Stanford Department of Computer Science
Report No. STAN-CS-80-786

'January 1980

ALGORITHMS IN MODERN MATHEMATICS AND COMPUTER SCIENCE

by

Donald E. Knuth

Research sponsored by

National Science Foundation
and
Office of Naval Research

COMPUTER SCIENCE DEPARTMENT
Stanford University

Computer Science Department
Report No. STAN-CS-80-786

January 1980

ALGORITHMS IN MODERN MATHEMATICS AND COMPUTER SCIENCE

by Donald E. Knuth

The life and work of the ninth century scientist al-Khwārizmī, "the father of algebra and algorithms," is surveyed briefly. Then a random sampling technique is used in an attempt to better understand the kinds of thinking that good mathematicians and computer scientists do and to analyze whether such thinking is significantly "algorithmic" in nature. (This is the text of a talk given at the opening session of a symposium on "Algorithms in Modern Mathematics and Computer Science" held in Urgench, Khorezm Oblast', Uzbek S.S.R., during the week of September 16-22, 1979.)

The preparation of this report was supported in part by National Science Foundation grant MCS72-03752 A03 and in part by Office of Naval Research contract N0014-76-C-0330. Reproduction in whole or in part is permitted for any purpose of the United States government. The author wishes to thank his Uzbek hosts for their incomparable hospitality.

ALGORITHMS IN MODERN MATHEMATICS AND COMPUTER SCIENCE

by Donald *E.* Knuth

My purpose in this paper is to stimulate discussion about a philosophical question that has been on my mind for a long time: What is the actual role of the notion of an algorithm in mathematical sciences?

For many years I have been convinced that computer science is primarily the study of algorithms. My colleagues don't all agree with me, but it turns out that the source of our disagreement is simply that my definition of algorithms is much broader than theirs: I tend to think of algorithms as encompassing the whole range of concepts dealing with well-defined processes, including the structure of data that is being acted upon as well as the structure of the sequence of operations being performed; some other people think of algorithms merely as miscellaneous methods for the solution of particular problems, analogous to individual theorems in mathematics.

In the U.S.A., the sorts of things my colleagues and I do is called Computer Science, emphasizing the fact that algorithms are performed by machines. But if I lived in Germany or France, the field I work in would be called *Informatik* or *Informatique*, emphasizing the stuff that algorithms work on more than the processes themselves. In the Soviet Union, the same field is now known as either *Kibernetika* (Cybernetics), emphasizing the control of a process, or *Prikladnaĭa Matematika* (Applied Mathematics), emphasizing the utility of the subject and its ties to mathematics in general. I suppose the name of our discipline isn't of vital importance, since we will go on doing what we are doing no matter what it is called; after all, other disciplines like Mathematics and Chemistry are no longer related very strongly to the etymology of their names. However, if I had a chance to vote for the name of my own discipline, I would choose to call it Algorithmics.

The site of our symposium is especially well suited to philosophical discussions such as I wish to incite, both because of its rich history and because of the grand scale of its scenery. This is an ideal time for us to consider the long range aspects of our work, the issues that we usually have no time to perceive in our hectic everyday lives at home. During the coming week we will have a perfect opportunity to look backward in time to the roots of our subject, as well as to look ahead and to contemplate what our work is all about.

I have wanted to make a pilgrimage to this place for many years, ever since learning that the word "algorithm" was derived from the name of al-Khwârizmî, the great ninth-century scientist whose name means "from Khwarizm." The modern Spanish word *guarismo* ("digit") also stems from his name. Khwarizm was not simply a notable city (Khiva) as many Western authors have thought, it was (and still is) a rather large district. In fact, the Aral Sea was at one time known as Lake Khwarizm (see, for example, [**17**, Plates 9–21]). By the time of the conversion of this region to Islam in the seventh century, a high culture had developed, having for example its own script and its own calendar (cf. al-Bîrûnî [**21**]).

Catalog cards prepared by the U.S. Library of Congress say that al-Khwârizmî flourished between 813 and 846 A.D. It is amusing to take the average of these two numbers, obtaining 829.5, almost exactly 1150 years ago. Therefore we are here at an auspicious time, to celebrate an undesesquicentennial.

Comparatively little is known for sure about al-Khwarizmi's life. His full Arabic name is essentially a capsule biography: Abu Ja'far Muhammad ibn Mûsâ al-Khwârizmî, meaning "Mohammed, father of Jafar, son of Moses, the Khwârizmîan." However, the name does not prove that he was born here, it might have been his ancestors instead of himself, We do know that his scientific work was done in Baghdad, as part of an academy of scientists called the "House of Wisdom," under Caliph al-Ma'mûn. Al-Ma'mûn was a great patron of science who invited many learned men to his court in order to collect and extend the wisdom of the world. In this respect he was building on foundations laid by his predecessor, the Caliph Harûn al-Rashid, who is familiar to us because of the Arabian Nights. The historian al-Tabari added "al-Qutrubbulli" to al-Khwârizmî's name, referring to the Qutrubbull district near Baghdad. Personally I think it is most likely that al-Khwârizmî was born in Khwarizm and lived most of his life in Qutrubbull after being summoned to Baghdad by the Caliph, but the truth will probably never be known.

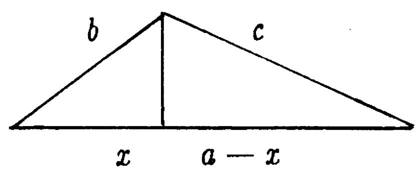
The Charisma of al-Khwârizmî.

In any event it is clear that al-Khwârizmî's work had an enormous influence throughout the succeeding generations. According to the *Fihrist*, a sort of "Who's Who" and bibliography of 987 A.D., "during his lifetime and afterwards, people were accustomed to rely upon his tables." Several of the books he wrote have apparently vanished, including a historical Book of Chronology and works on the sundial and the astrolabe. But he compiled a map of the world (still extant) giving coordinates for cities, mountains, rivers, and coastlines; this was the most complete and accurate map that had ever been made up to that time. He also wrote *a* short treatise on the Jewish calendar, and compiled extensive astronomical tables that were in wide use for several hundred years. (But nobody is perfect: Some modern scholars feel that these tables were not as accurate as they could have been.)

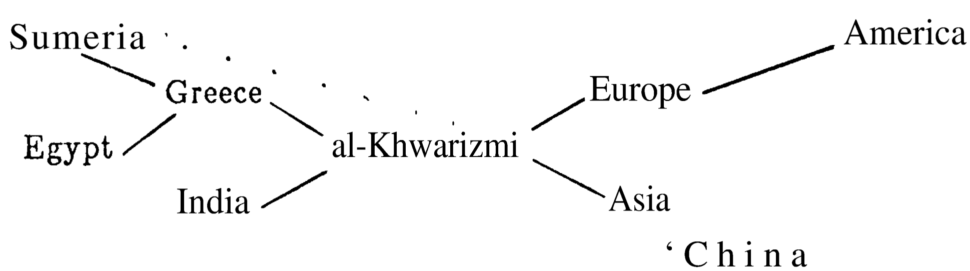
The most significant works of al-Khwârizmî were almost certainly his textbooks on algebra and arithmetic, which apparently were the first Arabic writings to deal with such topics. His algebra book was especially famous; in fact, at least three manuscripts of this work in the original Arabic are known to have survived to the present day, while more than 99% of the books by other authors mentioned in the *Fihrist* have been lost. Al-Khwârizmî's Algebra was translated into Latin at least twice during the twelfth century, and this is how Europeans learned about the subject. In fact, our word "algebra" stems from part of the Arabic title of this book, *Kitâb* al-jabr *wa'l-muqâbala*, "The Book of Aljabr and Almuqâbala." (Historians disagree on the proper translation of this title. My personal opinion, based on a reading of the work and on the early Latin translation restaurationis et *oppositionis* [3, p.2], together with the fact that *muqâbala* signifies some sort of standing face-to-face, is that it would be best to call al-Khwarizmi's algebra "The Book of Restoring and Equating.")

