



?

HOW ALGORITHMIC THINKING CAN HELP YOU THINK SMARTER

ALI ALMOSSAWI

As far as I can tell, there isn't a discipline in engineering that doesn't see efficiency as a favorable quality. And engineering is in several respects a reflection of life.

With software, there are at least two broad means by which efficiency is achieved: data structures and algorithms. Anyone going through an undergraduate course in computer science will have taken a data structures and algorithms class.

And in such a class, the impressionable student will have been told that an algorithm is a series of unambiguous steps that achieves some meaningful objective in finite time. The series might begin with some input and is expected to produce an output. Those are an algorithm's characteristics.

What's fascinating is that Babylonian tablets from the second millennium BCE reveal that ancient Babylonians wrote down their procedures for determining things like, say, compound interest or the width and length of a cistern given its height and volume using algorithms. All throughout history, and in a variety of domains, one can see approaches to problems that resemble what we refer to today as algorithms.

That realization is intriguing for a number of reasons. One, it shows that this way of thinking about problems is rooted in ancient history. Two, it shows that it is domain-agnostic.

And so, if one were to consider how best to make algorithms compelling to the broadest audience, it seems only natural to strive to not sell the field short, by describing it in its narrowest form, but to rather frame it as a tool for thinking, and a general-purpose one at that. One that can be applied to everyday problems that may have nothing at all to do with computers.

To define solutions to problems using the characteristics mentioned earlier is all well and good, but what's more useful is knowing how good a solution is by not only considering if it returns the correct result, but also how it arrives at that result. Put differently, how efficient is it? That goodness, that efficiency, can be measured using two types of approximations.

“All throughout history, and in a variety of domains, one can see approaches to problems that resemble what we refer to today as algorithms.”

Seeing the World Differently

Firstly, rates of growth. How is my way of solving a problem (of completing a task, of achieving an objective, and so on) impacted as the number of things I am working with changes?

Computer scientists use that way of thinking all the time to measure the “goodness” of algorithms. And there are ways of classifying algorithms based on different rates of growth.

And secondly, orders of magnitude. Is it useful to sometimes trade rigor for insight, by thinking about performance in logarithmic terms?

I was listening to an episode of *Radiolab* a while back, and they were interviewing a neuroscientist who talked about something I hadn't heard before, about how children might think in terms of orders of magnitude. That's why a child may be more hesitant to give away two of something versus one, because the child sees two as twice as much as one rather than one more than it.

And the useful thing about thinking in terms of orders of magnitude is that it makes comparing things more interesting. Faster or slower. More space or less space. Better or worse. Sweeter or less sweet. It's an approach you see in survey design, in user interface design, at the optometrist, at the milk tea shop, and all throughout life.

For instance, given a pile of socks, we can find pairs by picking a sock and then rummaging through the pile for its pair. Alternatively, we can put each sock that we find to one side and only pair socks when we find a match. Both methods are fast when we have a few socks. The second method is much faster when we have lots of socks.

The first method might be described as growing at a quadratic rate and the second method as growing at a linear rate.

That scenario, simple though it is, has offered a framework for thinking about a possibly mundane everyday task. It teaches us that the concept of memory—applied here when socks are kept to one side during the matching task—is useful. And it shows us, quantitatively and consistently, that its usefulness results in a process that terminates in linear versus quadratic time.

That is to say, for a large enough number of items, if we double the number of items, the time required for the task to complete will approximately double rather than quadruple.

As another example, consider shirts on a rack. We can find our size by starting at one end of the rack and moving to the other. Alternatively, we can start in the middle of the rack and then move left or right, thus halving the number of shirts we have to go through at each step.

Again, both methods are fast when we have a few shirts. The second method is much faster when we have lots of shirts. The first method may be described as growing at a linear rate and the second method at a logarithmic rate. The second method also has a name in computing—binary search.

