$

The above work requires students themselves to construct, or in this case more appropriately, re-construct an extended form of the above algorithm in their minds, even though they might not need to obtain and express the algorithm explicitly.

There is no doubt that learning of algorithm at this cognitive level will involve higher-order thinking skills and creative ability. In a sense, this level is largely corresponding to Level 5 in the classification of learning by Säljö (1979) or Level 5 (synthesis) and Level 6 (evaluation) in Bloom's taxonomy (Bloom, et. al, 1956). Moreover, learning at this level requires students to be engaged more in independent thinking and self-reflection.

It should be noted that although the three levels of cognitions described earlier are largely hierarchical, the learning of an algorithm can take place at different levels at the same time and not necessarily in a sequential manner from a lower level to a higher level.

## TEACHING OF ALGORITHM TO SUPPORT COGNITIVE DEVELOPMENT

Like the learning of algorithm can take place at different cognitive levels, the teaching of algorithm can also be carried out with focus on different cognitive levels. As the purpose of teaching is essentially to support learning, it is important for teachers to use effective teaching methods based on suitable cognitive level in students' learning and to facilitate their learning and cognitive development. Below I shall briefly explain how algorithms can be taught to suit to, and more importantly, support students' learning and cognitive development.

### Teaching at Cognitive Level 1: Knowledge and Skills

Corresponding to Cognitive Level 1, knowledge and skills, teaching of algorithm mainly involves the following four instructional activities, which can be basically described as "directing teaching":

- Telling: Teachers verbally let students what an algorithm is.
- Demonstration: Teachers show students, often using examples, how an algorithm works.
- Drill-and-Practice: Teachers ask students to do exercises in relatively straightforward situations, which are essentially routine and repetitive in nature.
- Remediation: Teachers help students correct their mistakes shown in their drill and practice.

To some degree, it is not possible for teachers to totally avoid directing teaching by lecture demonstration at this level because of different reasons (e.g., time constraint, students' level of learning, or the algorithm itself such as long division algorithm and some statistical methods). The effectiveness of teaching at this level, or in fact at any cognitive level, is also related to the factor of teachers and students (e.g., teachers' professional background and students' learning style also matter), for example, more experienced teachers can often teach more effectively than less experienced teachers when both using direct teaching.

Furthermore, as some researchers have argued, a reasonable amount of drill-and-practice and hence remediation, or correcting errors made by students, is not only necessary for the reinforcement of students' knowledge and skills, but often helpful for the development of students' cognition, and with variation, can lead to understanding (e.g., see Zhang, Li, & Tang, 2004; Li, 2006). The issue remaining is that how much amount of this type of learning is adequate.

On the other hand, as available literature has consistently revealed, teaching of algorithmic procedures merely focusing on this level, in lack of developing students' conceptual understanding, has been a main concern for many mathematics educators. Two such examples can be found in Leung (2006) about the multi-digit number multiplication and Lukhele, Murray, and Olivier (1999) for addition of fractions. In fact, it has been regarded as a main reason that students can only apply the algorithm mechanically or worse blindly, make procedural or conceptual mistakes and eventually develop negative attitudes about mathematics and learning of mathematics. Hence, it is necessary for teachers to teach algorithm with target beyond this level.

### Teaching at Cognitive Level 2: Understanding and Comprehension

Teaching of algorithm corresponding to this level often involves the following instructional activities, which can be described as "meaningful teaching" in connection with the term of meaningful learning.

- Explanation: Teachers explain to students what a part or the whole procedure of an algorithm means, and more importantly, why the algorithm works.

- Justification: Teachers let students understand how an algorithm is derived logically or can be proved mathematically.
- Making connections: Teachers help students make sense of the algorithm through connecting the algorithm with what students have learned or are familiar with.

Teachers' explanation about why an algorithm works can often be inductive and intuitive. For example, when teachers for the first time introduce to lower primary students the divisibility of an integer by 3, that is, if the sum of all the digits in a number can be divided by 3, then the number itself can be divided by 3, they can use some concrete numbers, for example, expressing 327 as $3\times(99+1)+2\times(9+1)+7 = 3\times99+2\times9 +(3+2+7)$ to explain why it works (instead of giving a rigorous proof).