We can get some idea of the reasons for al-Khwarizmi's success by looking at his *Algebra* in more detail. The purpose of the book was not to summarize all knowledge of the subject, but rather to give the "easiest and most useful" elements, the kinds of mathematics most often needed. He discovered that the complicated geometric tricks previously used in Babylonian and Greek mathematics could be replaced by simpler and more systematic methods that rely on algebraic manipulations alone. Thus the subject became accessible to a much wider audience. He explained how to reduce all nontrivial quadratic equations to one of three forms that we would express as $x^2 + bx = c$, $x^2 = bx + c$, $x^2 + c = bx$ in modern notation, where b and c are positive numbers; note that he has gotten rid of the coefficient of x^2 by dividing it out. If he had known about negative numbers, he would have been delighted to go further and reduce these three possibilities to a single case.

I mentioned that the Caliph wanted his scientists to put the existing scientific knowledge of other lands into Arabic texts. Although no prior work is known to have incorporated al-Khwarizmi's elegant approach to quadratic equations, the second part of his *Algebra* (which deals with questions of geometric measurements) was almost entirely based on an interesting treatise called the *Mishnat ha-Middot*, which Solomon Gandz has given good reason to believe was composed by a Jewish rabbi named Nehemiah about 150 A.D. [4]. The differences between the *Mishnat* and the Algebra help us to understand al-Khwarizmi's methods. For example, when the Hebrew text said that the circumference of a circle is $3\frac{1}{7}$ times the diameter, al-Khwarizmi added that this is only a conventional approximation, not a proved fact; he also mentioned $\sqrt{10}$ and $\frac{62832}{20000}$ as alternatives, the latter "used by astronomers." The Hebrew text merely stated the Pythagorean theorem, but al-Khwarizmi appended a proof. Probably the most significant change occurred in his treatment of the area of a general triangle: The Mishnat simply states Heron's formula $\sqrt{s(s-a)(s-b)(s-c)}$ where $s = \frac{1}{2}(a+b+c)$ is the semiperimeter, but the *Algebra* takes an entirely different tack. Al-Khwarizmi wanted to reduce the number of basic operations, so he showed how to compute the area in general from the simpler formula $\frac{1}{2}(\text{base} \times \text{height})$, where the height could be computed by simple algebra. Let the perpendicular to the largest side of the triangle from the opposite corner strike the longest side at a distance x from its end; then $b^2-x^2 = c^2-(a-x)^2$, hence $b^2 = c^2-a^2+2ax$ and $x = (a^2+b^2-c^2)/(2a)$. The height of the triangle can now be computed as $\sqrt{b^2-x^2}$; thus it isn't necessary to learn Heron's trick.



Unless an earlier work turns up showing that al-Khwarizmi learned his approach to algebra from somebody else, these considerations show that we are justified in calling him "the father of algebra." In other words, we can add the phrase "abu-aljabr" to his name! The overall history of the subject can be diagrammed roughly thus:



(I have shown a dotted line from Sumeria to represent a plausible connection between ancient traditions that might have reached Baghdad directly instead of via Greece. Conservative scholars doubt this connection, but I think they are too much influenced by old-fashioned attitudes to history in which Greek philosophers were regarded as the source of all scientific knowledge.) Of course, al-Khwârizmi never took the subject beyond quadratic equations in one variable, but he did make the important leap away from geometry to abstract reckoning, and he made the subject systematic and reasonably simple for practical use. (He was unaware of Diophantus's prior work on number theory, which was even more abstract and removed from reality, therefore closer to modern algebra. It is difficult to rank either al-Khwârizmî or Diophantus higher than the other, since they had such different aims. The unique contribution of Greek scientists was their pursuit of knowledge solely for its own sake.)

The original Arabic version of al-Khwârizmî 's small book on what he called the Hindu art of reckoning seems to have vanished. Essentially all we have is an incomplete 13th-century copy of what is a probably a 12th-century translation from Arabic into Latin; the original Arabic may well have been considerably different. It is amusing to look at this Latin translation, because it is primarily a document about how to calculate in Hindu numerals (the decimal system) but it uses Roman numerals to express numbers! Perhaps al-Khwarizmi 's original treatise was similar in this respect, except that he would have used the alphabetic notation for numbers adapted from earlier Greek and Hebrew sources to Arabic; it is natural to expect that the first work on the subject would state problems and their solutions in an old familiar notation. I suppose the new notation became well known shortly after al-Khwârizmî's book appeared, and that might be why no copies of his original are left.

The Latin translation of al-Khwarizmi's arithmetic has blank spaces where most of the Hindu numerals were to be inserted; the scribe never got around to this, but it is possible to make good guesses about how to fill in these gaps: The portion of the manuscript that survives has never yet been translated from Latin to English or any other Western language, although a Russian translation appeared in 1964 [16]. Unfortunately both of the published transcriptions of the Latin handwriting ([3],[27]) are highly inaccurate; see [18]. It would clearly be desirable to have a proper edition of this work in English, so that more readers can appreciate its contents. The algorithms given for decimal addition, subtraction, multiplication, and division—if we may call them algorithms, since they omit many details, even though they were written by al-Khwarizmi himself!--have been studied by Îushkevich [9] and Rosenfel'd [16]. They are interesting because they are comparatively unsuitable for pencil-and-paper calculation, requiring lots of crossing-out or erasing; it seems clear that they are merely straightforward adaptations of procedures that were used on an abacus of some sort, in India if not in Persia. The development of methods more suitable for non-abacus calculations seems to be due to al-Uqlidisi in Damascus about two centuries later [22].

Further details of al-Khwârizmî's works appear in an excellent article by G.

J. Toomer in the Dictionary *of* Scientific Biography [26]. This is surely the most comprehensive summary of what is now known about Muhammad ibn Mûsâ, although I was surprised to see no mention of the plausible hypothesis that local traditions continued from Babylonian times to the Islamic era.

Before closing this historical introduction, I want to mention another remarkable man from Khwârizm, namely, Abû Rayḥân Muhammad ibn Aḥmad al-Bîrûnî (973-1048 A.D.): philosopher, historian, traveler, geographer, linguist, mathematician, encyclopedist, astronomer, poet, physicist, and computer scientist, author of an estimated 150 books [12]. I have put "computer scientist" in this list because of his interest in efficient calculation. For example, al-Bîrûnî showed how to evaluate the sum $1 + 2 + \ldots + 2^{63}$ of the number of grains of wheat on a chessboard if a single grain is placed on the first square, two on the second, twice as many on the third, etc.: using a technique of divide and conquer, he proved that the total is $\left(((16^2)^2)^2\right)^2 - 1$, and he gave the answer 18,446,744,073,709,551,615 in three systems of notation (decimal, sexagesimal, and a peculiar alphabetic-Arabic). He also pointed out that this number amounts to approximately 2305 "mountains," if one mountain equals 10000 wadis, one wâdî is 1000 herds, one herd is 10000 loads, one load is 8 bidar, and one bidar is 10000 units of wheat [20; *21,* pp. 132-136; 23].

