Title	A Journey Through Computer Science
Author(s)	Andrew Chi-Chih Yao
Language	English
Event title	The 2021 Kyoto Prize Commemorative Lecture
Publisher	Inamori Foundation
Issue Date	10/01/2022
Start page	1
End page	7
URL	https://www.kyotoprize.org/wp-content/uploads/2022/10/2021_yao_en.pdf

URL for Japanese translation: https://www.kyotoprize.org/wp-content/uploads/2022/10/2021_yao_jp.pdf

The 2021 Kyoto Prize Commemorative Lecture
Andrew Chi-Chih Yao

A Journey Through Computer Science

Ladies and gentlemen, it's good to be here.

First of all, let me say, it is a tremendous honor to receive the Kyoto Prize. Knowing all the previous recipients and their brilliant achievements, I feel deeply humbled as to be considered worthy to join their ranks. It is a great pleasure and privilege for me to speak here today.

I would like to tell a little bit about where I came from, how I found computer science, and my journey through it all. In more detail, I will begin with a little bit about my background, then my early enchantment with physics, which led to my first career, and then how I switched fields by accident to become a computer scientist. Then, I would like to give a synopsis of some of my work; what problems I consider, and why they are interesting. And finally, I would like to pay tribute to several people, who have greatly influenced my life and work, and then, I'll wrap up.

I was born in Shanghai, China, in 1946. My family soon after moved to Hong Kong, and then, Taiwan. I grew up in a happy middle-class family. I had loving parents and two very close siblings. I was brought up with an emphasis on traditional Chinese values. In particular, it pays high regards to culture and learning. Happily, and to my parents' delight, I was an excellent student, straight A, throughout my schooling. I remember I loved math, science and history as a kid. For example, I was fascinated by historical figures who showed unusual bravery and wisdom. Great scientists like Galileo and Newton, they were my heroes, also in just the same way. They gave me a sense of awe and magnificence because of their brilliance and their courage to stand up for what they believed in. And I dreamed that this would also be my destiny someday.

1. Early Enchantment with Physics

In the third year of my high school, I came upon a copy of notes by Lord Eddington on the theory of relativity. It gives the most vivid and simple derivation of relativity, and the argument goes roughly like follows: Experimentally, we have known that light has constant speed. And from this fact, one can derive with clever argument that our familiar concept of time cannot be an absolute universal concept, and that was something that everybody had taken for granted for a long time. And this argument impressed me greatly. Physics can be read like a detective story, and more imaginative than any of the clever episodes in Sherlock Holmes. I was really impressed and encouraged.

In 1963, I went to college as a major in Physics, and not long after that, Feynman's lecture notes on physics were published. And the legend has it that Caltech wanted to radically reorganize their freshman course in physics, and Feynman agreed to do it with a condition that he would only teach it once. And this led to the legendary three-volume series of The Feynman's lectures on physics.

And reading the book, it was an eye-opener for me. Advanced concepts that are difficult to explain turned out to be possible to explain and derive, using only elementary mathematics. That was really impressive. That led me to see the depth of physics and its beauty. And actually, this was the first time that I felt that I really understood the principles of quantum mechanics. And 30 years later, when I went to work on the field of quantum computing, Feynman's explanation on quantum phenomena still looks to me to be the most illuminating and useful explanation. And that really settled it, I decided to study physics after college.

In 1967, I graduated from college, and after one year in the military service, I went to Harvard

University to do graduate study in physics. And in 1972, I obtained my Ph.D. degree in physics under Professor Sheldon Glashow. And finally, I was a certified, real physicist.

2. Transition: From Physics to Computer Science

That didn't last long, though. In 1973, my wife, Frances, who was a Ph.D. student at MIT at that time, introduced me to algorithms. The word algorithm was, at that time, quite unfamiliar to most people unlike today, and now, it is in everyday vocabulary. And an early draft of The Art of Computer Programming, Vol. 3, by Professor Donald Knuth, and that's a book on algorithms, it was having limited circulation. And, what an amazing masterpiece! It introduces a fascinating new science. And after reading the materials, I could not stop thinking about the research questions raised in the book.

And it became an obsession so much that I soon quit my postdoc job in physics in order to pursue graduate study in computer science full time. And I remember that my mother was really concerned because it seemed that I had given up all these years of work in physics. But my wife was very supportive, and so, I became a computer science graduate student at University of Illinois. Thanks very much to Professor C. L. Liu for willing to take me on.