This approach to teaching, oneself or others, is immensely powerful. It hones the learner's intuition by giving him or her the ability to spot patterns of problems and know how best to reason about them. The effect therefore manifests in a critical learner.

That to me is what thinking smarter is to a large extent about. It's the ability to not only receive knowledge, but to be able to immediately put it to practical use. And often, those capabilities come from looking at mundane things in novel ways.

“ This approach to teaching, oneself or others, is immensely powerful. It hones the learner's intuition by giving him or her the ability to spot patterns of problems and know how best to reason about them.

A Better Fit for Different Ways of Thinking

Another reason this way of thinking—let us call it algorithmic thinking—is useful, is because of its adaptability to different modes of learning. It is indeed the case that different people understand things in different ways. That is why prescription is such a tricky tool. It presupposes a lot about the audience and is ripe for being vulnerable to bias.

Over the years, I have come across many passages attesting to this fact. This is one from a student reminiscing about a professor. “I particularly enjoyed his lectures,” the author begins. “He had a mind like a freight train. It took so long to pull out of the station that you’d think he was stupid. But he would gradually speed up, becoming unstoppable once up to speed.”

Another passage, from the mathematician Laurent Schwartz, talking about himself as a schoolboy. He begins, “In spite of my success, I was always deeply uncertain about my own intellectual capacity; I thought I was unintelligent.” That’s a strange thing for a Fields Medal winner to say. He continues, “I was and still am rather slow.” And he goes on to talk about his memories from school and about his various anxieties.

Another passage, from Nikola Tesla’s autobiography, described how when he designed his machines, he would picture in his mind a DC machine. Then he would imagine an alternator, and visualize the motors and generators. “The images I saw were to me perfectly real and

tangible ... The motors I built there were exactly as I had imagined them.” And the machines behaved according to spec.

Each of those three individuals had a very different way of understanding. The first was slow and then gradually sped up. The second was slow throughout. The third was driven by seeing things in his imagination. Others are driven by seeing tangible results.

Another observation of algorithmic thinking is that when it’s placed in a context like everyday life, it lends itself to metaphors and analogies, both of which are approaches to learning that many find effective.

“The affinities of all the beings of the same class have sometimes been represented by a great tree,” Darwin writes when explaining common descent. “Rudimentary organs may be compared with the letters in a word, still retained in the spelling, but become useless in the pronunciation, but which serve as a clue in seeking for its derivation,” he writes when discussing inheritance.

His final passage contains yet another metaphor, “It is interesting to contemplate an entangled bank ... and to reflect that these elaborately constructed forms, so different from each other, and dependent on each other in so complex a manner, have all been produced by laws acting around us.”

Alan Turing talked of “burying” and “unburying” when describing a stack—one of the fundamental data structures in computing. Free software is defined on the GNU project’s page using a metaphor, “free as in speech, not as in beer.” Von Neumann reportedly modeled memory on how memories might be stored in the human brain, “an element which stimulates itself will hold a stimulus indefinitely.” Examples are aplenty.

It is precisely because people have different ways of understanding things that it’s useful to think about thinking in as broad and as timeless and as domain-agnostic a way as possible while still maintaining enough of the specifics to be valuable. It is in that spirit that I feel algorithmic thinking is as useful as other tools we have grown accustomed to having on hand, like arithmetic.

“It is precisely because people have different ways of understanding things that it’s useful to think about thinking in as broad and as timeless and as domain-agnostic a way as possible while still maintaining enough of the specifics to be valuable.”

An Essential Addition To Literacy

The near future is undoubtedly going to be one where machines are omnipresent. And one of the most important ways to prepare for a shift in the work force is to learn the lingua franca. Learning about the field in its most broadest sense (algorithmic thinking) down to its most specific sense (say, coding in a particular programming language for a particular purpose) is therefore not only useful, but necessary.

