
A Conversation with Don Knuth: Part I

Donald J. Albers and Lynn Arthur Steen

As a high school senior in Milwaukee, Donald E. Knuth had doubts about his ability to graduate from college. Four years later he received his B.S. in mathematics, *summa cum laude*, from the Case Institute of Technology in 1960. His work had been so distinguished that by a special (unprecedented) vote of the faculty he was simultaneously awarded an M.S. degree. In 1963 he received his Ph.D. in mathematics from the California Institute of Technology. Over the years he has received many prestigious awards. In 1979, at age 41, he was awarded the National Medal of Science by President Carter.

By any measure, Don Knuth is a remarkable man. He is generally regarded as the preeminent scholar of computer science in the world. He also is an accomplished organist, composer, and novelist.

He is a prolific writer on a host of topics, and he has contributed to an unusually large number of publications. His first publication was for *MAD Magazine*. Since then he has written for *Datamation*, the *Journal of Recreational Mathematics*, the *American Mathematical Monthly*, and dozens of other mathematics and computer science journals such as *Acta Arithmetica* and *Acta Informatica*. He is best known for his monumental series of books, *The Art of Computer Programming*, which has been translated into several languages, ranging from Chinese to Russian. Three of a projected seven volumes in the series are now completed, with part of the fourth in preprint form. His progress on the series has been slowed by a four-year-long diversion on computer-assisted typesetting.

On January 12, 1981, Donald J. Albers, Editor of the TYCMJ, and Lynn Arthur Steen, Past Editor of *Mathematics Magazine*, interviewed him in his Stanford office. What follows are excerpts from that interview.

TYCMJ: *You are a computer scientist, and yet you started out in mathematics. When did your mathematical interests first emerge?*

Knuth: In my freshman year of high school I got very interested in mathematics. In fact, I think I ruined my eyes drawing hundreds of graphs on orange graph paper with dim lighting. I started to get headaches from drawing those graphs, but I was fascinated with them. The typical graph would be some function like $y = \sqrt{ax + b} - \sqrt{cx + d}$, where I would fix b , c , and d and vary a , in order to see what would happen to the shape of the graph. I had hundreds and hundreds of such graphs where I wanted to see the behavior of functions.

TYCMJ: *Did you have an outstanding teacher along the way?*

Knuth: The mathematics teacher that I remember most and who inspired me the most was in college. My high school senior teacher also introduced me to things like binary numbers and encouraged me to do recreational things. During my senior year in high school, I entered the Westinghouse Science Talent Search. I made two entries. One was sort of physics oriented, and the other was a number system based



President Carter presenting the National Medal of Science to Knuth.

on π . I had thought about imaginary number bases and irrational number bases, and I played around with the kind of logarithm tables that would result from such bases. I didn't win the prize, but I do remember having a lot of fun thinking about number systems. I also played around a lot with absolute value functions. When I learned about the absolute value function, I started making another set of graphs. I worked out a system so that if somebody gave me a pattern of connected straight lines, I would be able to write down a function whose graph gave that pattern. I was absolutely fascinated with graphs in mathematics. My physics teacher was my favorite teacher in high school. I was torn between physics and music, but I enrolled in college as a physics major. I had done a lot of piano playing and some orchestrations, so I didn't know whether I should major in music or physics. My choice of majors was due essentially to the different scholarships I got. The college I chose was better in physics than in music. If I had gone to Valparaiso instead of Case, I would have majored in music.

TYCMJ: *Is there any other music or mathematics talent in your family?*

Knuth: My dad was a church organist and now I am.

TYCMJ: *Do you think there is much to the suggested connection between musical ability and mathematical ability?*

Knuth: There is definitely something to it. Go, for example, to the Mathematics Institute at Oberwolfach in Germany. Every week mathematicians come there for conferences, and music is the main recreation. They have a tremendous music library and many people will come with their instruments. Chamber music fills the halls almost every night as the mathematicians get together. In our department now we also are surrounded by chamber music. I just was talking to the administrative assistant of our department about this. She had previously worked in the law school, and she said in the law school one professor out of 20 might go to a concert once in awhile. They weren't that much interested in music. Here in the Computer Science Department she felt that more than half the people were musicians themselves.

TYCMJ: *Do you have a theory as to what you perceive as a real connection between music and mathematics?*

Knuth: No, I really don't understand why. I guess Euler liked music, but I can't say how many great mathematicians of bygone days were really good musicians. I haven't studied that. There is definitely a correlation, certainly at Stanford.

I also found this in my friend, Professor Dahl from Norway, who carries piano duet music all of the time with him in his briefcase. No matter where he goes, he finds someone to play with him. He was the one who introduced me to the beauty of four-hands piano music, and now I have quite a collection of it built up over the last ten years.