Some Questions.

Will Durant has remarked that "scholars were as numerous as the pillars, in thousands of mosques," during that golden age of medieval science. Now here we are, a group of scholars with a chance to be inspired by the same surroundings; and I would like to raise several questions that I believe are important today. *What* is the *relation of* algorithms to modern *mathematics?* Is there an essential difference between an algorithmic viewpoint and the traditional mathematical world-view? *Do* most mathematicians have an essentially *different* thinking process *from that of* most computer scientists? Among members of university mathematics departments, why do the logicians (and to a lesser extent the combinatorial mathematicians) tend to be much *more* interested in computer science than their colleagues?

I raise these questions partly because of my own experiences as a student. I began to study higher mathematics in 1957, the same year that I began to work with digital computers, but I never mixed my mathematical thinking with my computer-science thinking in nontrivial ways until 1961. In one building I was a mathematician, in another I was a computer programer, and it was as if I had a split personality. During 1961 I was excited by the idea that mathematics and computer science might have some common ground, because BNF notation looked mathematical, so I bought a copy of Chomsky's Syntactic Structures and set out to find an algorithm to decide the ambiguity problem of context-free grammars (not knowing that this had been proved impossible by Bar-Hillel, Perles, and Shamir in 1960). I failed to solve that problem, although I found some useful necessary and sufficient conditions for ambiguity, and I also derived a few other results like the fact that context-free languages on one letter are regular. Here, I thought, was a nice mathematical theory that I was able to develop with my computer-science intuition; how curious! During the summer of 1962, I spent a day or two analyzing the performance of hashing with linear probing, but this did not really seem like a marriage between my computer science personality and my mathematical personality since it was merely an application of combinatorial mathematics to a problem that has relevance to programming.

I think it is generally agreed that mathematicians have somewhat different thought processes from physicists, who have somewhat different thought processes from chemists, who have somewhat different thought processes from biologists. Similarly, the respective "mentalities" of lawyers, poets, playwrights, historians, linguists, farmers, and so on, seem to be unique. Each of these groups can probably recognize that other types of people have a different approach to knowledge; and it seems likely that a person gravitates to the particular kind of occupation that corresponds to the mode of thought that he or she grew up with, whenever a choice is possible. C. P. Snow wrote a famous book about "two cultures," scientific vs. humanistic, but in fact there seem to be many more than two.

Educators of computer science have repeatedly observed that only about 2 out of every 100 students enrolling in introductory programing courses really "resonate" with the subject and seem to be natural-born computer scientists. (For example, see Gruenberger [8].) Just last week I had some independent confirmation of this, when I learned that 220 out of 11000 graduate students at the University of Illinois are majoring in Computer Science. Since I believe that Computer Science is the study of algorithms, I conclude that roughly 2% of all people "think algorithmically," in the sense that they can rapidly reason about algorithmic processes.

While writing this paper, I learned about some recent statistical data gathered by Gerrit DeYoung, a psychologist-interested-in-computer-science whom I met at the University of Illinois. He had recently made an interesting experiment on two groups of undergraduate students taking introductory courses in computer science. Group I consisted of 135 students intending to major in computer science, while Group II consisted of 35 social science majors. Both courses emphasized non-numeric programming and various data and control structures, although numerical problems were treated too. DeYoung handed out a questionnaire that tested each student's so-called quantitative aptitude, a standard test that seems to correlate with mathematical ability, and he also asked them to estimate their own performance in class, Afterwards he learned the grades that the students actually did receive, so he had three pieces of data on each student:

$A =$ quantitative aptitude;

$B =$ student's own perception of programming ability;

$C =$ teacher's perception of programming ability.

In both cases *B* correlated well with C (the coefficient was about .6), so we can conclude that the teachers' grading wasn't random and that there is some validity in these scores. The interesting thing was that there was no correlation between *A* and *B* or between A and C among the computer science majors (Group I), while there was a pronounced correlation of about .4 between the corresponding numbers for the students of Group II. It isn't clear how to interpret this data, since many different hypotheses could account for such results; perhaps psychologists know only how to measure the quantitative ability of people who think like psychologists do! At any rate the lack of correlation between quantitative ability and programming performance in the first group reminds me strongly of the feelings I often have about differences between mathematical thinking and computer-science thinking, so further study is indicated.

I believe that the real reason underlying the fact that Computer Science has become a thriving discipline at essentially all of the world's universities, although it was totally unknown twenty years ago, is not tbat computers exist in quantity; the real reason is that the algorithmic thinkers among the scientists of the world never before had a home. We are brought together in Computer Science departments because we find people who think like we do.

At least, that seems a viable hypothesis, which hasn't been contradicted by my observations during the last half dozen or so years since the possibility occurred to me.

My goal, therefore, is to get a deeper understanding of these phenomena; the "different modes of thought" hypothesis merely scratches the surface. Can we come up with a fairly clear idea of just what algorithmic thinking is, and contrast it with classical mathematical thinking?

At times when I try to come to grips with this question, I find myself almost convinced that algorithmic thinking is really like mathematical thinking, only it concentrates on more "difficult" things. But at other times I have just the opposite impression, that somehow algorithms hit only the "simpler" kinds of mathematics.. , . Clearly such an approach leads only to confusion and gets me nowhere.

While pondering these things recently, I suddenly remembered the collection of expository works called Mathematics: Its Content, Methods, *and* Meaning

[1], so I reread what A. D. Aleksandrov said in his excellent introductory essay. Interestingly enough, I found that he makes prominent mention of al-Khwarizmi. Aleksandrov lists the following characteristic features of mathematics:

- Abstractness, with many levels of abstraction.
- Precision and logical rigor.
- Quantitative relations.
- Broad range of applications.

Unfortunately, all four of these features seem to be characteristic also of computer science; is there really no difference betwen computer science and mathematics?

A Plan.

I decided that I could make no further progress unless I took a stab at analyzing the question "What is mathematics?"-analyzing it in some depth. The answer, of course, is that "Mathematics is what mathematicians do." More precisely, the appropriate question should probably be, "What is good mathematics?" and the answer is that "Good mathematics is what good mathematicians do."

Therefore I took nine books off of my shelf, mostly books that I had used as texts during my student days but also a few more for variety's sake. I decided to look at page 100 (i.e., a "random" page) in each book and to study the first result on that page. This way I could get a sample of what good mathematicians do, and I could attempt to understand the types of thinking that seem to be involved.

