The Book Review Column[1]

by Frederic Green

Departments of Mathematics and Computer Science
Clark University
Worcester, MA 01610
email: fgreen@clarku.edu

My predecessor as book review editor, Bill Gasarch, held the post for 17 years. In his final column in January 2015, he said that that's longer than anyone should have the job. I count myself as lucky to have been nominated as his successor. At the time, Bill advised me not to stay on for more than 5 years. Well, here it is 6.5 years out and some 26 columns later, so it is high time for me to retire from the position.

This has been a tremendously rewarding and educational experience, becoming acquainted with reviewers, authors, publishers, and reading and writing about books I probably wouldn't have otherwise. Actually I also learned a lot about reading *and* writing in the process! I hope you, the readers of this column, have benefited even $\frac{5}{12}$ as much as I have.

My deepest thanks go to the reviewers for their contributions, to publishers for their helpfulness and providing titles without hesitation, and to the authors for their comments and good will all around. Thanks also to the editors of SIGACT News and my fellow columnists for their support and collegiality.

You haven't heard the last of me, just as you didn't hear the last from Bill when he stepped down. His influence on the column, and his presence as a reviewer (and as book author; e.g., see this very column where the curious fraction above is explained), continues to this day. Likewise, I expect to carry over my efforts to the future. There are still several titles sitting on my shelf waiting to be reviewed.

I would not have considered stepping down without a successor in the wings. Nicholas Tran graciously offered to serve and will be writing the next column. The column will remain in good hands!

And so, as a parting gesture, and in the interest of a smooth transition, we present three reviews in this column by the editors past, present, and future:

1. **The Engines of Cognition: Essays by the Less Wrong Community,** by Less Wrong. A book about... well, you can hardly go less wrong than reading the review. Review by William Gasarch.
2. **Mathematical Muffin Morsels – Nobody wants a small piece,** by William Gasarch, Erik Metz, Jacob Prinz, and Daniel Smolyak. The combinatorics and algorithmics of muffin cutting and distribution. Review by Frederic Green.
3. **The Algorithm Design Manual, 3^{rd} ed.**, by Steven S. Skiena. A textbook on algorithms and algorithm analysis. Review by Nicholas Tran.

For now, please contact me (fgreen@clarku.edu) *or* Nicholas Tran (ntran@scu.edu) to write a review! Choose from among the books listed on the next page. Or choose one of your own. The latter

is preferable (and quicker) in the current circumstances, as I/we can then ask the publisher to forward it directly to you.

BOOKS THAT NEED REVIEWERS FOR THE SIGACT NEWS COLUMN

Algorithms

1. *Algorithms and Data Structures Foundations and Probabilistic Methods for Design and Analysis*, by Helmut Knebl
2. *Algorithms and Data Structures*, by Helmut Knebl
3. *Beyond the Worst-Case Analysis of Algorithms*, by Tim Roughgarden

Computability, Complexity, Logic

1. *Applied Logic for Computer Scientists: Computational Deduction and Formal Proofs*, by Mauricio Ayala-Rincón and Flávio L.C. de Moura.
2. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, by Martin Grohe.
3. *Semigroups in Complete Lattices*, by P. Eklund, J. Gutiérrez García, U. Höhle, and J. Kortelainen.

Miscellaneous Computer Science

1. *Elements of Causal Inference: Foundations and Learning Algorithms*, by Jonas Peters, Dominik Janzing, and Bernhard Schölkopf.
2. *Partially Observed Markov Decision Processes,* by Vikram Krishnamurthy
3. *Statistical Modeling and Machine Learning for Molecular Biology*, by Alan Moses
4. *Language, Cognition, and Computational Models,* Theirry Poibeau and Aline Villavicencio, eds.
5. *Computational Bayesian Statistics, An Introduction,* by M. Antónia Amaral Turkman, Carlos Daniel Paulino, and Peter Müller.
6. *Variational Bayesian Learning Theory,* by Shinichi Nakajima, Kazuho Watanabe, and Masashi Sugiyama.
7. *Knowledge Engineering: Building Cognitive Assistants for Evidence-based Reasoning*, by Gheorghe Tecuci, Dorin Marcu, Mihai Boicu, and David A. Schum.
8. *Quantum Computing: An Applied Approach*, by Jack D. Hidary