TYCMJ: *I wonder if we could get back to your early schooling experiences from a different angle, namely, writing. When did your writing interests first emerge?*

Knuth: Our grade school was very good in English grammar. I remember one of the most interesting things for me in the 7th and 8th grades was to diagram sentences. A bunch of us would get together after class to try diagramming. We could diagram all of the sentences in the English book, but we couldn't diagram many of the other sentences we saw around us, especially the ones we saw in the hymnal. We couldn't figure out what was going on. They just didn't fit any of the rules we learned. We worked hard on this, and it was a big thing for us at the time. In high school, I found out that everyone from our grade school was whizzing through the English classes because of what we had learned; so it wasn't just me.

TYCMJ: *I was going to ask you if this was a public school.*

Knuth: It was a Lutheran school. My dad was a Lutheran school teacher, and I attended a Lutheran high school in Milwaukee. My grade school education in writing was really good. I'm trying to do that for my kids now. I have them write an essay every week. If they want permission to watch television the next week, they have to turn in their essay the previous week. When I got to college, I found out that writing was almost 50% of what I had to do well, and the other half was mathematics.

The other strange thing I remember about grade school occurred when I was in the 8th grade. There was a contest run by the manufacturers of Ziegler's Giant Bar in Milwaukee. The contest consisted of trying to find how many words could be made out of the letters in "Ziegler's Giant Bar." This contest appealed to me very much, and I told my parents I had a stomach ache so that I didn't have to go to school for two weeks. I spent all those two weeks with an unabridged dictionary

finding all the words I could get from the specified letters. I wound up with about 4,500 words, and the judges had only 2,500 on their master list. Afterwards I realized that I could have made even more words if I had used the apostrophe! My dad and mom helped type up the answers I had written out when they saw how interested I was in this project. The prize was a television set for the school. So our school got a TV set in the classroom, and we got to watch during class that year (1952) one of the first live transmissions from San Francisco across the country. We also got a Ziegler's Giant Bar for everyone in the class.

Professor Guenther was my freshman calculus teacher at Case, and he first exposed me to higher mathematics. Paul Guenther died about five years ago, but he was a great teacher for me mostly because he was so hard to impress. Every time I made a suggestion, I was put down, but he would grudgingly appreciate it when I finally came up with a good one. I don't know why I got so excited about him. In addition to his unimpressibility, he had a good sense of humor, and he seemed to know as much physics as my physics teachers and as much chemistry as my chemistry teacher. That impressed me: it seemed that mathematics was a little better somehow.

Also, I was scared stiff that I wasn't going to make it in mathematics. My advisors in high school told me that I had done well so far, but they didn't think I could carry it on in college. They said college was really tough, and the Dean had told us that one out of three would fail in the first year. In high school, I did have the all-time record for grades. We graded not on A, B, C, D, but on percentages in every course. And my overall percentage in classes was better than 97.5%.

TYCMJ: *Didn't that instill in you a great deal of confidence?*

Knuth: I always had an inferiority complex—that's why I worked so hard. I was an over-achiever probably.

At Case, I spent hours and hours studying the mathematics book we used—*Calculus and Analytic Geometry* by Thomas—and I worked every supplementary problem in the book. We were assigned only the even-numbered problems, but I did every single one together with the extras in the back of the book because I felt so scared. I thought I should do all of them. I found at first that it was very slow going, and I worked late at night to do it. I think the only reason I did this was because I was worried about passing. But then I found out that after a few months I could do all of the problems in the same amount of time that it took the other kids to do just the odd-numbered ones. I had learned enough about problem solving by that time that I could gain speed, so it turned out to be very lucky that I crashed into it real hard at the beginning.

I started as a physics major, but my turn towards mathematics came in my sophomore year. I took a course in abstract mathematics from Professor Green, who is still teaching at Case. He had written his own textbook for the class, where he would give axioms for Boolean algebra, logic, etc. All of a sudden I realized that it was something I liked very much. And he gave a special problem without telling us whether it was possible or not. He said that if anyone could solve the problem, they would get an automatic "A" in the course. So, of course, none of us tried it. It was obviously hopeless, for he had quite a reputation. As far as we were concerned, there was no way to do it.

I was in the marching band that fall. But I missed the bus and had nothing to do, so I decided to kill time by working on that impossible problem. By a stroke of luck I was able to solve it, so I handed a solution in on Monday. He said, "Okay, Knuth, you get an 'A' in the course." I cut class the rest of the quarter, and he lived up to his bargain. I felt guilty about cutting classes afterwards, so I became the grader for the course the next year. Instead of doing the homework, I was grading it.

In physics I was having a terrible time in the welding lab. I never was very good in laboratories, either in chemistry or in physics. My experiments just wouldn't work. I would drop things on the floor, and would always be the last one to finish. Once I had to report an experimental error of 140% in chemistry lab. I objected to their formula for experimental error, since I thought nobody could be more than 100% wrong, but they wouldn't listen to me.

In welding it was even worse. I was too tall for the welding tables, and my eyes weren't good enough; I couldn't wear my glasses underneath the goggles. Everything would go wrong, and I was terrified by what seemed like hundreds of thousands of volts of electricity! I just wanted nothing to do with it, yet physics majors were required to do this lab work.