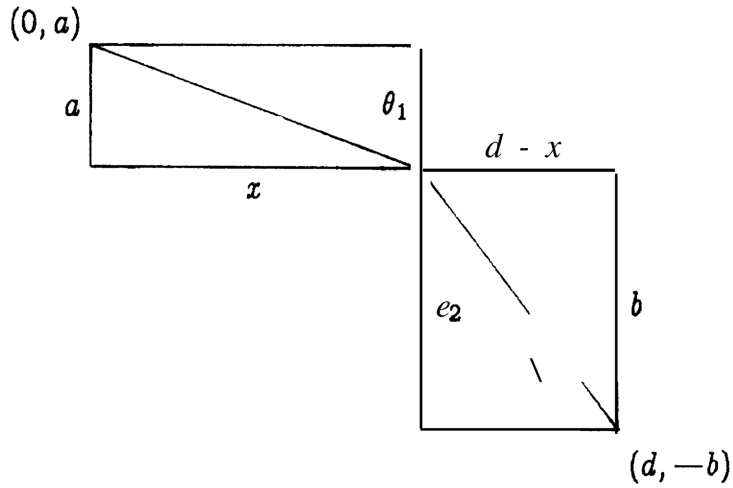
From the standpoint of computer science, the notion of "types of thinking" is not so vague as it once was, since we can now imagine trying to make a computer program discover the mathematics. What sorts of capabilities would we have to put into such an artificially intelligent program, if it were to be able to come up with the results on page 100 of the books I selected?

In order to make this experiment fair, I was careful to abide by the following ground rules: (1) The books were all to be chosen first, before I studied any particular one of them. (2) Page 100 was to be the page examined in each case, since I had no a priori knowledge of what was on that page in any book. If somehow page 100 turned out to be a bad choice, I wouldn't try anything sneaky like searching for another page number that would give results more in accord with my prejudices. (3) I would not suppress any of the data; every book I had chosen would appear in the final sample, so that I wouldn't introduce any bias by selecting a subset.

The results of this experiment opened up my eyes somewhat, so I would like to share them with you. Here is a book-by-book summary of what I found.

Book 1: Thomas's Calculus.

I looked first at the book that first introduced me to higher mathematics, the calculus text by George B. Thomas [25] that I had used as a college freshman. On page 100 he treats the following problem: What value of x minimizes the travel time from $(0, a)$ to $(x, 0)$ to $(d, -b)$, *if you* must go at speed s_1 *from* $(0, a)$ *to* $(x, 0)$ *and at speed* s_2 *from* $(x, 0)$ *to* $(d, -b)$*?*



In other words, we want to minimize the function

$$f(x) = \sqrt{a^2 + x^2}/s_1 + \sqrt{b^2 + (d - x)^2}/s_2.$$

The solution is to differentiate $f(x)$, obtaining

$$f'(x) = \frac{x}{s_1\sqrt{a^2 + x^2}} - \frac{d - x}{s_2\sqrt{b^2 + (d - x)^2}} = \frac{\sin\theta_1}{s_1} - \frac{\sin\theta_2}{s_2}.$$

As x runs from 0 to d, the value of $(\sin\theta_1)/s_1$ starts at zero and increases, while the value of $(\sin\theta_2)/s_2$ decreases to zero. Therefore the derivative starts negative and ends positive; there must be a point where it is zero, i.e., $(\sin\theta_1)/s_1 = (\sin\theta_1)/s_2$, and that's where the minimum occurs. Thomas remarks that this is "Snell's Law" in optics; somehow light rays know how to minimize their travel time.

The mathematics involved here seems to be mostly a systematic procedure for minimization, based on formula manipulation and the correspondence between formulas and geometric figures, together with some reasoning about changes in function values. Let's keep this in mind as we look at the other examples, to see how much the examples have in common.

Book 2: A Survey of Mathematics.

Returning to the survey volumes edited by Aleksandrov et al. [1], we find that page 100 is the chapter on Analysis by Lavrent'ev and Nikol'skiĭ. It shows how to deduce the derivative of the function $\log_a x$ in a clever way:

$$\frac{\log_a(x+h)-\log_a x}{h}=\frac{1}{h}\log_a\frac{x+h}{x}=\frac{1}{x}\log_a\left(1+\frac{h}{x}\right)^{x/h}.$$

The logarithm function is continuous, so we have

$$\lim_{h\to 0}\frac{1}{x}\log_a\left(1+\frac{h}{x}\right)^{h/x}=\frac{1}{x}\log_a\lim_{h\to 0}\left(1+\frac{h}{x}\right)^{x/h}=\frac{1}{x}\log_a e,$$

since it has already been established that $(1+\frac{1}{n})^n$ approaches a constant called e when n approaches infinity through integer or noninteger values. Here the reasoning involves formula manipulation and an understanding of limiting processes.

Book 3: Kelley's General Topology.

The third book I chose was a standard topology text [10], where page 100 contains the following exercise: *"Problem A. The image under a continuous map* of a connected space is connected." No solution is given, but I imagine something like the following was intended: First we recall the relevant definitions, that a function f from topological space X to topological space Y is continuous when the inverse image $f^{-1}(V)$ is open in X, for all open sets V in Y; a topological space X is connected when it cannot be written as a union of two nonempty open sets. Thus, let us try to prove that Y is connected, under the assumption that f is continuous and X is connected, where j(X) = Y. If $Y = V_1 \cup V_2$, where V_1 and V_2 are disjoint and open, then $X = f^{-1}(V_1) \cup f^{-1}(V_2)$, where $f^{-1}(V_1)$ and $f^{-1}(V_2)$ are disjoint and open. It follows that $f^{-1}(V_1)$ or $f^{-1}(V_2)$ is empty, say $f^{-1}(V_1)$ is empty. Finally, therefore, V_1 is empty, since $V_1 \subseteq f(f^{-1}(V_1))$. Q.E.D.

(Note that no properties of "open sets" were needed in this proof.)

The mathematical thinking involved here is somewhat different from what we have seen before; it consists primarily of constructing chains of implications from the hypotheses to the desired conclusions, using a repertoire of facts like "$f^{-1}(A \cap B) = f^{-1}(A) \cap f^{-1}(B)$". This is analogous to constructing chains of computer instructions that transform some input into some desired output, using a repertoire of subroutines, although the topological facts have a more abstract charater.

Another type of mathematical thinking is involved here, too, and we should be careful not to forget it: Somebody had to define the concepts of continuity and connectedness in some way that would lead to a rich theory having lots of applications. This generalizes many special cases that had been proved before the abstract pattern was perceived.

Book 4: From the 18th Century.

Another book on my list was Struik's Source Book in Mathematics, which quotes authors of famous papers written during the period 1200-1800 A.D. Page 100 is concerned with Euler's attempt to prove the fundamental theorem of algebra, in the course of which he derived the following auxiliary result: "Theorem 4. Every quartic polynomial $x^4 + Ax^3 + Bx^2 + Cx + D$ with real coefficients can be factored into two quadratics."

Here's how he did it. First he reduced the problem to the case $A = 0$ by setting $x = y - \frac{1}{4}A$. Then he was left with the problem of solving $(x^2 + ux + \alpha) \times (x^2 - ux + \beta) = x^4 + Bx^2 + Cx + D$ for u, α, and β, so he wanted to solve the equations $B = \alpha + \beta - u^2$, $C = (\beta - \alpha)u$, $D = \alpha\beta$. These equations lead to the relations $2\beta = B + u^2 + C/u$, $2\alpha = B + u^2 - C/u$, and $(B+u^2)^2 - C^2/u^2 = 4D$. But the cubic polynomial $(u^2)^3 + 2B(u^2)^2 + (B^2 - 4D)u^2 - C^2$ goes from $-C^2$ to $+\infty$ as u^2 runs from 0 to ∞, so it has a positive root, and the factorization is complete.