Discrete Mathematics and Computing

1. *Mathematics in Computing*, by Gerard O'Regan
2. *Understand Mathematics, Understand Computing – Discrete Mathematics That All Computing Students Should Know*, by Arnold L. Rosenberg and Denis Trystram

Cryptography and Security

1. *Computer Security and the Internet: Tools and Jewels*, by Paul C. van Oorschot

Combinatorics and Graph Theory

1. *The Zeroth Book of Graph Theory: An Annotated Translation of Les Réseaux (ou Graphes) – André Sainte-Laguë (1926)*, translated by Martin Charles Golumbic
2. *Finite Geometry and Combinatorial Applications*, by Simeon Ball
3. *Combinatorics, Words and Symbolic Dynamics,* Edited by Valérie Berthé and Michel Rigo

Programming etc.

1. *Formal Methods: An Appetizer*, by Flemming Nielson and Hanne Riis Nielson
2. *Sequential and Parallel Algorithms and Data Structures*, by P. Sanders, K. Mehlhorn, M. Dietzfelbinger, R. Dementiev

Miscellaneous Mathematics

1. *Algebra and Geometry with Python*, by Sergei Kurgalin and Sergei Borzunov.

Review of[2]
The Engines of Cognition:
Essays by the Less Wrong Community
Author: Less Wrong
Publisher: Less Wrong Press
https://www.lesswrong.com/books/2019
720 pages, Year: 2019
$30.00

Reviewer: William Gasarch (gasarch@umd.edu)

1 Introduction

For those who read my review of the first Lesswrong collection of essays, *A Map that Reflects the Territory*[3], this intro will give you a sense of what the Klingons call nlb'poH, the French call Déjá vu, and the English call Déjá vu.

Less Wrong is a forum founded by Artificial Intelligence Theorist Eliezer Yudkowsky in 2009. The stated philosophy is:

We are a community dedicated to improving our reasoning and decision-making. We seek to hold true beliefs and to be effective at accomplishing our goals. More generally, we work to develop and practice the art of human rationality.

That seems to cover a lot of ground! The actual topics seem to be (1) how does one find the truth in science and in life, (2) AGI (Artificial General Intelligence), and (3) probability. The most common non-trivial word in this book might be *Bayes*. Another common non-trivial word is *Goodhart*. (Goodhart's law is that when a measure becomes a target, it stops being a measure. It is often referred to when an AI system performs well but for the wrong reasons.) A trivial word would be something like *the* which is likely more common but less interesting. (Or is it trivial? The SIGACT News book review editor Fred Green pointed out that Ohio State has trademarked *the*. See
https://www.cnn.com/2022/06/23/us/ohio-state-university-trademarks-the/index.html

I do not know if that is more or less absurd than Donald Trump's failed attempt to trademark *you're fired*. See
https://www.cobizmag.com/who-owns-the-trademark-to-youre-fired/
to see who really owns the trademark to *you're fired*.)

The Engines of Cognition are actually a set of four books, titled *Trust*, *Incentives*, *Modularity*, and *Failure*. Each book is small—about 9 inches long and 5 inches wide. They can be read in any order. This set of book is a best-of-2019 collection as decided by the readers in some fashion.

2 General Comments

PROS: Many of the essays bring up a topic point that I had not thought of before, or have interesting thoughts about a topic I had thought of before.

[2] ©2022 William Gasarch

[3] For the review, see https://www.cs.umd.edu/~gasarch/bookrev/FRED/lesswrong.pdf, or on these pages, SIGACT News **53**(1), March 2022, pp 13–24.