On the other hand, I had Professor Green's course in abstract mathematics, which seemed very appealing to me. Just for fun I had made up sort of random axioms for what turned out to be a ternary logic, something that looked a little like Boolean algebra. The idea was to see if those axioms would lead to any theorems. I was working hard, trying to get something to follow from those axioms, so hard that I found my other grades were going down, so I had to stop working on it. I had set up—it's probably pretty trivial now, I suppose—some operation that would be analogous to a truth table, and I finally proved a theorem that went something like this: "*The absitive of the posilute of two cosmoframemics is equal to the posilute of their absitives.*" I just made up words for certain abstract concepts, and it appealed to me that I could prove a theorem that was analogous to de Morgan's Law.

All the way through my student work I had been joyfully stuck in Chapter One of my math books, thinking about the definitions of things and trying to make little modifications, seeing what could be discovered and working from there. I really enjoyed that, but the physics labs were killing me: The combination resulted in my switching to a math major at the end of my sophomore year.

TYCMJ: *Somewhere along the line you began to work with computers. Was there some point where it became clear to you that you would work with them for a long time?*

Knuth: Between my freshman and sophomore year, I had a summer job drawing graphs for statisticians at Case. In the room next to where I worked was a computer.

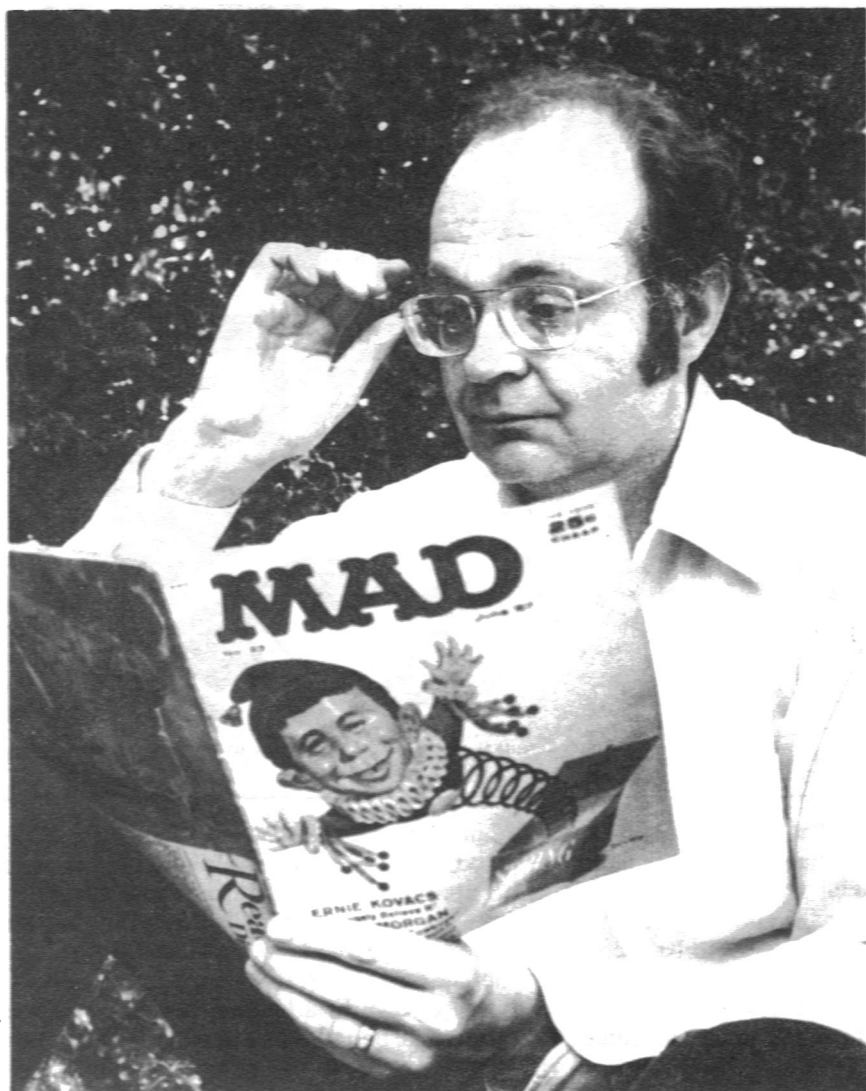
TYCMJ: *So the graphing continued?*

Knuth: Yes, I could draw graphs.

TYCMJ: *Was this compulsive?*

Knuth: Well, yes; to some extent, METAFONT (the system for computer-assisted typography that I recently developed) probably reflects my love of graphing.