(Euler went on to generalize, arguing that every equation of degree 2^n can be-factored into two of degree 2^{n-1}, via an equation of odd degree $\frac{1}{2}\binom{2^n}{2^{n-1}}$ in u^2 having a negative constant term. But this part of his derivation was not rigorous; Lagrange and Gauss later pointed out a serious flaw.)

When I first looked at this example, it seemed to be more "algorithmic" than the preceding ones, probably because Euler was essentially explaining how to take a quartic polynomial as input and to produce two quadratic polynomials as output. Input/output characteristics are significant aspects of algorithms, although Euler's actual construction is comparatively simple and direct so it doesn't exhibit the complex control structure that algorithms usually have. The types of thinking involved here seem to be (a) to reduce a general problem to a simpler special case (by showing that A can be assumed zero, and by realizing that a sixth-degree equation in u was really a third-degree equation in u^2); (b) formula manipulation to solve simultaneous equations for a, β, and u; (c) generalization by recognizing a pattern for the case of 4th degree equations that apparently would extend to degrees 8, 16, etc.

Book 5: Abstract Algebra.

My next choice was another standard textbook, Commutative Algebra by Zariski and Samuel [28]. Their page 100 is concerned with the general structure of arbitrary fields. Suppose k and K are fields with $k \subseteq K$; the transcendence degree of K over k is defined to be the cardinal number of any "transcendence basis" L of K over k, namely a set L such that all of its finite subsets are algebraically independent over k and such that all elements of K are algebraic over $k(L)$; i.e., they are roots of polynomial equations whose coefficients are in the smallest field containing $k \cup L$. The exposition in the book has just found that this cardinal number is a well-defined invariant of k and K, i.e., that all transcendence bases of K over k have the same cardinality.

Now comes Theorem 26: *If $k \subseteq K \subseteq \mathcal{K}$, the transcendence degree of $\mathcal{K}$ over k is the sum of the* transcendence degrees of K over k and of $\mathcal{K}$ over K. To prove the theorem, Zariski and Samuel let L be a transcendence basis of K over k and $\mathcal{L}$ a transcendence basis of $\mathcal{K}$ over K; the idea is to prove that $L \cup \mathcal{L}$ is a transcendence basis of $\mathcal{K}$ over k, and the result follows since L and $\mathcal{L}$ are disjoint.

The required proof is not difficult and it is worth studying in detail. Let $\{x_1, \ldots, x_m, X_1, \ldots, X_M\}$ be a finite subset of $L \cup \mathcal{L}$, where the x's are in L and the X's in $\mathcal{L}$, and assume that they satisfy some polynomial equation over k, namely

$$\sum_{\substack{e_1,\ldots,e_m \geq 0 \\ E_1,\ldots,E_M \geq 0}} a(e_1, \ldots, e_m, E_1, \ldots, E_M)\, x_1^{e_1} \ldots x_m^{e_m} X_1^{E_1} \ldots X_M^{E_M} = 0 \qquad (*)$$

where all the $a(e_1, \ldots, e_m, E_1, \ldots, E_M)$ are in k and only finitely many a's are nonzero. This equation can be rewritten as

$$\sum_{E_1,\ldots,E_M \geq 0} \Bigl(\sum_{e_1,\ldots,e_m \geq 0} a(e_1, \ldots, e_m, E_1, \ldots, E_M)\, x_1^{e_1} \ldots x_m^{e_m} \Bigr) X_1^{E_1} \ldots X_M^{E_M} = 0, \qquad (**)$$

a polynomial in the X's with coefficients in K, hence all of these coefficients are zero by the algebraic independence over $\mathcal{L}$ over K. These coefficients in turn are polynomials in the x's with coefficients in k, so all the α's must be zero. In other words, any finite subset of $L \cup \mathcal{L}$ is algebraically independent.

Finally, all elements of K are algebraic over $k(L)$ and all elements of $\mathcal{K}$ are algebraic over $K(\mathcal{L})$. It follows from the previously developed theory of algebraic extensions that all elements of $\mathcal{K}$ are algebraic over $k(L)(\mathcal{L})$, the smallest field containing $k \cup L \cup \mathcal{L}$. Hence $L \cup \mathcal{L}$ satisfies all the criteria of a transcendence basis.

Note that the proof involves somewhat sophisticated "data structures," i.e., representations of complex objects, in this case polynomials in many variables. The key idea is a *pun,* the equivalence between the polynomial over k in $(*)$ and the polynomial over $k(L)$ in $(**)$. In fact, the structure theory of fields being developed in this part of Zariski and Samuel's book is essentially a theory about data structures by which all elements of the field can be manipulated. Theorem 26 is not as important as the construction of transcendence bases that appears in its proof.

Another noteworthy aspect of this example is the way infinite sets are treated. Finite concepts have been generalized to infinite ones by saying that all finite subsets must have the property; this allows algorithmic constructions to be applied to the subsets.

Book 6: Metamathematics.

I chose Kleene's *Introduction to Metamathematics* [13] as a representative book on logic. Page 100 talks about "disjunction elimination": Suppose we are given (1) $\vdash A \vee B$ and (2) $A \vdash C$ and (3) $B \vdash C$. Then by a rule that has just been proved, (2) and (3) yield

$$(4)\quad A \vee B \vdash C.$$

From (1) and (4) we may now conclude "(5) $\vdash C$". Kleene points out that this is the familiar idea of reasoning by cases. If either A or B is true, we can consider case 1 that A is true (then C holds); or case 2 that B is true (and again C holds). Hence C holds in any case.

The reasoning in this example is simple formula manipulation, together with an understanding that familiar thought patterns are being generalized and made formal.

I was hoping to hit a more inherently metamathematical argument here, something like "anything that can be proved in system X can also be proved in system Y," since such arguments are often essentially algorithms that convert arbitrary X-proofs into Y-proofs. But page 100 was more elementary, this being an introductory book.

Book 7: **Knuth.**

3s my own work [14] algorithmic? Well, page 100 isn't especially so, since it is part of the introduction to mathematical techniques that appear before I get into the real computer science content. The problem discussed on that page is to get the mean and standard deviation of the number of "heads" in n coin flips, when each independent flip comes up "heads" with probability p and "tails" with probability $q = 1 - p$. I introduce the notation p_{nk} for the probability that k heads occur, and observe that

$$p_{nk} = p \cdot p_{n-1,k-1} + q \cdot p_{n-1,k}.$$

To solve this recurrence, I introduce the generating function

$$G_n(z) = \sum_{k\geq 0} p_{nk} z^k$$

and obtain G,(z) $= (q+pz)G_{n-1}(z)$, $G_1(z) = q+pz$. Hence $G_n(z) = (q+pz)^n$, and

$$\mathrm{mean}(G_n) = n\,\mathrm{mean}(G_1) = pn; \qquad \mathrm{var}(G_n) = n\,\mathrm{var}(G_1) = pqn.$$

Thus, the recurrence relation is set up by reasoning about probabilities; it is solved by formula manipulation according to patterns that are discussed earlier in the book. I like to think that I was being like al-Khwarizmi here-not using a special trick for this particular problem, rather illustrating a general method.