3. Synopsis of My Work

Now, I would like to tell you a little bit about my work. Initially, I focused on solving existing open problems in algorithms, such as minimum spanning trees, B-trees, etc. But not soon after I graduated, I started to get interested in developing new frameworks and new theories for computer science. And over several decades, I had the opportunities to work in several top-rated universities and I have spent 10 years at Berkeley-Stanford, followed by 18 years at Princeton, and in 2004, I moved to Tsinghua University, where I now work.

In each period, I was doing somewhat different things. And it was quite interesting that the topics that I focused on during these different periods, they had a lot to do both with the change in times and the development of the computer science as a discipline and also with the environment in the universities that I encountered.

And what I would like to do is to pick the three topics, the minmax complexity, communication complexity, and cryptography and MPC.

Now, the way that I find best to do research is to find insightful, bold questions that are important. And if you find good questions, then you invariably will do good research, finding nice results that are applicable and important to the research world. And so, what I would like to do is for each topic I will describe the motivating questions that led me to the discovery and why they are important.

3.1 Minmax Complexity

The first one is about minmax complexity in 1977, and this holds a special place in my heart because this really was the first significant chance that I formulated my own questions and found good ways for dealing with it.

And now, we know that algorithm is essentially equivalent to some sort of a recipe, for example in cooking, you would say that I would put three ounces of salt and some grams of meat, step by step.

And in the mid-1970s, a new style of algorithms came to the attention of people, namely that is called the randomized algorithms. In this novel type of algorithms, they incorporated stochastic moves, so, metaphorically, it's just like that, instead of having a definite procedure of putting two

spoonfuls of salt, you say that I can throw a coin and decide whether to put two spoonfuls of salt or to put a cup of red wine into the cooking process. And so, for the traditional way of thinking, this looks like a crazy way to do things, but in the 1970s, people have demonstrated that actually there are advantages in proposing, in carrying out algorithms in this fashion and they led to some spectacular results in some cases.

But the question that people weren't able to understand to analyze is that, what's the limitations of these algorithms? And so, this led me to the question that I asked myself: Which is better? Is this the randomized approach, just proposed, or it actually would be better just to take the traditional approach of looking at average case behavior, just looking at the data distribution and tailor your algorithm for doing it? But once you phrase the question in this fashion, then, a most pleasing connection leaps out that gives a lot of insight into randomized algorithms.

When you look at the randomized approach versus the traditional distribution approach, you can regard it as a game played between algorithms and data (Fig. 1). The algorithm would choose how to make stochastic moves, while the data, the adversary, can try to pick the distribution to make the life of the algorithms more difficult.

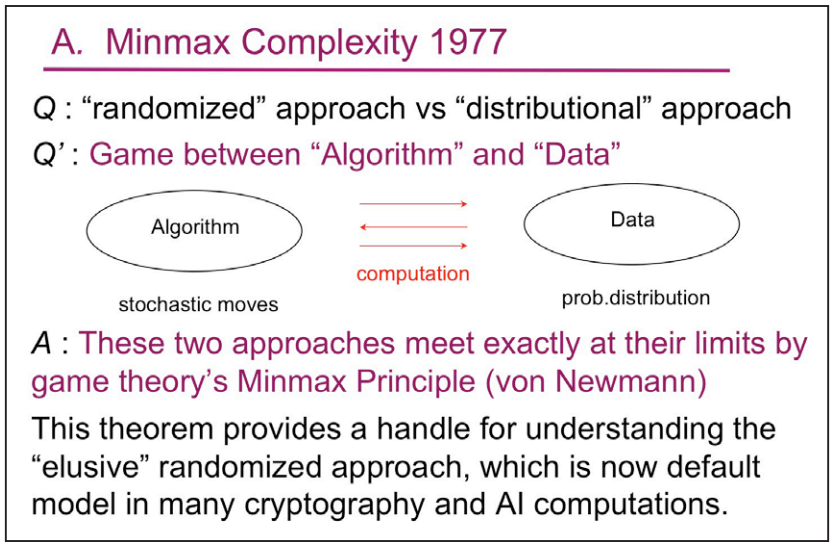

Fig. 1

Now, these two approaches meet exactly at their limit, by the game theories, minmax principle due to von Neumann. And so, this connection gives us the theorem that we would like to prove, namely that these two approaches are the same, and it provides a handle for understanding the randomized approach.

And the important thing is that these novel types of algorithms at that time now have become the default model for many cryptographic and AI algorithms, and there are reasons that people now want to understand the limitations of the randomized algorithms. And so, in over 40 years, the algorithms that I have found are still regularly used by many researchers in solving their problems.