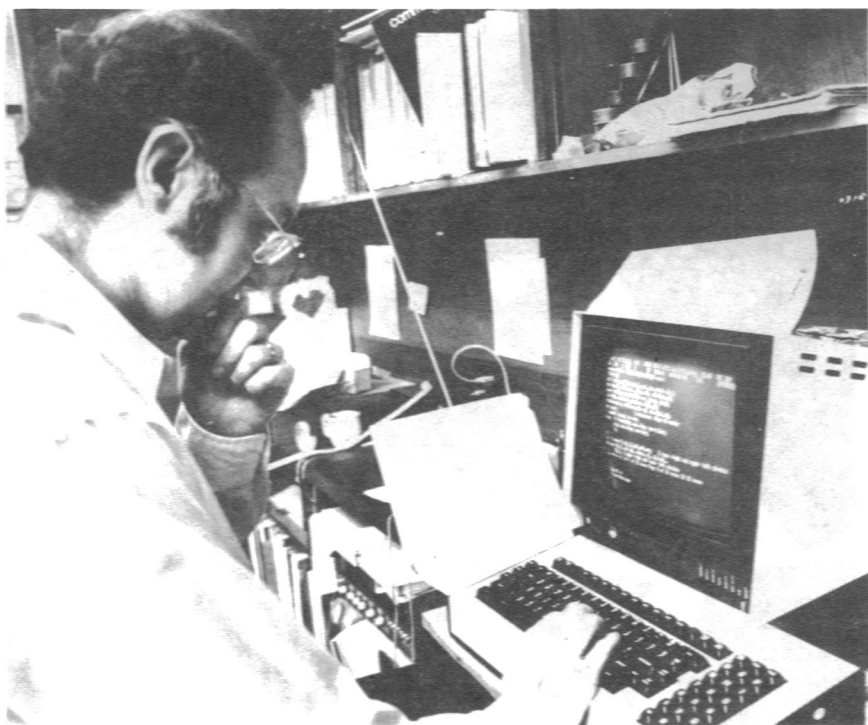
The Statistics Department at Case was located right next to a new computer, a



What's this?—The world's foremost computer scientist reading MAD Magazine! Don Knuth's first publication was in MAD.

wonderful machine with flashing lights. Early that summer, someone explained to me how it worked. Pretty soon I was hooked. I spent a lot of nights, all night long, at the console of the computer. Nobody else was there. I discovered girls in my sophomore year. This was before that; I had computers first.

I still have my first computer program. It factored numbers into primes. You would dial a ten-digit number into the console, and it would punch the factors on



Knuth working at the computer terminal in his study. He refers to the study as his "book factory."

cards. The program initially was about 70 instructions long, and as I recall, by the time I finished it, I had removed more than 100 errors out of 70 lines. In other words, I made a lot of mistakes, but I always felt I learned from those mistakes. The program wouldn't work, and I kept on fixing it, and finally it worked. My second program was to do base conversions.

My third program was to play tic-tac-toe, and it also would learn *how* to play tic-tac-toe. I worked hard on this one. I developed a learning strategy where every position in a game the computer won would be rated as a little bit better; but if it lost the game, every position would be marked as bad. This was a memory that would adapt itself. Each position in the game had a number from zero to nine representing how good it was thought to be; the neutral value was four, so if it was a drawn game, the position ratings would tend to go towards four. I wrote another program that would play tic-tac-toe perfectly, and then I had these two programs playing each other. After 90 games, the learning program learned how to draw against the good one. In another experiment, after 300 games, two learning programs starting with blank memories learned to draw against each other. They played a very conservative game, not very exciting, but it was interesting to see "the blind leading the blind."

That was my first month of learning to program. Those things were really fun. The next thing I did was somewhat different: I wrote an assembly program for the

machine. I started to read the code of other people, and I got especially interested in programming because most of what I read wasn't very well written. I could look at programs and say here I am, only a college freshman, and I can do better than these professionals. I didn't know that lots of people could do better than those professionals. The standards of software at that time were pretty bad. I began to think that I had a special talent for it. Maybe I did, but I don't think now that it was as special as I had once thought. Then I read the code for Stan Poley's assembly program, which I thought was truly beautiful, a masterpiece of elegant programming, so I was inspired to carry his ideas one step further.

TYCMJ: *Feelings of inferiority were certainly not present in that work.*

Knuth: Well, part of me was anxious to prove to the other part that I was OK! So, I was always motivated by seeing publications by someone who was apparently an expert, where I thought I could do a little better. Now I always tell my students when I make mistakes in class that I'm just trying to motivate them.

TYCMJ: *The kinds of experiences with computers you had as a college sophomore, with minor modifications, are going on in junior high school now. What effect do you see that having on the future?*

Knuth: The students aren't learning how to write. That's serious. They don't know how to spell "mnemonic." They're specializing too early. There's a danger that people aren't seeing the other side of the coin, which is writing. That's what I worry about. As long as people keep open to a lot of aspects of life, then it's good, but if they become involved too much with one subculture, then it's going to limit them later.