Book 8: **Pólya** and **Szegö.**

The good old days of mathematics are represented by Pólya and Szegö's famous *Aufgaben und Lehrsätze*, recently available in an English translation with many new Aufgaben [19]. Page 100 contains a real challenge:

$$217.\ \lim_{n\to\infty} \int_{-\pi}^{\pi} \frac{n!\,2^{2n\cos\theta}}{|(2ne^{i\theta}-1)\ldots(2ne^{i\theta}-n)|}\,d\theta = 2\pi.$$

Fortunately the answer pages provide enough of a clue to reveal the proof that they had in mind. We have $|2ne^{i\theta} - k|^2 = 4n^2 + k^2 - 4nk\cos\theta = (2n-k)^2 + 4nk(1-\cos\theta) = (2n-k)^2 + 8nk\sin^2\theta/2$. Replacing θ by $x/\sqrt{n}$ allows us to rewrite the integral as

$$\frac{n!\,2^{2n}}{((2n-1)\ldots n)\sqrt{n}}\int_{-\infty}^{\infty} f_n(x)\,dx,$$

where $f_n(x) = 0$ for $|x| > \pi\sqrt{n}$, and otherwise

$$f_n(x) = 2^{2n\,(\cos x/\sqrt{n}\,-\,1)} \prod_{1\le k\le n}\left(\frac{1}{1+\frac{8nk}{(2n-k)^2}\sin^2\frac{x}{2\sqrt{n}}}\right)^{1/2}$$

$$= \exp\left((2\ln 2)n\left(\cos\frac{x}{\sqrt{n}} - 1\right) + \sum_{1\le k\le n}\frac{1}{2}\ln\left(\frac{1}{1+\frac{8nk}{(2n-k)^2}\sin^2\frac{x}{2\sqrt{n}}}\right)\right)$$

$$= \exp\left(-x^2\ln 2 + O\left(\frac{x^4}{n}\right) + \frac{1}{2}\sum_{1\le k\le n}\left(\frac{-2nk}{(2n-k)^2}\frac{x^2}{n} + O\left(\frac{x^4}{n^2}\right)\right)\right)$$

$$= \exp\left(-x^2\ln 2 - (1-\ln 2)x^2 + O\left(\frac{1+x^4}{n}\right)\right).$$

Thus, $f_n(x)$ converges uniformly to e^{-x^2} in any bounded internal. Furthermore $|f_n(x)| \le 2^{2n(\cos x/\sqrt{n}\,-\,1)}$ and

$$\cos\frac{x}{\sqrt{n}} - 1 \le -\frac{x^2}{2n} + \frac{x^4}{24n^2}$$

$$\le -\left(\frac{1}{2} - \frac{\pi^2}{24}\right)\frac{x^2}{n} \qquad \text{for } |x| \le \pi/\sqrt{n},$$

since the cosine function is "enveloped" by its Maclaurin series; therefore $|f_n(x)|$ is less than the integrable function e^{-cx^2} for all n, where $c = 1 - \pi^2/12$. From this uniformly bounded convergence we are justified in taking limits past the integral sign,

$$\lim_{n\to\infty}\int_{-\infty}^{\infty} f_n(x)\,dx = \int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}.$$

Finally, the coefficient in front of $\int_{-\infty}^{\infty} f_n(x)\,dx$ is $2^{2n+1}n!^2/\sqrt{n}(2n)!$, which is $2\sqrt{\pi}\bigl(1+O(1/n)\bigr)$ by Stirling's approximation, and the result follows.

This derivation gives some idea of how far mathematics had developed between the time of al-Khwârizmî and 1920. It involves formula manipulation and an understanding of the asymptotic limiting behavior of functions, together with the idea of inventing a suitable function f_n that will make the interchange $\lim_{n\to\infty}\int_{-\infty}^{\infty} f_n(x)\,dx = \int_{-\infty}^{\infty}\bigl(\lim_{n\to\infty} f_n(x)\bigr)\,dx$ valid. The definition of $f_n(x)$ requires a clear understanding of how functions like exp x and cos x behave.

Book 9: Bishop's Constructive Mathematics.

The last book I chose to sample turned out to be most interesting of all from the standpoint of my quest; it was Errett Bishop's *Foundations of Constructive* Mathematics [2], a book that I had heard about but never before read. The interesting thing about this book is that it reads essentially like ordinary mathematics, yet it is entirely algorithmic in nature if you look between the lines.

Page 100 of Bishop's book contains Corollary 3 to the Stone-Weirstrass theorem developed on the preceding pages: Every uniformly continuous function *on a* compact set $X \subseteq$ R can be arbitrarily closely approximated *on X by polynomial* functions over R. And here is his proof: "By Lemma 5, the function $x \mapsto |x - x_0|$ can be arbitrarily closely approximated on X by polynomials. The theorem then follows from Corollary 2."

We might call this a compact proof! Before unwrapping it to explain what Lemma 5 and Corollary 2 are, I want to stress that the proof is essentially an algorithm; the algorithm takes any constructively given compact set X and continuous function j and tolerance ϵ as input, and it outputs a polynomial that approximates j to within ϵ on all points of X. Furthermore the algorithm operates on algorithms, since j is given by an algorithm of a certain type, and since real numbers are essentially algorithms themselves.

I will try to put Bishop's implicit algorithms into an explicit ALGOL-like form, although the capabilities of today's programming languages have to be stretched considerably to reflect his constructions. First let's consider Lemma 5, which states that for each $\epsilon > 0$ there exists a polynomial p : R $\to$ R such that $p(0) = 0$ and $\bigl||x| - p(x)\bigr| \le \epsilon$ for all $|x| \le 1$. Bishop's proof, which makes the lemma an algorithm, is essentially the following.

```
R polynomial procedure Lemma 5(real ε);
begin integer N;   R polynomial g, p;
N := suitable function of ε;
g(t) := 1 − Σ_{1≤n≤N} (1/2 choose n)(−1)^n t^n;
p(t) := g(−t^2) − g(1);
return p;
end.
```

Here N is computed large enough that $|g(t) - (1-t)^{1/2}| \leq \frac{1}{2}\epsilon$ for $0 \leq t \leq 1$.

The other missing component of the proof on page 100 is Corollary 2, which states that if X is any compact metric space and if G is the set of all functions $x \mapsto \rho(x, \text{xc})$, where $x_0 \in X$ and where $\rho(x, y)$ denotes the metric distance from x to y, then "$\mathcal{A}(G)$ is dense in C(X)," That is, all uniformly continuous real-valued functions on X can be approximated to arbitrarily high accuracy by functions obtained from the functions G by a finite number of operations of addition, multiplication, and multiplication by real numbers. As stated, Corollary 2 turns out to be false in the case that X contains only one point, since G and A(G) then consist only of the zero function. I noticed this oversight while trying to formulate his proof in an explicitly algorithmic way, but the defect is easily remedied.

For our purposes it is best to reformulate Corollary 2 in the following way: *"Let X bc a compact metric space containing at* least *two points, and let G be the set of* all *functions of the form $x \mapsto c\rho(x, x_0)$, where* c > 0 and $x_0 \in$ X. Then *G is a separating* family *over X."* I'll repeat his definition of separating family in a minute; first I want to mention his Theorem 7, the Stone-Weierstrass theorem whose proof I shall not discuss in detail, namely the fact that A(G) is dense in C(X) whenever G is a separating family of uniformly continuous functions over a compact metric space X. In view of this theorem, my reformulation of Corollary 2 leads to the corollary as he stated it.