3.2 Communication Complexity

And the second topic is communication complexity that I raised in 1979.

Let me explain the mathematical problem first. Alice and Bob are two people who are in separate locations, and they each hold a piece of data, say, x and y. These are n-bit integers respectively.

And the question we would like to solve is that, suppose they would like to compute jointly some quantity, F (Fig. 2). How many bits need to be communicated between them? And that's now called the communication complexity of this particular function.

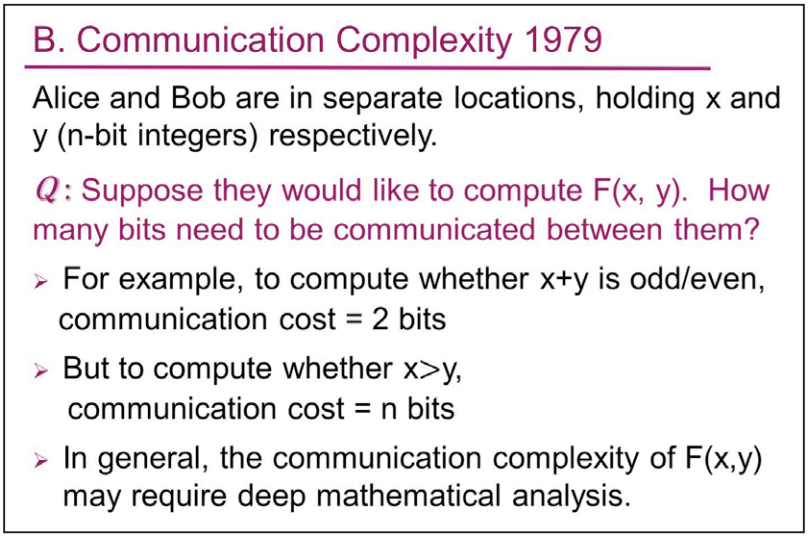

Fig. 2

Now, this of course depends on what functions you are computing. For example, to compute whether the sum of these two integers is odd/even, clearly, you only need two bits of communication. So, each one just tells the other even or odd, then they can decide on the answer. On the other hand, if you want to compute something whether x is greater than y, then you would take n bits and you can see that the natural algorithm you really have to send the entire stream from one person to another in order to solve the problem.

And the deeper part is that you have to realize and prove that there are no better ways to solve this problem than just doing it in this naive way. And in general, it's quite a difficult problem. If I give you a particular computation F, it usually requires quite deep mathematical analysis to do it.

And the reason for considering this problem is because, during the late 1970s, it becomes clear that there is a change in paradigm of computing from the mainframe computing everybody was familiar with before that time, gradually moving into network computing that we are now familiar today, namely that people are interested in solving problems in distributed way and many people would like to collaborate to solve a problem. And so, this means that we have to adjust the model we used to have about computing into the network model of computing.

In this new world, the communication cost is often the expensive part, namely that because you have to move data around and that part is the most expensive part. And so, the concept of communication complexity, as I just described to you, is to model and reflect this change in the paradigm. And since this model is proposed and analyzed, communication complexity has found wide application in areas ranging from chip design to data streaming.

3.3 MPC and Millionaires' Problem

And the last topic I will discuss in some detail is about cryptography and MPC. And in 1982, I wrote three papers, and this became a significant contribution to modern cryptography. The three papers deal with the Dolev-Yao security model, pseudo-random number generation, and multi-party secure computation, or MPC. I will just address this last one.

MPC is a cryptographic concept that enables computation to be done on encrypted data. If you use MPC, it's possible to have multiple databases do any joint computations without leaking its own data, namely that essentially, it's saying that we can share data without seeing them.

And let me explain it in a special case, then I think it will become clear. I will use the well-known example of the Millionaires' Problem that was also raised in the paper. Two millionaires, Alice and Bob, they wish to figure out who is the richer person without revealing any quantitative

information (Fig. 3). So, Alice has x millions, Bob has y millions, so the mathematical question is, they want to talk to each other to decide whether x is less than y. The question is, is it possible to carry out a conversation so that, in the end, both Alice and Bob will know the answer who is richer but without knowing anything else about the data on the other side.