TYCMJ: *What is it about computers that makes people so compulsive, either pro or con? A lot of people just can't stand programming. There are others who just get consumed by it.*

Knuth: It's partly a strange way of thinking. There are so many different modes of thinking, not really understood yet by psychologists. Teachers of computer science regularly find that 2% of the people who enroll in their courses are natural-born computer scientists who really resonate with computer programming. There seems to be a correlation between that and mathematical logic. I said I enjoyed the abstract algebra course where I first really studied axioms and Boolean algebra. If you look in math departments, the faculty who have traditionally been the closest to computer science have been the people in logic and combinatorics. Conversely, the mathematicians who are best at geometrical visualization tend not to enjoy a discrete universe like the computer world.

Another difference is between finite and infinite mathematics. I used to say to Peter Crawley at Cal Tech that he and I intersect at countable infinity, because I never think of anything more than a countable infinity, and he never thinks of anything less than a countable infinity. Higher infinities involve a kind of reasoning and intuition that doesn't apply very much to computers at all.

There are different flavors of mathematics, based on what kinds of peculiar minds we have; we aren't going to change that. People find out what things are best for them. And as for their mentalities, physicists are different from mathematicians, as are lawyers from doctors. Each of these fields seems to have predominant modes of thinking, which people somehow recognize as best for them. Computer science, I

am convinced, exists today in universities because it corresponds to a mode of thinking, a peculiar mind-set that is the computer scientist's way of looking at knowledge. One out of 50 people, say, has this peculiarity.

Historically, such people were scattered in many walks of life; we had no home to call our own. When computer science started out it was mostly treated as a tool for the existing disciplines and not of interest in its own right. But being a useful tool is not enough in itself to account for the fact that computer science is now thriving in thousands of places. For example, an electron microscope is a marvelous tool, but "electron microscope science" has not taken the world by storm; something other than the usefulness of computers must account for the rapid spread of computer science. What actually happened was that the people who got interested in computers started to realize that their peculiar way of thinking was shared by others, so they began to congregate in places where they could have people like themselves to work with. This is how computer science came to exist. Now we can look back in old writings and see that certain people were really computer scientists at heart. It was latent back in Babylonian times, and throughout history.

TYCMJ: *Are computer scientists really different from mathematicians, then?*

Knuth: I think you can recognize a difference.

TYCMJ: *But how would I recognize a computer scientist if he walks in the door?*

Knuth: By the thought process. I've been trying to answer exactly that question. In order to get a handle on it, I tried to study several works on mathematics to discover the typical paradigms of good mathematicians, using a random sampling technique. What I did was to take nine books that would represent mathematics, and I looked at page 100 of each book. I analyzed that page very carefully, until I understood what kinds of things were there on that page. It was interesting to see what aspects of mathematical thought processes were involved. I asked myself: "If I had to write a computer program to discover the mathematics on page 100, what capabilities would I have to put in that program?"

I found that one of the most striking things distinguishing mathematicians from computer scientists was their strong geometric reasoning and reasoning about infinity. The things that were common to both computer science and mathematics were primarily things like the use of abstractions and the manipulation of formulas. The main thing that was prominent in computer science that wasn't in mathematics was an emphasis on the state of a process as it changes, where it changes in time in a discrete way. In computer science when you say n is replaced by $n + 1$, the old value disappears and the new value takes over. We know how to think about an algorithm that is half-way executed; it has a state consisting of the current values of all the variables, and the state also specifies what rule to apply next. In order to formulate this for most mathematicians, it requires putting subscripts on everything. Traditional mathematics doesn't have this notion of a process in highly developed form, but it is vital in computer science.

The other striking difference was that computer scientists are willing to deal with diverse case analyses. The more pure the mathematician, the more he or she instinctively likes to have a clean formula that covers everything in all cases. But computer scientists are able to reason comfortably about things that have different cases, where we do step one, step two, then step three. A mathematician likes to have one step that you can apply over and over again.

TYCMJ: *What you are saying reminds me of an argument by Edsger Dijkstra, that computer scientists have now learned enough about the process of mathematics—in terms of how formulas are manipulated and how things change—that it should feed back into the process of teaching mathematics. Does the computer scientist actually now know enough to develop a science of mathematics that would be useful in teaching?*

Knuth: Since people have different modes of thinking, I doubt if any one way of teaching will be simultaneously the best way to reach different types of students; and I also doubt if many people can design educational plans that work for students having a different mind-set from the educational planners. So I can't be confident that a method best for me would be best for the world. But certainly Dijkstra's proposal would be the best way to teach mathematics to a natural-born computer scientist. From my own perspective, I feel that I have really learned some subject of mathematics at the point when I understand how it works in an algorithmic formulation.

For example, consider Volume 2 of my books. I think every theorem of elementary number theory is in there somewhere, but it's in the context of an algorithm that somebody needed because of a computational problem that had to be solved. I believe that the original discovery of these ideas was because of the need for such algorithms, so I presented it that way. It's a different aesthetic from mathematics, you see, from what is mathematically "clean" to what is not elegant in the same way. It's a different way of thinking, and I can't argue that one is better than the other.