CONS: Some of the essays go on and on about some point and either don't have much to say, or take too long saying it. This is most notable in the essays on AI where I want to yell at the author *try it out and see what happens rather than yakking about it*. When I posted a review of another Lesswrong collection, *A Map that Reflects the Territory* here
`https://www.lesswrong.com/posts/JXTEDFCC5r4dW2tta/review-of-a-map-that-reflects-th`
I had the same complaint. Some comments said that building AI systems is dang hard. Okay. Even so, stop yakking about it. It's getting boring.

CAVEAT (both a PRO and a CON): Some of the essays use words or phrases as though I am supposed to already know them. If I was a regular member of the forum then perhaps I *would* know them. In the modern electronic age I can try to look them up. This is a PRO in that I learn new words and phrases. For me this is a really big PRO since *I collect new words and phrases as a hobby*. This is a CON in that going to look things up disrupts the flow of the essays.

In the *third to last* section of this review I will have a list of all of the words and phrases I learned by reading these books and either their meaning or that I could not find their meaning. Why *third to last?* Because the second to last section is my summary opinion, and the reader of this review should be able to find it quickly (the last section is acknowledgments).

CAVEAT: As an extension of the last caveat, the essays tend to be written for other Lesswrongers. Now that I've read 9 Lesswrong books (5 from *Map*, 4 from *Engines*, and a few other occasional essays) I have may have become a Lesswronger; hence, this is no longer a problem for me. However, I sometimes read a paragraph and think "*A mundane*[4] *would not understand this.*"

In the spirit of the Lesswrong's quest for objective truth I will, in each section (and at the end), tabulate how many of the essays were Excellent (E), Good (G), and Meh (M) (none were bad). This will be an objective record of my subjective opinion.

3 Trust

I quote the first paragraph:

The first book is about trust, the belief in something in the absence of understanding.

There are 16 essays of which 7 are excellent, 6 are good, and 3 are meh. I will describe two that are excellent and linked, and one that is meh.

Excellent

Book Review: The Secret to our Success by Scott Alexander and **Reason isn't Magic by Ben Hoffman**

Lesswrong is devoted to reason. Yes indeed, reason is how humans succeeded and is a valuable tool today. Hence it was great to see an article in Lesswrong that uses reason to challenge the notion that reason is so great.

Meta time: I am reviewing Scott Alexander's review of a book. I wonder if when I post this on Lesswrong someone will review my review of Scott's review.

I will only discuss one aspect of the review; however, the review is fascinating and I assume the book is also.

The book that Scott reviews is *The Secret to our Success*. How did human beings survive? Did you ever try hunting and gathering—it's really hard! One common answer is that humans survived because they are smarter. The book Scott reviews challenges this notion. The book contends that the biggest advantage was cultural learning. Over time techniques that worked were learned *and passed down to the next generation*.

[4] slang term for people who are not Lesswrongers

We discuss one of their examples: Manioc. This is a plant that some peoples used as a staple. The time and effort they used to prepare it was very intense. Was this just a tradition (and hence perhaps a waste of time) or was it beneficial? The answers are Yes and Yes. Manioc has a lot of cyanide in it and the process they used got rid of the poison. Of course, they didn't know this. But that's not quite enough— how would they know the *long term affects* of eating the plant? They didn't; however, the process removed the bitter taste and got rid of some short term affects.

Someone could have tried to make the process less time consuming and still get rid of the bitter taste. This would have seemed reasonable but lead to cyanide deaths in the long term. But NO – nobody did this. So their *lack of reason*, their adhering to tradition for no good reason, was beneficial.

The second essay, *Reason isn't Magic* challenges this view. Hoffman points out that the time spent processing the food is also time lost— and perhaps some people starved since the process also made the supply less. This reminds me of the joke:

Vegans don't live longer, it just seems that way.

Meh

Chris Olaf's Views on AI Safety by Evan Hubinger

This is typical of the essays both in these books, in the last set of books I reviewed, on the Lesswrong blog, and other blogs that discuss AI Safety, AI alignment, and other AI issues. They seem to talk a lot but not really say anything. Or, more to the point, they have some ideas. Fine. TRY THEM OUT, then come back with what you found.

