

People of ACM - Robert Sedgewick

September 12, 2019

Why did you and Philippe Flajolet decide to write Analytic Combinatorics?

At the outset of our careers in the 1970s, there were very few textbooks in computer science. In particular, although the first three volumes of Knuth's *The Art of Computer Programming* were full of fascinating information, most students need preparation in mathematics and computer science to appreciate it. We soon realized that the basics would fill one textbook (our book *An Introduction to the Analysis of Algorithms*), and that we had a great deal to learn about the advanced material. Through two decades of research, mainly by Philippe and collaborators, we came to understand that general patterns were at work, and a calculus for studying discrete structures of all kinds emerged, with applicability to statistical physics, chemistry, genomics, and many other fields, not just computer science.

How does all that relate to algorithms and the effectiveness of computer programs?

Analytic combinatorics provides a mathematical basis for what I have lately been calling "algorithm science." The approach goes back to Turing and von Neumann, was popularized by Knuth, and is also the basis for my *Algorithms* books. The idea is to apply the scientific method to the study of computer programs. We develop mathematical models for the essential characteristics of algorithms; use them to develop hypotheses about program performance; validate the hypotheses with experimental studies using actual implementations and realistic inputs; and iterate the process to improve the algorithms and implementations, to the point that practitioners can use them with confidence.

In your many years of experience writing computer science textbooks, what is an important insight you've learned that you would share with a first-time textbook author?

I'm glad you asked this question, as I think that there is a need for more CS textbooks. Web content and other modern artifacts are important, but I believe a textbook written by an expert who is trying to lay out what a student can reasonably learn about a subject in a semester is still a critical component and a basis for effectively disseminating knowledge. Authors should always keep this goal in mind.

In a recent interview, you said that many academic departments could double the number of their CS faculty to meet enrollment demand. Aside from this, what is one way that computer science education at the university level could be improved?

Actually, enrollments could increase by a factor of four or five at many places. I think the main place where there is room for improvement is in educating all students, not just CS majors (and they need to know much more than just how to code). That is the primary goal of our recent book *Computer Science: An Interdisciplinary Approach*. It is intended as an introduction to the field for any college student, like the widely-used texts in calculus, physics, chemistry, biology, psychology, economics, and other disciplines. Nowadays, one could argue that most students need three or four courses in CS, and this is an important first step toward that goal. Broadening the base for first- and second-year classes also enriches the experience for CS majors, as they see the importance of applications in other fields.

What is the most significant way in which MOOCs, and other online platforms, will transform education in the coming years?

Let's just talk about online videos versus large live lectures. There is an exact parallel with streaming TV content. It used to be the case that consumers had to watch what TV networks wanted them to watch, when they wanted them to watch it, with ads, no ability to pause, skip, or rewatch. Now, the consumer chooses what to watch and when and where to watch it, and the market and demand for it have literally exploded.

The same potential exists for education. Large live lectures can serve only a very narrow subset of the population (and not as well as online ones, where students can vary the speed, pause, and rewatch). Others have had to work around or do without. With online lectures, anyone interested in learning the material can benefit from the very best lecture content available.

At the outset, I thought of our online videos and web content as important enhancements to the textbook model of the 20th century that teachers around the world can make use of to accommodate the learners at their own institutions. I continue to believe that, but we're now seeing that large numbers of individuals are learning from the material, outside of any institution. The idea of access to education, when and where you want it, is an extremely powerful one. Who knows where it will lead?

Robert Sedgewick is a Professor of Computer Science at Princeton University, where he was the founding Chair of the Department. His research interests include analytic combinatorics, algorithm science, curriculum development, and innovations in the dissemination of

knowledge. Sedgewick has also authored 20 books. He is best known for his series of *Algorithms* textbooks (12 books in four editions covering five programming languages).

Sedgewick was named an ACM Fellow (1997) for seminal work in the mathematical analysis of algorithms and pioneering research in algorithm animation. He was named the recipient of the **2018 ACM Karl V. Karlstrom Outstanding Educator Award** [↗](#) for developing classic textbooks and online materials for the study of algorithms, analytic combinatorics, and introductory computer science that have educated generations of students worldwide. This year he also received (with the late Philippe Flajolet) the Leroy P. Steele Prize for Mathematical Exposition for the 2009 book they co-authored, *Analytic Combinatorics*.

People of ACM Archive [↗](#)