Dijkstra and I are natural-born computer scientists. We found that out after we got older. Such ways of organizing knowledge are not going to be for everybody, but for our subset of the population, an algorithmic approach works best.

The knowledge that we have a computer scientist mentality is also a challenge because we have to do our best for the other 49 out of 50 people who don't think as computer scientists; computers are affecting everybody's lives. We have to find a way to make it comfortable for other people to use computers, even though we don't really understand the way they think any more than they understand the way we think. You need people who are halfway between the different modes of thinking to bridge these gaps.

Of course, there is no definite boundary that you cross in going from computer scientist to mathematician to physicist, etc. There tends to be, in a multidimensional space of different kinds of abilities, a focus around the place that is most representative of computer scientists, and another one that is more typical of mathematicians. Maybe musicians are also close. Who knows? But it is a continuous thing.

Computer science departments thrive because there are a lot of people near our focal point in "thought space." In past ages, some of the people we now would call computer scientists were called mathematicians, physicists, chemists, doctors, businessmen. Now they have a home. That's what is holding the field together. But to develop our field well, if we want other people to make use of computer science, we have to realize that we aren't able to do it all ourselves. We need others who can understand the other modes of thinking.

Maybe Dijkstra would argue that our mode of thinking is more powerful somehow, that it includes the other ones. I'm not quite so bold yet to do that, but

the reason I raise this possibility is because I once met someone who had been a computer science major in graduate school and then went on to become one of the main advisors to President Lopez in Mexico. He told me that his training in computer science was of great value to him in working with all the people he had to deal with. Even though he wasn't programming computers any more, he felt that his approach to knowledge enabled him to understand lots of different people who were talking to him but who couldn't understand each other very well. The computer science view seemed to be more powerful in its models of reality. This might be true because computer scientists are accustomed to models that can handle a variety of cases. The real world breaks down into cases, and mathematical models are better or worse depending on the uniformity of what they're modeling. The computer scientist dealing with less uniform models is perhaps able to cope with more general things. On the other hand, when it gets to something that is truly uniform, a computer scientist will not be able to go as deeply as a mathematician. Uniformity might really be the most striking difference between the fields.

TYCMJ: *A few have suggested that mathematics may be a part of computer science. Others say that computer science is a part of mathematics. What do you think?*

Knuth: I really think that they are two things, although they are related. There is a lot of overlap, but also I think I can tell when I'm in my mathematics mode, wearing my mathematician's cap, and that I can almost feel the changes when I go into a computer-science mode. I can't exactly say why; maybe I'll never know. But I can definitely feel when I'm behaving as a computer scientist.

I remember once giving a lecture about a number-theoretic problem that seemed to straddle the two fields. I started with the mathematical definitions and took things as far as I could using traditional mathematical tools. Then I said, "Now let's tackle this problem as a computer scientist would." And I carried on for 15 minutes with an algorithmic viewpoint, after which I said, "This is as far as the computer scientist is going to get. Now let's be mathematicians again for a while." And I could feel strongly that this was true, that I was sometimes doing what a mathematician would definitely do, while at other times I knew I was doing something that a mathematician wouldn't do.

Neither the mathematician nor the computer scientist is bound by a study of nature. With a pencil and paper we can control exactly what we are working on. A theorem is true or it's false. The fact that we deal with man-made things is common both to mathematics and computer science, but the nature of the thought process is sufficiently different that there probably is a justification for considering them to be two different views of the world—two different ways of organizing knowledge abstractly that have some points in common, but in a way they have their own domains.

TYCMJ: *In addition to these intellectual contrasts between mathematics and computer science, there is getting to be a lot of social and educational concern about their interaction. The growth of the computer field, for example, is drawing so many people into bachelor's level computing that there aren't very many people going on into advanced work in mathematics or computer science or into high school mathematics teaching. Computer science and mathematics appeal to the same limited group of people, and at the moment, the momentum seems to be running pretty well in favor of computer science.*

Knuth: But that's an economic consideration. Right now bachelors in computer science are getting better salaries than anybody else. This is our problem because people are going into it who aren't natural-born computer scientists; they're just going into it for the money, and not for the love of it. This trend can actually be to some advantage to mathematicians because they've now got motivated students in their classes instead of people who are just there for some external reason.

I don't agree with your statement that "computer science and mathematics appeal to the same limited group of people," but our disagreement is probably due to the fact that I have been thinking mostly of extreme cases, the differences between the best computer scientists and the best mathematicians. According to the law of large numbers, there will of course be very few people who rate a 10 on a scale of 1 to 10, under almost any ranking criteria, and there aren't many 9's either. So let's consider somebody who is an 8 at mathematics and a 6, say, at computer science. That person is somewhat likely to go into computer science instead of mathematics, nowadays, because the salary is better; and perhaps society will be better off since there is a pressing need for computer scientists. I agree that economic pressure is causing mathematics to lose a lot of its 7's and 8's (if you'll excuse my callous use of numbers in place of human beings); but I hope that the differences between computer science and mathematics will be well enough understood that you don't lose the 9's and 10's. Their mathematical abilities are vital to people in all other fields, including computer science.

