Olympiads in Informatics, 2017, Vol. 11, Special Issue, 11–20

DOI: 10.15388/ioi.2017.special.02

International Olympiad in Informatics: Roads to Algorithmic Thinking

Interview with Donald KNUTH

The interview with Donald Knuth for the 2017 special issue of the Informatics Olympiad Journal took place in his residence at the Stanford campus on December 20, 2016. Yahya Tabesh, Amir Zarkesh and Mohsen Hejrati were present and Don generously answered all the questions. Mohsen recorded the interview and Jill Knuth (Don's wife) kindly facilitated it.

Donald Knuth a legendary in computer science and professor emeritus at Stanford University, is the author of the multi-volume work *The Art of Computer Programming*. He contributed to the development of the rigorous analysis of the computational complexity of algorithms and systematized formal mathematical techniques for it. In addition to fundamental contributions in several branches of theoretical computer science, Knuth is the creator of the TeX computer typesetting system. Knuth created the WEB and CWEB computer programming systems designed to encourage and facilitate literate programming. In 1971, Knuth was the recipient of the first ACM Grace Murray Hopper Award, he also received the Turing Award in 1974 and he has received various other awards including the National Medal of Science, the John von Neumann Medal, and the Kyoto Prize.

Donald Knuth

Yahya: *It is almost three decades that International Olympiad in Informatics (IOI) had been organized, we would like to have your impression about IOI and these kind of programs.*

Donald: It sounds like a wonderful motivation it is better than trying to get to new levels of the games. It would be good for their creative thinking for the problems they are going to face in their lives every day. While the games, I know a lot of students learn a great deal by playing, like my grandsons, they play a lot of these games that have little bit of programming. I think having the real challenge of where there are working against other students is much more valuable.

Amir: *So do you think this kind of gaming can be a good learning environment?*

Donald: Sure, and there is also other kinds of skills and programming skill. You can have good response, learning how to organize, like the football or best tennis players they program three moves ahead of the ball. Those are similar in mathematical problems as well. I get to play video games not quite often but for example there was a game, it was based on a James Bond movie, I don't want to talk about it too much but it was very interesting. There is a connection with games and programming because a lot of scientists are creating new games. And of course lots of great researchers came to how to play chess and golf but video games and some other games are in a different category.

Yahya: *It is very interesting, the advances of the Artificial Intelligence (AI) which used to be kind of more slowly going forward as far as unsuccessful applications just recently because of the computer power has suddenly opened a lot of possibilities. Even games, you can make it very responsive to what we do and figure out ways.*

Donald: Exactly, a lot of the ideas have a sort of a critical point where until you get to the critical point it looks like you will never be going to get really far and then after those more things happen. I don't know any general move so that I can tell about these critical points, and of course a lot of the most recent things are situations where probably never will be a rule that is not understandable by humans anymore, because there are a lot of things comes up with rules that just works but nobody knows why. Between a neural network that seems what you have got is kind of a hologram of the solution instead of the solution. Little parts that are scattered all over the place. I had a situation couple of years ago where I was working on the satisfiability problem and there are random heuristics that go on in this problem and the idea was to prove that a certain class of graphs could not be colored with three colors and there is mathematical proof of this but a mathematical proof that I knew if you have the graph with size N it took order N^2 in steps and $N log N$ is definitely not in, but my computer got random found refutations proofs of that it looked very much linear and so I thought oh, there is another kind of proof that I didn't know of this graph here and so I thought okay, all I will look at what the computer proof was and I will see where I would understand and I spent so many days on this and I found no idea of how to construct what the computer came up with. A lot of these heuristic methods come up with answers are probably beyond human comprehension. I also went to the famous problem of P verses NP because people say

if P is same as NP then there is a polynomial algorithm to solve the satisfiability, then all of these problems will suddenly be easy to solve. But if that, it is quite possible that there exists an algorithm but it is too hard for anybody to know and so asking somebody if there is an algorithm exists or not it is very wage and If somebody tells me yes there is an algorithm, that is not good to me unless they say what the algorithm is but we know for this problem there exist an algorithm to solve this problem but nobody has the idea that how fast the algorithm runs. There is one and you can prove that there is one but it does not tell you anything more than existence.