A *separating* family is a collection of real-valued functions G over X, together with a function δ from the positive reals $\mathbb{R}^+$ into $\mathbb{R}^+$, and also together with two selection algorithms σ and τ. Algorithm σ takes elements x, y of X and a positive real number ϵ as input, where $\rho(x, y) \geq \epsilon$, and selects an element g of G such that for all z in X we have

$$\rho(x, z) \leq \delta(\epsilon) \text{ implies } |g(z)| \leq \epsilon,$$
$$\rho(y, z) \leq \delta(\epsilon) \text{ implies } |g(z) - 1| \leq \epsilon.$$

Algorithm τ takes an element y of X and a positive real number ϵ as input, and selects an element g of G such that the second of the above implications holds, for all z in X.

Thus the reformulated Corollary 2 is an algorithm that takes a nontrivial compact metric space X as input and yields a separating family (δ, σ, τ), where σ and τ select functions of the form $\rho(x, x_0)$. Here is the construction:

```
X-separating family procedure Corollary 2(compact metric space X;
                                          X-element y0, y1);
comment y0 and y1 are distinct elements of X;
begin R+ → R+ function δ;
X x X → R+ function d;
X × X × R+ → C(X) function σ;
X x R+ → C(X) function τ;
X x X → R function d;
d(x, y) := X.ρ(x, y);   comment the distance function in X;
δ(ε) := min(ε², ½εd(y0, y1)));
σ(x, y, ε) := (R procedure g(X-element z);
                     return d(x, z)/d(x, y));
τ(y, ε) := (R procedure g(X-element z);
                     return(if d(y, y1) ≤ ½d(y0, y1)
                            then d(y, z)/d(y, y0)
                            else d(y, z)/d(y, y1)))
return (δ, σ, τ);
cud.
```

My notation for the complicated types involved in these algorithms is not the best possible, but I hope it is reasonably comprehensible without further explanation. The selection rule σ determined by this algorithm has the desired property since, for example, $\rho(x, y) \geq \epsilon$ and $\rho(y, z) \leq \delta(\epsilon) \leq \epsilon^2$ implies that $|g(z) - 1| = |\rho(x, z) - \rho(x, y)|/\rho(x, y) \leq \rho(y, z)/\rho(x, y) \leq \epsilon$.

Bishop's proof of Corollary 3 can now be displayed more explicitly as an algorithm in the following way. If X is a compact subset of R, under Bishop's definition, we can compute A4 = bound(X) such that X is contained in the closed interval $[-M, M]$. Let us assume that his Theorem 7 is a procedure whose input parameters consist of a compact metric space X, a separating family (δ, σ, τ) over X that selects functions from some set G $\subseteq$ C(X), and a uniformly continuous function $f : X \rightarrow R$, and a positive real number ϵ. The output of this procedure is an element A of A(G), namely a finite sum of terms of the form $C g_1(x) \ldots g_m(x)$ where $m \geq 1$ and each $g_i \in G$; this output satisfies $|A(x) - f(x)| \leq \epsilon$ for all x in X.

Here is the fleshed-out form of Corollary 3:

```
R polynomial procedure Corollary 3(compact real set X;
                                   X-continuous function f;
                                   positive ε);
begin R polynomial p, q, r;  real M, B;  X-element y0, y1;
A(G)-element A, where G is the set of functions x ↦ c|x − x0|;
M := bound(X);
y0 := element(X);
if trivial(X) then r(t) := f(y0)
else begin y1 := element(X \ {y0});
    A := Theorem 7(X, Corollary 2(X, ye, y1), f, ½ε);
    B := suitable function of A, see below;
    p(t) := Lemma 5(ε/B);
    q(t) := 2Mp(t/2M);
    comment ||x − x0| − q(x − x0)| ≤ ε/B for all x;
    r(x) := substitute cq(x − x0) for each factor gi(x) = c|x − x0|
            of each term of A;
    comment B was chosen so that |q(x − x0) − |x − x0|| ≤ ε/B
                implies that |r(x) − A(x)| ≤ ½ε;
    end;
return r;
end.
```

Clearly it would be an extremely interesting project from the standpoint of high-level programming language design to find an elegant notation in which Bishop's constructions are both readable and explicit.

Tentative Conclusions.

What insights do we get from these nine randomly selected examples of mathematics? In the first place, they point out something that should have been obvious to me from the start, that there is no such thing as "mathematical thinking" as a single concept; mathematicians use a variety of modes of thought, not just one. My question about computer-science thinking as distinct from math thinking therefore needs to be reformulated. Indeed, during my student days, I not only would wear my CS hat when programming computers and my math hat when taking courses, I also had other hats representing the modes of thought I used when I was editing a student magazine or when I was acting as officer of a fraternity, etc. And al-Bîrûnî's biography shows that he had more hats than anybody else.

Thus, it seems better to think of a model in which people have a certain number of different modes of thought, something like genes in DNA. It is probable that computer scientists and mathematicians overlap in the sense that they share several modes of thought, yet there are other modes peculiar to one or the other. Under this model, different areas of science would be characterized by different "personality profiles."

I tried to distill out different kinds of reasoning in the nine examples, and I came up with nine categories that I tentatively would diagram as follows. (Two x's means a strong use of some reasoning mode, while one x indicates a mild connection.)

	Formula manipulation	Representation of reality	Behavior of function values	Reduction to simpler problems	Dealing with infinity	Generalization	Abstract reasoning	Information structures	Algorithms
1 (Thomas)	xx	xx	xx						
2 (Lavrent'ev)	xx		x		xx				
3 (Kelley)	x					xx	xx		
4 (Euler)	xx		xx	x		xx			x
5 (Zariski)	x			x	xx	x	xx	xx	
6 (Kleene)	x					xx	xx		x
7 (Knuth)	xx	x		x					
8 (Pólya)	xx		xx	xx	xx				
9 (Bishop)	xx		xx	xx		x	xx	xx	x
"Algorithmic thinking"	x	xx		xx			xx	xx	xx

These nine categories aren't precisely defined, and they may represent combinations of more fundamental things; for example, both formula manipulation and generalization involve the general idea of pattern recognition, spotting certain kinds of order. Another fundamental distinction might be in the type of "visualization" needed, whether it be geometric or abstract or recursive, etc. Thus, I am not at all certain of the categories, they are simply put forward as a basis for discussion.

I have added a tenth row to the table labeled "algorithmic thinking," trying to make it represent my perception of the most typical thought processes used by a computer scientist. Since computer science is such a young discipline, I don't know what books would be appropriate candidates from which to examine page 100; perhaps some of you can help me round out this study. It seems to me that most of the modes of thought listed in the table are common in computer science as well as in mathematics, with the notable exception of "reasoning about infinity." Infinite-dimensional spaces seem to be of little relevance for computer scientists, although most other branches of mathematics have been extensively applied in many ways.