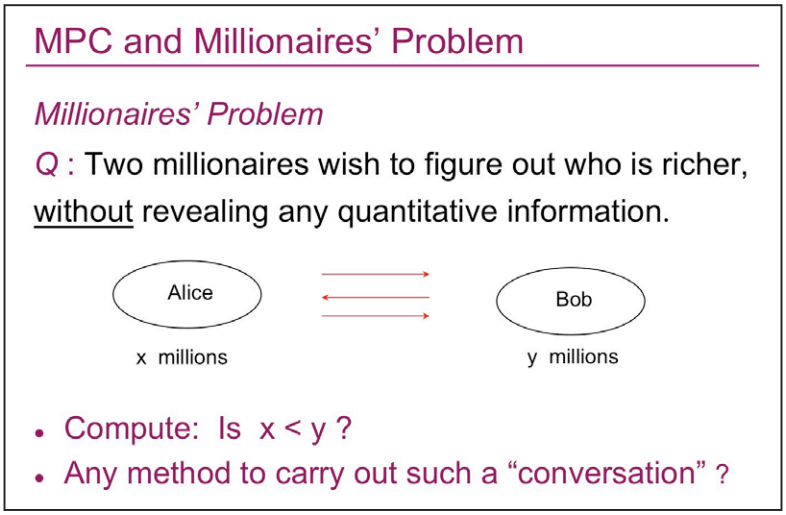

Fig. 3

And naively, intuitively, you would think it's impossible. How can I find out something about who is richer without revealing any information from either party? And, if you think about it for a few minutes, you will realize that it is impossible if you adopt the standard security definition in those, in 1982, namely the Shannon's information theory, and you can prove it is not possible under that model.

But I think that necessity is the mother of all inventions, and if you really have to do it, you will think of a way. So, if you think outside the box, then actually it turns out that it is possible. And, by "outside the box," what we mean is that you should discard the very rigid and tight box that Shannon constructed in this case. And you bring Alan Turing into the picture. I will not say too much about it, but basically, it turns out that if you relax the security definition, but still a pragmatically very good definition, then it can be accomplished.

And in particular, I showed it, this can be accomplished using what is now called the "garbled circuit." During the past, nearly four decades of development, advances have been made in hardware and algorithms that it's now becoming nearly feasible. And there are many research works in this regard, ready for work in Fintech, data-trading, and joint drug discovery.

3.4 Other Topics and Overall Summary

And so, there are other subjects that I work on, I'm not going to go into details, I just list them here: Quantum Computing, it's a revolutionary approach with promising exponential speedup; Auction theory, they would cast amorphous economic problems in the neat game-theory framework; and Artificial Intelligence, witnessing the incredible feat achieved by machine learning algorithms like Alpha-GO, but the reason for success is still mysterious.

So, all these were very interesting new areas and still under development.

As you can see, I have worked on quite a number of different subjects. And these diverse and colorful subjects, they actually reflect not just my own personal taste, but it's really about the blossoming of information science over half a century, and the growing interdisciplinary connections that we see today.

4. My Inspirational Mentors

Finally, I would like to say a few words about the people that I have met. Now, during these many years, as a computer scientist, I had a good fortune of encountering many extraordinarily talented people. I was especially fortunate to have had two inspirational mentors: Professor Sheldon Glashow and Professor Knuth.

Professor Glashow was my physics Ph.D. advisor at Harvard University (Fig. 4). He was one of the group of people who first predicted that there exists a new particle called the charm quark, and he was its most enthusiastic advocate.

I learned from Professor Glashow that in science, you have to be bold, and you have to be persistent in what you believe in. And, another thing that I learned from Professor Glashow is that mathematics is different from physics.

For physicists, it's most important to be able to find out the truth about the physical reality, rather than to insist on precise mathematical arguments. And I think that pragmatic spirit helped me a lot in my research afterwards.

There's one other thing that's important that I learned from Professor Glashow: Life should be fun. As a young student, in the spring of 1971, I tagged along on his sabbatical to CNRS at Marseille, France. And, what a marvelous and charming city! And that was also my first trip to Europe. Later that summer, he took me to a summer school in Sicily, Italy. And again, a most wonderful experience. It's a very important lesson that Professor Glashow has shown me that joy for life and intellectual pursuit can go hand in hand.