Yahya: *So did you see because of the fact that every kid has a super computer in hand now, that this can be the start of a change in a type of advances in computer science, kind of you look back 30 years ago and the whole volume that you wrote and the whole subject that you covered, is a change going to happen from existence proof and from more practical kind of searches, opening new doors, do you see any new transitions happening?*

Donald: Certainly there have been huge transitions in the problems that I can solve and that is what I write three problems three program a week because I can't resist and I see a problem now that I couldn't solve before. Just before you came here I thought tonight I am going to write the program. We are varying things that also getting more familiar with subject developers, so we are seeing giant steps instead of baby steps. We could understand little parts of the picture but now we are seeing enough of it that we can go

Yahya, Knuth, Amir

further even if computers are going fast. But a lot of the matters about space and speed and forgetting the space now is another reason that now I can solve. On the other hand, that can make people so lazy. We used to think when there was a problem and now we just have super computers that can solve the problems that I was not able to before and that could make you sloppy and then you can't push the envelope. The Olympiad is a push that people who are in it have to think.

Amir: *what are you programming these days?*

Donald: I use CWEB[1] which is a literate programming which is a combination of C and TeX and I could go on and on to say that this is the best possible thing but essentially the most important thing is I imagine myself not as just trying the computer solve it but I try to imagine telling a class how to solve the problem telling audience and human beings how to solve it, and so I have to explain myself and I have to get my thoughts a little in order as if I am at a blackboard or something, rather than just hacking and that discipline says I define the data structure here and maybe I should write down invariant and I declare variables as this is a number of items in this list and this verifies and gives some explanation of, and the method of the programming says that each program is made of different parts that are put together in simple ways and but you have to see the sculpture and you can start up but at the top to break it down, say at the start I will break it down to groups imagine A and B and C and then I start looking on A. Another way to write a program is to start with some routine items that I want to use some kind of library for the problem and start there and go, but in either case it is corresponds to the way my memo conception of the program and it means that I write more keystrokes and the program working faster because it is a very reliable way to organize and then I can also come back to, a year later and figure out with the other kind of programming.

Amir: *How do you see this progression of languages used to be C now, people learn Java and more recently they learnt Python, how do you see this?*

Donald: It kind of kept going, I think every generation wants to write its own proper language I don't know why this is true but I guess I know what is the truth. Because any complicated thing can be improved and if you can prove it that it is so complicated then you can improve it very simple but there is no optimum program of all of the although there is local optimum. But also one side is that because people have different intuition at things naturally. I have very little experience with Python, I had a deadline and I was giving a lecture about so if I could understand it's program and then I would be prepared for the illustration for my lecture, I have never used Python before but it is easy to understand programming, so I had an hour to look for this program and modified it and I had the other half hour to learn about Python and 5 minutes to go and present, it was pretty intuitive and I had some worse experiences later. but then I started with Java-script and other language but when I write three programs a week but there is 50

[1] CWEB is a computer programming system created by Donald Knuth and Silvio Levy as a follow-up to Knuth's WEB literate programming system.

weeks in a year and I would say 150 programs this year and I would say 140 of them in CWEB.

Yahya: *You have always talked about the art of programming, how the sense of beauty and joy could be reflected in programming?*

Donald: Well, my answer is if they start using literate programming they would be happy but really it gives me special pleasure because I get programs that works, so you get a program that I know why it works and I can see it but as something that makes a nice story, telling a nice story is really important for communication between people and I think of programming as communication between people. Now it is the communication of person to machine (and not person to person), that gives you joy to tell the true story.

Yahya: *We are using the libraries in Python or C and they can help to develop more fast and do better programming but somehow for the beginners I am not sure if they should go through these libraries and how they could be creative and how good will develop their own ability in programming.*

Donald: If programming has only such a situation that let you every week that you write a new, you get another library instead of new start doing that so your job become stranded to applying only parameters and let somebody else make the definition, so you look only on what is on the menu for this package and if that is the fact, then it is hard to sustain creatively. But on the other hand, if you got a powerful library and you're going to reinvent the wheel all the time if not using it, and so the other canned programs that I wrote in Mathematica[2] where I take advantage of the fact that this programs would do integrations for me and lots of over cursing calculations that would have been much harder can go easier. I can see also why it would be also exciting to have a plot full library that could make pretty picture or good music and set things going. And Java has a great graphic designer and certainly I want to have a rich library of tools but for my own. For all the illustrations in my book I do MetaPost[3] which is a language for illustrations but I don't use certain things that other people packaged for me.

Yahya: *In more high level also they put API (Application Programming Interface) for various applications and just when you have the API you can use the application, of course it is good for developing new applications. I remember some guys went to put Craigslist[4] on the Google map before they released the API, it was really a hard job to*

[2] Mathematica is a mathematical symbolic computation program, used in many scientific, engineering, mathematical, and computing fields. It was conceived by Stephen Wolfram.

[3] MetaPost refers to both a programming language and the interpreter of the MetaPost programming language. Both are derived from Donald Knuth's Metafont language and interpreter. MetaPost produces diagrams in the PostScript programming language from a geometric/algebraic description. The language shares Metafont's declarative syntax for manipulating lines, curves, points and geometric transformations.