Justification here can be viewed as a higher level of explanation. It is more on rigorous logical reasoning and proof, which is particularly important for students at a more advanced level. For example, teachers can use "completing the square method" to explain how the quadratic formula for solving general quadratic equations is derived. Another example is, to teach the formulas for finding the sum of the first *n* terms of an arithmetic progression, teachers can use the so-called Gauss's method to show how the formula can be obtained logically.

Making connection requires teachers to present an algorithm in a context that is known or familiar to students. It does not necessarily require rigorous logical deduction or proof, instead, the focus is on making meaningful connections so students can make sense of the algorithm.

Let me just briefly share one such example, i.e., the case of multi-digit number multiplication, which has been a difficult part for many young primary students to understand, remember and follow, and hence many mistakes were reported in literature (e.g., see Leung, 2006).

As a teacher educator, I personally have taught pre-service primary teachers the multiplication algorithm for years. The algorithm can be demonstrated using the following example:

$$\begin{array}{r} 1\,2 \\ \times\,1\,3 \\ \hline 3\,6 \\ 1\,2\phantom{\,0} \\ \hline 1\,5\,6 \end{array}$$

It was found that most trainee teachers remembered the algorithm in isolation, and they were unable to explain why it works, though they did how each number is obtained. Even when they were reminded that it can be linked to the expression of (10+2) × (10+3), many of them still said "that does not make much sense".

To help the trainees to connect the written algorithm with their previous knowledge of using marbles to represent multiplication of single digit numbers as repeated addition, I used the following pictorial representation for students to visualize the four parts: 3 × 2, 3 × 10, 10 × 2, and 10 × 10 (see Figure 1) and connect each of them to the corresponding number expressed in the long multiplication algorithm. The experiences show that the result was highly positive, and the main reason appears to be that the vertical written algorithm can be viewed as a symbolic representation of the pictorial representation, which is more familiar to students.

