PRESSZOOM
GLOBAL NEWS SERVICE

Today is **February 20, 2006**

Princeton professor foresees computer science revolution

At the annual meeting of the American Association for the Advancement of Science, Bernard Chazelle, professor of computer science at Princeton University, plans to issue a call to arms for his profession, challenging his colleagues to grab society by the lapels and evangelize the importance of studying computer science. According to the most recent data available, the top 36 computer science departments in the United States saw enrollments drop nearly 20 percent between 2000 and 2004.

(PressZoom) - "The big paradox is that the computer science revolution is just unfolding," Chazelle said. "Why, then, are students are running away from it; why is there this decline when the field has never been more exciting?"

Chazelle will be on the "Computer Science Behind Your Science" panel at AAAS on February 17. His presentation is titled "Why CS Theory Matters." In the following Q & A, Chazelle discusses the promise and potential of computer science.

What is computer science?

Most people think of computer science, if they think of it all, as being something useful -- the way that, say, people think of plumbing as being useful. And it is useful, obviously. But it is much more than that.

Such as?

Computer science is not just about gaming, not just about the Internet. Computer science theory offers a profound window through which to view the world. Computing promises to be the most disruptive scientific paradigm since quantum mechanics. It will transform society in profound way.

Isn't computer science really just a stepchild of mathematics?

As the recent breakthroughs on Fermat's Last Theorem indicate, the field of mathematics has never been more fertile with new ideas. Mathematics is original and deep, but it does not force you to think differently. If a math giant from the past –- someone like Gauss – were to come back to Earth, he would have a lot of catching up to do but he would find that math is done much the same way that it was done during his life.

Computer science, by contrast, is a new way of thinking, a new way of looking at things. For example, mathematics can't come near to describing the complexity of human endeavors in the way that computer science can. To make a literary analogy, mathematics produces the equivalent of one-liners – equations that are pithy, insightful, and brilliant. Computer science is more like a novel by Tolstoy: it is messy and infuriatingly complex. But that is exactly what makes it unique and appealing -- computer algorithms are infinitely more capable of capturing nuances of complex reality in a way that pure mathematics cannot.

What exactly is an algorithm?

An algorithm is not a simple mathematical formula. It is a set of rules that govern a complex operation. You can look at Google as a giant algorithm. Or you can think of an economy or an ecological system as an algorithm in action.

Physics, astronomy, and chemistry are all sciences of mathematical formulae. The quantitative sciences of the 21st century such as proteomics and neurobiology, I predict, will place algorithms rather than formulas at their core. In a few decades we will have algorithms that will be considered as fundamental as, say, calculus is today.

If you are right, then why aren't students clamoring to major in computer science?

With the dot.com bust, kids got scared and went to law school. Computer science departments have lost students all across the country. That's why we saw Bill Gates in recent months galavanting across North American campuses, trying to turn kids on to the field.

But mainly I think that computer science lacks a great popularizer – someone who can describe to a wide audience how exciting it is. More than 25 years ago the book Godel, Escher, Bach, by Douglas Hofstadter, got a whole generation of people excited about the future of computer science. Computer science as a field doesn't have anyone the way that, for example, physics has Stephen Hawking – someone who in a sustained way explains to the broader public the beauty and wonder and potential of the field.

If you study computer science, doesn't that mean you will simply be a programmer when you grow up? And aren't all the programming jobs being shipped overseas? As a practical matter, what kind of job can you get if you study computer science?

First, computer science is integral to all of the sciences. Biology, for example, is very quantitatively driven, so a computer science background is imperative.

At Princeton I am part of a pioneering course developed by the eminent geneticist David Botstein and others. The course simultaneously incorporates physics, molecular biology, chemistry, mathematics, and computer science. Mathematics has long been the lingua franca, the Esperanto, of science. But I would argue that science now has two Esperantos: math and computer science. Science magazine recently ran an article listing all of the interesting scientific problems of the 21st century. Not once did the article use the term "computer science"; yet many of the problems listed were fundamentally about computer science.

Second, for those of an entrepreneurial bent, the Internet is paramount; if you don't understand computer science you are lost. I don't think it is just coincidence that two of the biggest Internet visionaries -- Jeff Bezos of Amazon and Eric Schmidt of Google -- are products of the computer science and electrical engineering departments at Princeton.

Third, and (since I am a theorist) most important, are careers in the field of theoretical computer science. Theoretical computer science would exist even if there were no computers. Computer science is not bound by the laws of physics; it is inspired by them but, like mathematics, it is something that is completely invented by man.

A few short years before Einstein turned our world upside down with his theory of relativity, the great Lord Kelvin declared that "There is nothing new to be discovered in physics now." Not his lordship's finest hour.

I think that computer science bears an uncanny resemblance to pre-Einstein physics. Moore's Law – Gordon Moore's prediction that computing power would increase exponentially because the number of transistors on microchip would double every 18 months or so – put computing on the map. But algorithms are going to unleash computing's true potential. I predict that there will be an Einstein of computer scientists. The revolution is yet to come.

Contact: Teresa Riordan
triordan@princeton.edu
609-258-9754
Princeton University