[4] Craigslist is a classified advertisements website with sections devoted to jobs, housing, personals, for sale, items wanted, services and discussion forums.

adapt the houses in the Craigslist over the Google map and they made a hard job to make it. now the API is released and they can make it easily. But where is the balance point using APIs and how should we have new creativity and innovative thinking?

Donald: If every week number of API come up, this is a full time job just knowing what they are, and not getting anything done. I started running to this ages when X Windows and different systems of applications came over and I already show the same for books and that was when the open source was coming around and for me it meant that okay, I have all these books that are open source and I could look at each of the programs and see what they did. What open source meant to me was that I can open books all the time and getting a lot of things, and I still have to look up all the time when I am writing and look up the foremost parameters every time and some of the interfaces are intuitive but each works fine for a different person so you have to learn everybody else's intuition if you wish to switch into and sometimes when you talk about a box and screen you have to give the lower left corner first and sometimes the x corners is and y corner is down. But to me, the key thing that makes computer scientist especially good, is the job levels of abstraction and distraction so good programmer would have a lot of low level things and know them in order to solve the problem and you have to have a certain kind of experience and you have to start this particular process and going from high level to low level is the top skill and people who have that are worth.

Yahya: *So you mean in a modular way?*

Donald: Just in general to understand the instruction but there is another level that this package may have and something could go wrong at this step then you have to understand and when you are a creative programmer to figure out what is the problem.

Amir: *So they can leverage the abstraction of programming and they can zoom in?*

Donald: Sort of not knowing what even to do. Yeah, this is a difference with mathematics and mathematician, they need to know one or two rules that governs the work on different levels but in computer programming there are several steps and each step is different so in dealing with programming those two skills are somehow the most important.

Yahya: *So for high school, mathematical learning in high schools, somehow I had the feeling that this kind of step by step whatever you say, problem solving, algorithm thinking is missing, they have just focused on the other kind of mathematics but as you mentioned this other kind of mathematics also is needed.*

Donald: What do you mean the other kind of mathematics?

Yahya: I mean, as you mentioned they just go toward calculation, and not kind of algorithmic thinking.

Donald: Because you know there are a lot of kinds of mathematics so there are some kind of mathematics that are closer to computer programming but people who try to do

computer programming with little understanding and idea of proof and that may not get them to deep understanding.

Yahya: *But in learning programming, they may just focus on syntax, not the problem solving abilities and computational thinking and algorithms.*

Donald: Yeah, I saw a quotation couple of years ago that mathematics without proof is like soccer without ball or football you know. Well, and computer programming are also informal proofs all the time and you know explaining why what I am doing is supposed to work and that is not necessary for the very simplest programming and is not necessary for the plugin and libraries you may use. But still on using that I have to know for example what is the purpose of the sorting and what are the consequences, and if I sort a set of data, I can look at them in a certain way and I will know that what I am trying to solve is helpful, so I will look at something that officially requires to follow but I have to look at it in a certain way.

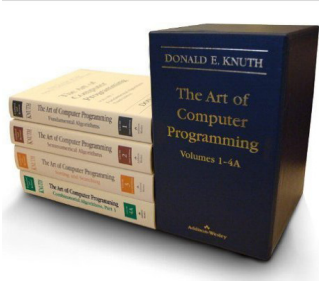
Yahya: *You mentioned about the open source, what do you think about the open source movement, it brought a lot of opportunity but how we can manage it?*

Donald: Well, if open source is opposite of closed source I am in favor because I don't like secrets. In black box, they cannot investigate why something works and if we cannot build on most ideas if everybody kept everything to themselves then everything would advance in a much slower rate and I guess your next question from me is about patents. I don't think that develop the creativity, because after people started having patents on trigger idea that you expected students to do as homework, now I had to pay for permission to use them, so all these things I think are problematic and these become intellectual property. There are also a lot of wonderful ideas in these commercial programs and they have to pay for because they cannot be discovered and so they deserve some credit for doing it. I benefit a little bit form this because my books years ago, I had to write the books once so over the years and I got more rewards for writing the books and so maybe I should feel guilty about that like this organ here that was, I couldn't have that without having written books.

A comprehensive monograph written by Donald Knuth that covers many kinds of programming algorithms and their analysis.

Knuth began the project, originally conceived as a single book with twelve chapters, in 1962. The first three volumes of what was then expected to be a seven-volume set were published in 1968, 1969, and 1973. The first installment of Volume 4 (a paperback fascicle) was published in 2005. The hardback Volume 4A, combining Volume 4, Fascicles 0–4, was published in 2011. Additional fascicle installments are planned for release approximately biannually; Volume 4, Fascicle 6 ("Satisfiability") was released in December 2015.