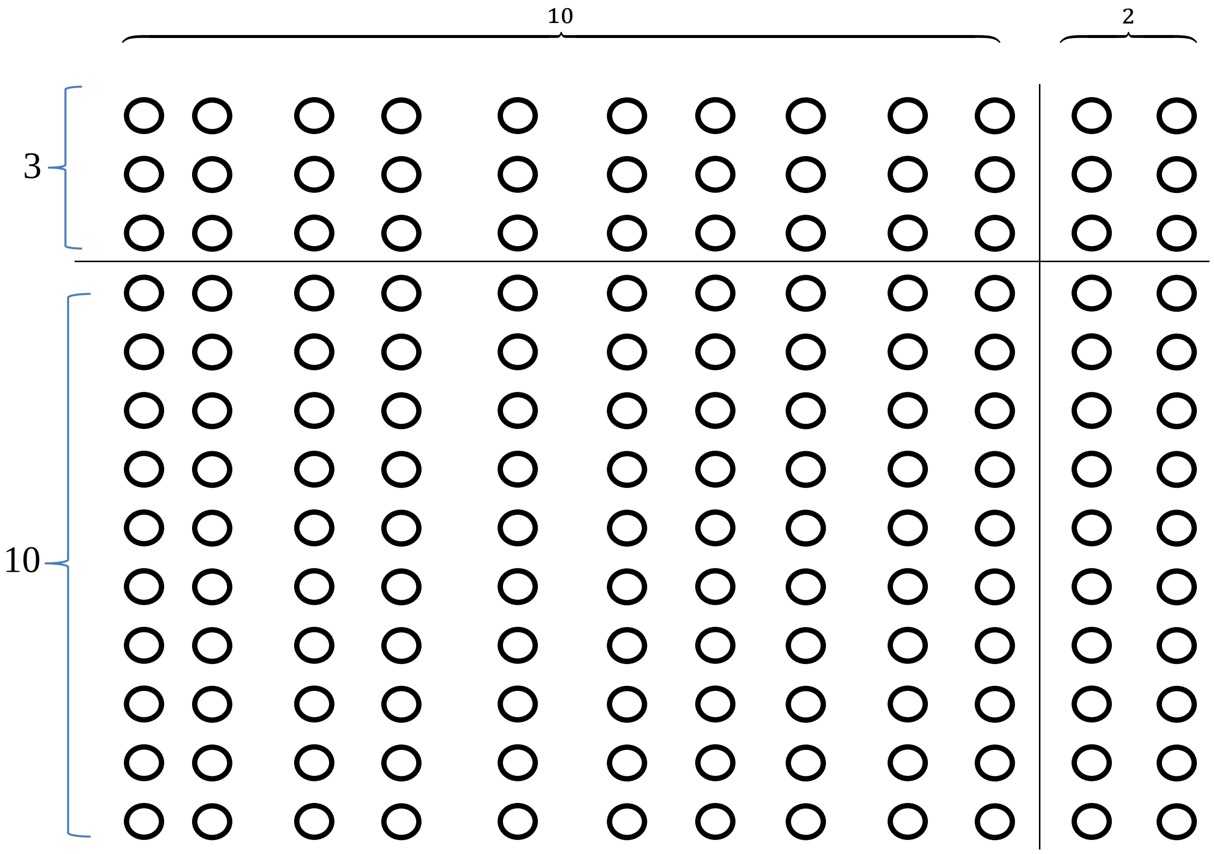


Figure 1. A visual representation to explain the standard written method for 12 x 13

It should be mentioned that the three teaching activities, explanation, justification, and making connections are often inter-connected and sometimes inseparable in teaching. The focus of teaching at this level is to help students make sense of an algorithm, which is a most important difference compared to teaching at the previous level. Apparently, teaching of algorithm at this level can also help students develop their knowledge and skills at Level 1.

**Teaching at Cognitive Level 3: Evaluation and Construction**

To develop and support students' learning at this cognitive level, teachers should provide the opportunities for students to engage in learning activities such as observing, analysing, identifying, constructing, and presenting the patterns.

According to students' learning activities, teaching at this level can involve guided exploration and open exploration.

- Guided exploration: Teachers create learning activities for students to explore and obtain the algorithm, while teachers provide a certain level of guidance in the process of students' exploration.
- Open exploration: Teachers create learning activities for students to explore and obtain the algorithm, and the process of exploration is basically or completely independent.

Guided exploration about an algorithm can often take place before the algorithm is introduced. For example, when teaching the factor theorem of polynomial division by $x - a$, teachers can let students first observe a number of concrete examples, for instance, $(3x^3 - x^2 + 5x - 7) \div (x - 1)$, then ask them to find the values of f($a$) and look for the common property, and finally obtain the theorem. Furthermore, depending on the students' background, teachers can also ask students to prove the theorem mathematically. On the other hand, open exploration can take place both before and after an algorithm is introduced. For example, after the divisibility of an integer by 3 is introduced, teachers can ask students, by open exploration, to investigate the divisibility of an integer by other integers such as 9 (easiest), 11 (easier), and 7 (most challenging) (e.g., see Lee, 1988).

The key philosophy of teaching at this level is to let students have more opportunities to be independent learners, make their own judgement, and construct their own knowledge (algorithm).

## CONCLUDING REMARKS

The following table provides a summary of the conceptual model discussed earlier in this paper.

Table 1: Learning and teaching of algorithm in different cognitive levels

| Cognitive level of learning | Teaching activities |
|---|---|
| 1. **Knowledge and skills**: recognizing what an algorithm is and how it is carried out in a straightforward situation | Telling, demonstration, drill and practice, and remediation |
| 2. **Understanding and comprehension**: knowing why an algorithm works and how it can be used in a relatively complex situation | Explanation, justification, and making connection |
| 3. **Evaluation and construction**: Being able to judge the value of an algorithm and to construct new algorithm on their own | Guided exploration and open exploration. |

Finally, I wish to point out that the model presented in this article is mainly based on my personal experiences as a researcher as well as a teacher educator. In particular, about ten years ago, working somehow like a mathematician instead like a mathematics educator as I was trained, I established a new algorithm for polynomial division by generalizing the classical synthetic division (Fan, 2002, 2003). While I have noted, gladly, that the algorithm has been used in computational sciences, the reflection of my own working process on the new algorithm provoked me to relook at the issues related to the teaching and learning of algorithm in school. The result is far from satisfactory and, in a sense, worrisome. It appears clear to me that the long-existing criticism, misconception, and mistreatment about the teaching and learning of algorithm in school curriculum and classrooms, as mentioned at the beginning of this paper, is more related to the fact that the teaching and learning has been unsatisfactorily at a low cognitive level, and less related to algorithm itself. As the model presented in this article suggests, the teaching and learning of algorithm does not simply mean routine learning, memorization, or lead to a low level of cognition. More broadly, given the fundamental value of algorithm in mathematics, I think it is both important and imperative for mathematics educators to seriously and systematically address the issues concerning the teaching and learning of algorithm in both theory and practice.