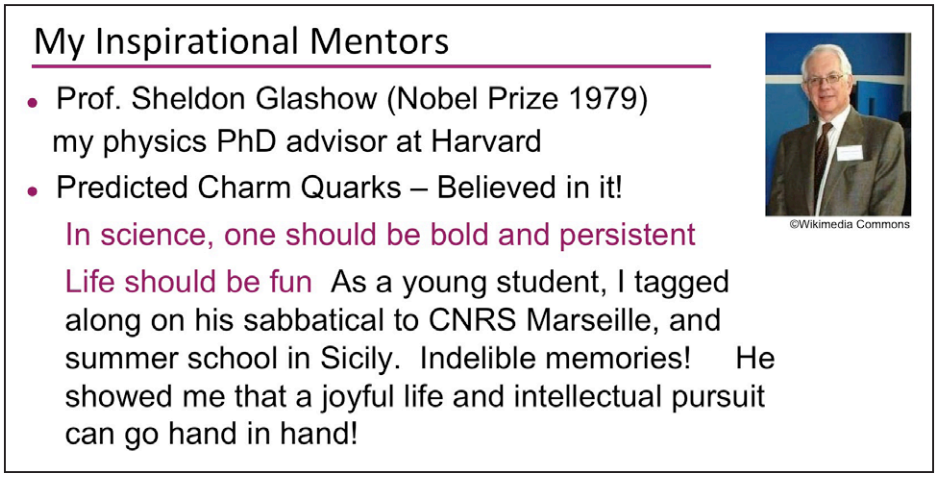

Fig. 4

Now, I would like to mention Professor Donald Knuth (Fig. 5). As I said before, his book, when I read The Art of Computer Programming, it virtually changed my life.

In this master book, he literally created a new field of study that also has inspired generations of new computer scientists. For example, I began my career in computer science by reading his book and solved some of the problems that he masterfully formulated.

And later, I was lucky to become his colleague at Stanford. It is well-known that Professor Knuth is good in many things besides mathematics and computer science. He is an accomplished pipe organ player. He is also a composer, a fiction writer, among many other things. So, he is very multi-talented, but yet, he is sincere and generous, and he always sees the good things in other people.

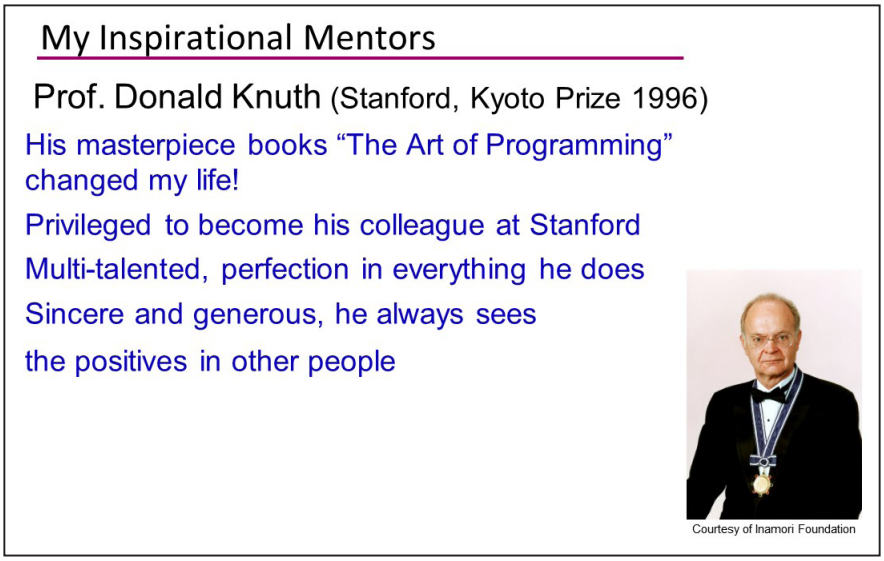

Fig. 5

5. Concluding Remarks

And to conclude, I have had a wonderful journey in computer science with many twists and turns.

And I have found out that actually starting on the wrong foot may not be a disadvantage. Actually, the early training in physics turned out to be most helpful to me in at least two regards. First, I learned what good theories look like. In physics, there are many good paradigms of theories such as relativity and quantum mechanics. And that helped me a lot later on when I needed to build theories in computer science. And the second thing I benefit from physics is its pragmatic spirit, that the important thing is that you want to solve the particular problem at hand. It doesn't matter what method you use. You should use, learn or invent as the case may be, with the goal that in the end the problem is solved.

In science, the paradigm is the search for truth. In this process, we sometimes discover patterns and beauty which can lift the human spirit. It also leads to innovations that can improve human conditions and prepare us for future human challenges. I agree completely with the Inamori Foundation's vision that science and humanity shall work together for the betterment of mankind.

Again, it is a great honor for me to receive the Kyoto Prize and a privilege to give this lecture, sharing my experiences with the audience.

Thank you very much.