TYCMJ: *One of the things that we keep hearing today is that good instruction in mathematics is reversing dramatically and perhaps in other subjects as well. Many say the reversal is due to the influence of computers and calculators.*



At the age of three, Don Knuth was already attracted to keyboards.



Knuth is an accomplished organist and composer. "I want to write some music for organ with computer help.—If I live long enough, I would like to write a rather long work that would be based on the book of Revelation. The musical themes would correspond to the symbolism in the book of Revelation."

Knuth: I have been quite disappointed in the mathematics textbook my son has as a sophomore in high school. But his teacher is very good and compensates for it. The worst excess of this book is the pedantry. The second-worst is that its three authors seem to have written three different kinds of chapters without much awareness of what the other authors had done. But the book has some good points, too, like its emphasis on graphs. My suggestion for teaching mathematics at the young levels is to draw graphs! I was glad to see there was much more use of graphs here than in other books I had seen.

I read an article a week ago where someone said he had started to employ older textbooks, and the older the textbook the better the students would do on exams. He went back to about 1900, and he said that was the Golden Era.

Clearly the Hungarian educational system has been the most successful for pure mathematics; it's a model that ought to be studied very carefully because it works. It produces so many good mathematicians *per capita*.

As I see my children learning mathematics now, I find that the books have too much emphasis on flashy things and on memorizing formulas, rather than on what

an idea is good for as a general tool, or on how to reconstruct a formula from a few basic principles instead of from memory.

After reading these books, you don't remember what the degree of a polynomial is, or how to answer various precise questions. My son had learned algebra, but it was not clear to him how to add fractions; that had been glossed over and needed in only one homework assignment. He would remember for a few weeks what the distributive law was, but that would come and go. The high level wording for things wasn't being put to any use. So it disappeared from his mind. In my own case, I got along fine without knowing the name of the distributive law until my sophomore year in college; meanwhile, I had drawn lots of graphs.

TYCMJ: *Do you think students should be introduced to computing earlier in the formal curriculum?*

Knuth: I would like to see an approach by means of algorithms, but I really hesitate to say much about it for fear somebody will believe me when I really haven't thought it through. The only experience I had personally with doing something on an algorithmic basis was at a higher level. At Cal Tech I taught an introductory abstract algebra course. We were studying matrices and there was a question on canonical forms of matrices. The book was very obscure on that point. We wanted to know when one matrix was similar to another. How can you tell? Well, you have a canonical form that says that if two matrices are similar, they have the same canonical form. Books rarely even say that. They just say here is the name of the form, and here is the definition of it.

So we said then, how can we determine if two matrices are similar? There is a little operation you can do on a matrix, something like this: "Subtract a multiple of the third row from the second row, and then add the same multiple of the second column to the third column." That operation preserves similarity: although it is very simple, it can be used to get another matrix similar to it. So let's take a look at what it does. We start with a matrix and try to zero out most of the first row and the first column, and we see that we can almost always do this by simple transformations that preserve similarity. But every once in awhile we get stuck.

The point is that we are solving a problem. We are trying to do it step by step with an algorithm; pretty soon we find this canonical form. Of course, we might fail because some special case might happen. Well, in that case, we won't be able to make so many elements zero. This, I believe, is the way these canonical forms were first found. But later on there was a tendency for people who had learned about them in a concrete way to present them in a high-level, abstract fashion. For example, they would look at the way linear operators behave on subspaces. This is elegant, but pedagogically unsatisfactory because it conceals the method of discovery.

I think you need both views, especially when you are learning a subject at first. The algorithmic point of view tends to be at a more intuitive level. It helps very much in the early stages of teaching. But as I say I haven't been teaching these courses myself for a long time, and my own intuition might not correlate well with that of the majority of students. Careful experiments should be tried, since an algorithmic approach might well be very successful.

TYCMJ: *It was my impression that you were perhaps very excited by your tic-tac-toe learning program. Now that's using the computer for playing games. What do you think of games as an introduction to computing?*

Knuth: When children are young, they get to a stage where they like to make up rules of games. They enjoy arguing with each other. They argue incessantly about the rules because they like to make up rules to games. I think this might be related to making up computer programs. There is probably in children a tendency that can be well exploited to encapsulate these kinds of rules.

I'm way out of my expertise, of course, when thinking about elementary education. But I would like to see an approach where the students learning some concept not only learn how to follow the rules but also how to explain the rules. Suppose you are teaching somebody to add. We carry this out by giving a bunch of examples and tell children that you go from right to left and you carry, and so on. If there were only a simple enough computer language for these second and third graders, it might be good to teach addition by having them write little programs in this language. The teacher would say: "Here's the way to add numbers and we're going to teach it to this goofy machine." I think it's definitely worth a try. Students learn quickly to make up computer programs that will draw pictures. It might be that similar skills would work on things like arithmetic.