**References**

Bailey, S. (2002). Teaching students and supporting learning – is it the same thing? *Learning and Teaching in Action, 1*(3). Retrieved from http://www.celt.mmu.ac.uk/ltia/index

Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives: Cognitive domain*. New York: Longman.

Craigen, R. (2011). Responses to "Why do kids need to learn algorithms?" Retrieved from http://ahypatia.wordpress.com/2011/10/20/why-do-kids-need-to-learn-algorithms

Davis, G. (2011). Response to "Why it's okay to teach algorithms". Retrieved March 28, 2012, from http://mathfour.com/general/why-its-okay-to-teach-algorithms

Fan, L. (2002). A new algorithm for division of polynomials. In H. P. Lee & K. Kumar (Eds.), *Recent Advances in Computational Science and Engineering: Proceedings of the International Conference on Scientific and Engineering Computation* (pp. 717-721). London: Imperial College Press.

Fan, L. (2003). A generalization of synthetic division and a general theorem of division of polynomials. *Mathematical Medley*, *30*(1), 30–37.

Fan, L. (2004). Rethinking the values and ways of teaching algorithm in school mathematics. Powerpoint presentation at the annual conference of Association of Mathematics Educators, Singapore.

Hasemann, K. (1981). On difficulties with fractions. *Educational Studies in Mathematics*, *12*(1), 71-87.

Givvin, K. B., Jacobs, J., Hollingsworth, H., & Hiebert, J. (2009). What is effective mathematics teaching? International educators' judgments of mathematics lessons from the TIMSS 1999 Video Study. In J. Cai, G. Kaiser, B. Perry, & N.-Y. Wong (Eds.), *Effective mathematics teaching from teachers' perspectives: National and cross-national studies* (pp. 37-69). Boston: Sense Publishers.

Kamii, C., & Dominick, A. (1997). To teach or not to teach algorithms. *Journal of Mathematical Behavior*, *16*(1), 51-61.

Lee, P. Y. (1988). Divisibility by 7. *The New Zealand Mathematics Magazine*, 25-26, 17-22.

Leong, Y. H., Dindyal, J., Toh, T.L. Quek, K. S., Tay, E. G., & Lou, S. T. (2011). Teacher education for a problem-solving curriculum in Singapore. *ZDM: The International Journal on Mathematics Education*, *43*(6-7), 819-831.

Leung, F. K. S. (2006). Mathematics education in East Asia and the West: Does culture matter? In F. K. S. Leung, K-D. Graf, & F. J. Lopez-Real (Eds.), *Mathematics education in* *different cultural traditions: A comparative study of East Asia and the West* (pp. 21-46). New York: Springer.

Lukhele, R.B., Murray, H. & Olivier, A. (1999). Learners' understanding of the addition of fractions. In *Proceedings of the Fifth Annual Congress of the Association for Mathematics Education of South Africa* (Vol. 1, pp. 87-97). Port Elizabeth: Port Elizabeth Technikon.

Li, S. (1999). Does practice make perfect? *For The Learning of Mathematics*, *19*(3), 33-35.

Ramsden, P. (1992). *Learning to teach in higher education*. London: Routledge.

Rooney, K. (Ed.) (1999). *Encarta World English Dictionary*. London: Bloomsbury.

Säljö, R. (1979). Learning in the learners perspective: I. Some common sense conceptions (Technical Report No. 76). Gothenburg, Sweden: Institute of Education, University of Gothenburg.

Streefland, L. (1982). Subtracting fractions with different denominators. *Educational Studies in Mathematics*, *13*(3), 233-255.

Taylor, R. G. (1998). *Models of computation and formal languages*. New York: Oxford University Press.

Waddington, T. (1995). *Understanding as connected knowledge*. Unpublished PhD dissertation. Chicago, IL: University of Chicago.

Zhang, D., Li, S., & Tang, R (2004). The "two basics": Mathematics teaching and learning in mainland China. In L. Fan, N. Y. Wong, J. Cai, & S. Li (Eds.). *How Chinese learn mathematics: Perspectives from insiders* (pp. 189-207). Singapore: World Scientific.