Computer scientists will notice, I think, that one type of thinking is absent from the examples we have studied, so this may be the thing that separates mathematicians from computer scientists. The missing concept is related to the "assignment operation" $:=$, which changes values of quantities. More precisely, I would say the missing concept is the dynamic notion of the state of a process: "How did I get here? What is true now? What should happen next if I'm going to get to the end?" Changing states of affairs, or snapshots of a computation, seem to be intimately related to algorithms and algorithmic thinking. Many of the concepts of data structures, which are so fundamental in computer science, depend very heavily on an ability to reason about the notion of process states, and so do the studies of the interaction of processes that are acting simultaneously.

Our nine examples don't have anything resembling "$n := n + 1$", except for Euler's discussion where he essentially begins by setting $x := x - \frac{1}{4}A$. The assignment operations in Bishop's constructions aren't really assignments, they are simply definitions of quantities, and those definitions won't be changed. This discrepancy between classical mathematics and computer science is well illustrated by the fact that Burks, Goldstine, and von Neumann did not actually have the notion of assignment in their early notes on computer programming; they used a curious in-between concept instead (see [15]).

The closest thing to "$:=$" in classical mathematics is the reduction of a relatively hard problem to a simpler one, since the simpler problem replaces the former one. Al-Khwarizmi did this when he divided both sides of a quadratic equation by the coefficient of x^2; so I shall conclude this lecture by once again paying tribute to al-Khwarizmi, a remarkable pioneer in our discipline.

References.

[1] A. D. Aleksandrov, A. N. Kolmogorov, and M. A. Lavrent'ev, eds., Mathematics: Its Content, Methods *and* Meaning 1 (Cambridge, Mass.: MIT Press, 1963). Translated by S. H. Gould and T. Bartha from *Matematika: Eë Soderzhanie, Metody i Znachenie* (Akad. Nauk SSSR, 1956).

[2] Errett Bishop, *Foundations of Constructive Analysis* (N.Y.: McGraw-Kill, 1967).

[3] Baldassarre Boncompagni, ed., Algoritmi de Numero Indorum. *Trattati D* 'Aritmetica 1 (Rome, 1857).

[4] Solomon Gandz, "The Mishnat Ha Middot," *Proc. Amer. Acad. for Jewish* Res. 4 (1933), l-104. Reprinted in S. Gandz, Studies in Hebrew *Astronomy and Mathematics* (New York: Ktav, (1970), 295–400.

[5] Solomon Gandz, "Sources of al-Khowârizmî 's Algebra," Osiris 1 (1936), 263-277.

[6] Solomon Gandz, "The origin and development of the quadratic equations in Babylonian, Greek, and early Arabic algebra," Osiris 3 (1938), 405-557.

[7] Solomon Gandz, "The algebra of inheritance," Osiris 5 (1938) [sic], 319-391.

[8] Fred Gruenberger, "The role of education in preparing effective computing personnel," in F. Gruenberger, ed., Effective vs. *Efficient Computing* (Englewood Cliffs, N.J.: Prentice-Hall, 1973), 112–120.

[9] A. P. Îushkevich, "Arifmeticheskiĭ traktat Mykhammeda Ben Musa Al-Khorezmi," *Trudy Inst. Istorii Estestvoznaniîa i tekhniki* 1 (1954), 85-127.

[10] Louis Charles Karpinski, ed., Robert of Chester's Latin Translation of the *Algebra* of Al-Khowârizmî. *Univ. Michigan Humanistic* Series 11, part 1 (Ann Arbor, 1915), 164 pp. Reprinted in 1930.

[11] John L. Kelley, General Topology (Princeton: D. Van Nostrand, 1955).

[12] E. S. Kennedy, "al-Bīrūnī," Dictionary of *Scientific* Biography 2 (N.Y.: Charles Scribner's Sons, 1970), 147–158.

[13] Stephen Cole Kleene, *Introduction* to Metamathematics (Princeton: D. Van Nostrand, 1952).

[14] Donald E. Knuth, *Fundamental Algorithms* (Reading, Mass.: Addison-Wesley, 1968).

[15] Donald E. Knuth and Luis Trabb Pardo, "The early development of programming languages," *Encyclopedia of Computer Science* and Technology 7 (N.Y.: Marcel Dekker).

[16] Iu. Kh. Kopelevich and B. A. Rosenfel'd, tr., Mukhammad al'-Kh'orezmi's *Matematicheskie Traktaty* (Tashkent: Akad. Nauk Uzbekskoĭ SSR, 1964). [Includes al-Khwârizmî's arithmetic and algebra, with commentaries by B. A. Rosenfel'd.]

[17] Seyyed Hossein Nasr et al., Historical Atlas of Iran (Tehran, 1971).

[18] D. Pingree, review of [27], Math. Reviews 30 (July, 1965), No. 5.

[19] G. Pólya and G. Szegö, Problems and Theorems *in* Analysis 1 (Berlin: Springer, 1972).

[20] Ed. Sachau, "Algebraisches über das Schach bei Bîrûnî," *Zeitschrift* d. *Deutsche Morgenländische Gesellschaft* 29 (1876), 148-156.

[21] C. Edward Sachau, transl. and ed., al-Bîrûnî's *Chronology of Ancient* Nations (London: William H. Allen and Co., 1879).

[22] A. S. Saidan, *The* Arithmetic *of al-Uqlīdisī* (Dordrecht: D. Reidel, 1975).

[23] S. Kh. Sirazhdinov and G. P. Matvievskaia, *Abu Raĭkhan Beruni i Ego* Matematicheskie *Trudy* (Moscow: Prosveshchenie, 1978).

[24] D. J. Struik, ed., A Source Book *in* Mathematics, *1200-1800* (Cambridge, Mass.: Harvard University Press, 1969).

[25] George B. Thomas, Calculus and Analytic Geometry, 2nd ed. (Cambridge, Mass.: Addison-Wesley, 1956).

[26] G. J. Toomer, "al-Khwārizmī," Dictionary *of Scientific Biography* 7 *(N.Y.:* Charles Scribner's Sons, 1973), 358-365.

[27] Kurt Vogel, ed., Mohammed Ibn Musa Alchwarizmi's *Algorismus,* Das früheste Lehrbuch zum Rechnen mit indischen Ziffern (Aalen/Osnabruck: Otto Zeller Verlagsbuchhandlung, 1963). [This edition contains a facsimile of the manuscript, from which a correct transcription can be deduced.]

[28] Oscar Zariski and Pierre Samuel, Commutative Algebra 1 (Princeton: D. Van Nostrand, 1958).

Note *on the spelling of Khwârizm:* In the first and second editions of my book [14] I spelled Muḥammad ben Mûsâ's name "al-Khowârizmî," following the convention used in most American books up to about 1930 and perpetrated in many other modern texts. Recently I learned that "al-Khuwârizmî" would be a more proper transliteration of the Arabic letters, since the character in question currently has an 'oo' sound; the U.S. Library of Congress uses this convention. The Moorish scholars who brought Arabic works to Spain in medieval times evidently pronounced the letter as they would say a Latin 'o'; and it is not clear to what extent this particular vowel has changed its pronunciation in the East or the West, or both, since those days. At any rate, from about 1935 until the present time, the leading American scholars of oriental mathematics history have almost unanimously agreed on the form "al-Khwârizmî" (or its equivalent, "al-Khwarizmi", which is easier to type on conventional typewriters). They obviously know the subject much better than I do, so I am happy to conform to their practice.