The collection has been translated to at least ten more languages.



The Art of Computer Programming

Amir: *Wow this is fascinating; do you play a lot?*

Donald: Yeah, I am actually working now a dream for more than 40 years to write a major piece of music for organ and I am in the process of finishing it now.

Amir: *Aare you using math in it more or are you just using it as art form?*

Donald: There is math in it but only if it sounds good. I read a page about it, I am just preparing to leave for Sweden and where else and I am going to give a talk in Stockholm to royal college of music. The Swedish people are interested in crazy new ideas.

Amir: *You mentioned something very interesting, you said that programming to you is more of communication and storytelling. Can you explain it a little more, because I think that for education is important, to get the feeling?*

Donald: I think also Stanford finds that very important, thing is that students learn from each other and instead of having students keeping stuff they learn from each other, one student showing another how to do something as a teacher, it is very viable for the student who is doing the teaching.

Of course there is a tendency now for a student to "ok, show me how to do this" and then this student would never learn, but the teacher at least learns. And that's, you learn much more if you view a program if you just do the program as something that writing it down for the computer to process. The reason is it has to make more sense. I cannot make this as a real strong point because certainly if I as a person to say it is okay so I am going to reformulate my thinking but when I am writing a program, the idea is that I am trying to explain, I get it right much more often than if I just run a program and that's because I force myself to explain it and I guess it would help.

Donald playing organ!

Amir: *Recently there have been mentioned to the point that this, the type of programming and and computational thinking behind syntax is important for everybody to learn and not just those who want to become programmers. What do you think?*

Donald: I think It is better to understands more of the processes with a mental model, because it makes more precision, suppose I am trying to learn music, theatre or some better that other notes. So now let me try to imagine that from the stand point of computational thinking, so how could I write a program and give me a bunch of notes and my program would say that this sound is pleasant or not, well there is no real answer to this because what you want to do, any combination might be the right answer, but if you want to know does it sound like Mozart or Tchaikovsky or does this sound like or some other people this is a good question where you can learn much more music by asking that question and thinking about it as a computational process deciding whether something is good or not. I have examples from chemistry or law or poems or English I mean, but there is a way to think about these things as computation and I think that leads to, it makes it easier for people to understand what they don't know and then they can learn more as they try to answer these questions.

Amir: *How could we make this computational thinking approach more popular for high school kids?*

Donald: It is going to take some generations. You can get, one more becomes part of peoples for doing things and how they do or talk to each other about these things. But until you get to that to level it will be specialized. I think things like Olympiad are a way to reform everybody, it would take a long time even for the teachers to understand but maybe you can automate it and just have enough things but it gets few great teachers to explain it and that be everybody loves and it is not going to also happen unless they have best friends who do it.

Amir: *Maybe instead of classes and teachers put in the games that teaches them.*

Donald: Yeah, if a little bit of more computational thinking every year than the past year, it would go. But I doubt that if we assume that almost everybody on the block does. Right? Instead of not just the ones who are Olympiad champions and stars. I personally don't think we are going have secure system if everybody is going to have everything, I disagree with the fact that everybody can learn anything. Because many things I have tried to do and I finally said to myself that well, this is not going to be easy for me and same I am explaining to people about programming. Some of them were even very motivated but they were good in other things. So I am not, here I am talking about programming at the highest level, not at a useful level. But when we hear, but it certainly would be wonderful if enough people had computational thinking and learn how to use it. Everybody is supposed to know 1/3+1/4, but I don't know how many people really do this. It is what you are learn in the process about the structure of that there is such a thing as fractions and understanding of that helps. If you understand that then you have that what you can do. But everybody needs to learn problem solving, even I may say dogs learn how to solve problems when a dog learns how to climb a fence, so problem solving is like that.

Yahya: *Just go back again to Informatics Olympiad and have some sort of final words more about this IOI and for these kids around the world how we can attract them and when they got attracted how we can keep them go further?*

Donald: It all depends on the quality of problems and that's great for the people to come up with the problems and so to me that is the most critical thing, is to generate experts in creating the problems. Because if we have good problems it would be irresistible for us.

Yahya: *How we can make it more popularize and attract more people?*

Donald: I suppose you have to get into that somebody let people admire and show that they are interested. So it goes with, I am sure each country work, how much they have tennis playing, golf playing and chess playing and programming and flying drones and robots and make paintings, each culture has themes that they are currently hot and I am not a sociologist but the more you can try to something that makes it cool to do this, and sometimes people are good programmers but that doesn't motivate, so whatever solution is, it is one of the beauties that we didn't know about it in the past.

Yahya: *We highly appreciate your time and for this enjoyable talking.*

December 20, 2016