I recently toured seventeen elementary schools, both public and private, and found that in the overwhelming majority of them, some form of computer science teaching had been integrated into the curriculum. In a subset of those schools, active, project-based learning of the field was encouraged by way of policy and dedicated space.

That sort of reality suggests to one that elementary schools, no doubt pushed by the market and by non-profits and by eager parents, are realizing that in order for their boys and girls to be competitive in the work force twelve years down the line, they would benefit from picking up these new ways of thinking at an early age. The goal being not to reverse or impede the spread of machines and machine-assisted processes, something that is unlikely to be a tenable goal, but rather to develop skills in areas that would ultimately complement those processes. A machine that runs on a model of the world still needs someone to train it.

For that reason, this latest addition to K-12 education is unlikely to be a passing fad, one might argue, because of how the world looks. It's not a purely academic problem that schools are trying to solve. It's one that is driven by observable facts. By indisputable shifts in the world around us that demand anyone going into the workforce in a decade or less, irrespective of his or her domain, have working knowledge of fundamental computer science concepts.

I see this shift as a net positive and am optimistic about its effects on the upcoming generation.

“The near future is undoubtedly going to be one where machines are omnipresent. And one of the most important ways to prepare for a shift in the work force is to learn the lingua franca.”

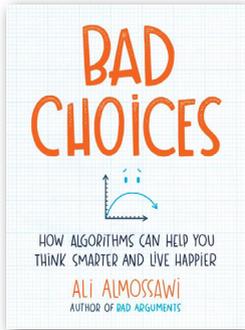
Thinking Smarter

Algorithmic thinking is of course not the only way to look at the world differently, but it is how someone who comes from a computer science background might see the world.

And as I claim in my book, *Bad Choices*, there's value in listening to that perspective. Just as there's value in understanding how a security engineer or a statistician or a journalist's view of the world and of everyday life is colored by their understanding of their respective fields.

The language varies in each case, the way of thinking is different, the assumptions aren't always the same. **I have been privileged to have spent the majority of my life working and living with people who are different in more than one respect, and I've found that exposing oneself to diversity of thought, in general, leads to smarter thinking.** 📖

Info



BUY THE BOOK | Get more details or buy a copy of [Bad Choices](#).

ABOUT THE AUTHOR | Ali Almosawi is the creator and maintainer of *An Illustrated Book of Bad Arguments*, which has been read by 2.5 million readers and translated into 18 languages, 12 of which were done by volunteers from across the world. His second book, *Bad Choices*, is an illustrated guide to algorithmic thinking. Ali currently works as a data visualizer at Apple and was formerly a data visualization engineer on the Firefox team at Mozilla, a research associate at Harvard, and a collaborator with the MIT Media Lab. He is an alumnus of MIT's Engineering Systems Division and Carnegie Mellon's School of Computer Science.

→ **SEND THIS** | [Pass along a copy](#) of this manifesto to others.

→ **SUBSCRIBE** | [Sign up for e-news](#) to learn when our latest manifestos are available.

This document was created on April 19, 2017 and is based on the best information available at that time.

The copyright of this work belongs to the author, who is solely responsible for the content. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs License. To view a copy of this license, visit [Creative Commons](#) or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Cover image from [Adobe Stock](#). You are given the unlimited right to print this manifesto and to distribute it electronically (via email, your website, or any other means). You can print out pages and put them in your favorite coffee shop's windows or your doctor's waiting room. You can transcribe the author's words onto the sidewalk, or you can hand out copies to everyone you meet. You may not alter this manifesto in any way, though, and you may not charge for it.

About ChangeThis

ChangeThis is a vehicle, not a publisher. We make it easy for big ideas to spread. While the authors we work with are responsible for their own work, they don't necessarily agree with everything available in ChangeThis format. But you knew that already.

800ceoread

ChangeThis is powered by the love and tender care of 800-CEO-READ. Visit us at 800ceoread.com, and keep up with the latest developments in business books on our review site, [In the Books](#).