I like to see the rules and their exceptions. Other people are very comfortable without the rules; maybe people who are good with language just absorb dictionaries very quickly and don't even make rules out of things. To them, everything is an exception. So I suppose an algorithmic approach will be of no help to them; we need a variety of teaching methods.

TYCMJ: *I have been impressed with a toy called Big Track that uses a language like TURTLE, which comes out of Papert's laboratory. The rules are very explicit and very tight.*

Knuth: That's very good for teaching algorithms. I think arithmetic is another thing that people should still keep learning even though we have calculators. It's nice to be able to add and to multiply numbers by hand, in a pinch. Not only nice: it introduces important patterns of thought.

TYCMJ: *Are you very enthusiastic about calculators?*

Knuth: Well, no. Computer scientists were the last to get enthusiastic about calculators. One of my colleagues remarked the other day that this is probably the only department at the university where nobody has a pocket calculator at the faculty meeting. This may be because we have all these super-computers here in our offices. I can do my symbolic calculations at M.I.T., too, three thousand miles away, using a computer network.

TYCMJ: *Can we turn to some other games for a minute? One of your former teachers at Case tells a good story about you. I don't know whether it is true or not, but it is a good story. He says you were absolutely instrumental in the success of the Case basketball team in 1960. Is that true?*

Knuth: Well, it would be nice to say that. We did win the league title that year, and I'll be glad to take all of the credit for it. But here's what really happened: As the manager of the team, I worked out a formula, a rather complicated jumble of

symbols that I really don't believe in any more. It was a system that would rate each player with a magic number. The magic number would tell how much each player really contributed to the game, not just the points he scored. If he missed a shot, there was a certain chance that our team would not get the rebound, and the other team would get the ball. Thus, he would lose something for each shot missed. Conversely, when he stole the ball, that was very good because our team got possession.

In fact, it is interesting to watch a basketball game and imagine that possession of the ball counts one point. Such an assumption isn't true in the last seconds, but during a good part of the game it isn't out of the question to say that if your team has possession of the ball, it's worth a point. In other words, you look at the scoreboard, and if it's 90 to 85, you add one to whichever team has the ball then. It tends to give you a better feeling of the current score. If you look at basketball this way, then when someone makes a field goal, the score hasn't really changed because his team has gained two points but lost possession of the ball, while the other team has gained possession. The two points sort of cancel each other out. If the other team just goes back and makes another field goal, we're back to where we started. But if somebody steals the ball, he is really making the previous field goal count.

According to my magic formula, possession of the ball turned out in most games to be worth about .6 of a point, so you got some credit for field goals too. As I said,



Student Knuth and Coach Heim, inputting player ratings into an IBM 650 computer. In 1960, Don Knuth was student manager of the basketball team of Case Institute of Technology. Knuth developed a formula for rating each player. Coach Heim used Knuth's formula and Case's Rough Riders went on to win the league championship.

I don't really believe in that formula any more, but it did include all the statistics like steals, fumbles, etc., and you could plug into it and get a number. The coach liked these numbers. He said it did correlate with what he felt the players had contributed; and the players on the team, instead of competing for the most points, tried to get a good score by this number. The coach thought that was a good thing. It was written up in *Newsweek Magazine*, and it was on Walter Cronkite's Sunday News. They sent a cameraman out to take a picture of me taking the statistics and feeding them into the computer.

TYCMJ: *Did you receive offers to become a consultant for professional teams?*

Knuth: There was some talk about the Cleveland Browns wanting to use computers already in those days, but I never was involved with that.

(Part 2 will appear in the March issue.)

Integer Matrices Whose Inverses Contain Only Integers

Robert Hanson



Robert Hanson is Professor of Mathematics at James Madison University, Harrisonburg, Virginia. Dr. Hanson received his Ph.D. degree from Virginia Polytechnic Institute and State University under the direction of P. H. Doyle. His interests include algebra, topology, and problem solving.

If a square matrix and its inverse contain only integers, the matrix will be called *nice*. A simple method for constructing nice matrices will be given and some of the uses of nice matrices will be discussed. Then a proof of the validity of the method will be given. Finally it will be shown that this method does in fact generate *all* nice matrices.

The following method shows how to construct nice matrices.

(1) Form a triangular integer matrix A with all zero entries below (or above) the main diagonal, with elements on the main diagonal chosen so their product is ± 1 ; for example,

$$A = \begin{bmatrix} 1 & -2 & 3 \\ 0 & -1 & 4 \\ 0 & 0 & 1 \end{bmatrix}.$$

(2) Let θ be any elementary row or column operation other than multiplying any row or column by a constant other than ± 1 . Any such operation may be applied to the initial matrix and may be followed by a similar operation as often as desired.

Example 1. Let θ_1 : multiply row 1 by 2 and add to row 2. Then

$$\theta_1(A) = \begin{bmatrix} 1 & -2 & 3 \\ 2 & -5 & 10 \\ 0 & 0 & 1 \end{bmatrix} = B.$$