Of course there is a caveat: If I rate all of the AI article as *Meh* then does that mean there is something wrong with them (too long, not enough info) or with me (to impatient, not in the area of AI)?

4 Modularity

I quote the first paragraph:

The second book is about modularly. Well-designed or evolved structures are often not just made of parts, but made of parts with simple interfaces. These interfaces allow the parts to be reused in alternative contexts, and thus recombined in different ways.

There are 14 essays of which 6 are excellent, 2 are good, and 6 are meh. I will describe one from each category, which is not a fair sample.

Excellent

Gears-Level Models are Capital Investments by John S. Wentworth

When doing research should you strive to understand *why* things are the way they are (Gears-Level) or just *what* is happening?

An Example from Marketing:

Gears-Level: With massive data find correlations like "People who earn over $100,000 prefer to buy brand name chocolate," and use these to guide your ad campaigns.

Black-Box: Run lots of ad campaigns and see which one works.

The article discusses the pros and cons of these two approaches and gives lots of examples.

Good

Forum Participation as a Research Strategy by Wei Dai

If I read two articles and find a novel way to combine them, do an experiment to verify that my insight is correct, and publish the result, that's clearly research. If I write a blog on Lesswrong (or some other forum)

or write a comment on someone else's blog, is that research? Probably not, but it can *lead to* research. This essay discusses the PROS of participating on a forum and how it can contribute to research.

Giving this a *good* instead of an *excellent* might not be fair since I've had a blog on theoretical computer science, shared with Lance Fortnow since 2007, so there was nothing new in it for me. The blog is at `https://blog.computationalcomplexity.org/`

Meh

The Credit Assignment Problem by Abram Demski

This essay begins with examples of how to assign credit to success or failure and asserts correctly that this is an important problem. They then have some good ideas about the problem. But then the article goes off topic and is too long.

5 Incentives

I quote the first paragraph:

The third book is about incentives, which are patterns of what is rewarded and what is punished.

There are 16 essays of which 10 are excellent (wow!), 2 are good, and 4 are meh. I will review 3 of the excellent essays (2 of which are tied together) and none of the others. This is asymmetric, which is the topic of the first essay I describe. More to the point, it seems like all of my critiques of the Meh essays are the same: talk too much, don't say much. Hence my critiques of them also talk too much and don't say much, so I omitted them.

Excellent

Asymmetric Justice by Zvi Mowshowitz and **The Copenhagen Interpretation of Ethics by Jai Dhyani**

The Copenhagen Interpretation of Ethics is that when you observe or interact with a problem you can be blamed for it. Or perhaps you will be blamed for not doing enough. I give two examples, one from the second essay, and one from neither essay.

1. (This is from the second essay.) At one time Detroit was having a hard time with high water bills. People for the Ethical Treatment of Animals (PETA) told families that they would pay their water bills for a month, if the family went vegan for that month. This article

   `https://www.nbcnews.com/news/us-news/peta-detroit-go-vegan-month-well-pay-your`

   says that PETA was criticized for this. One quote:

   Water is a human right. Period. Holding it out like a prize proves PETA doesn't value human life.

2. (This is not from either essay.) During the Flint Michigan Water crisis Ted Cruz donated water to crisis pregnancy centers, which are really places women go to thinking they will get help, but instead they are lectured about why they should NOT get an abortion. He was criticized for this. And I also thought badly of him (more than usual).

BUT WAIT A MINUTE! Did the guy who blasted PETA give any water or money to Detroit? Did I do anything for Flint? It is unfair to criticize them for doing *something* as opposed to doing *nothing*.

The two essays are about issues of justice. One is the issue above, that there may be a disincentive to help. Another issue is asymmetry: bad actions are punished but good actions are not rewarded. How to fix this? The two essays give you a lot to think about.

Moloch Hasn't Won by Zvi Mowshowitz

This essay is one in a sequence of essays that go back before Lesswrong was a forum. In *Hierarchy for Philosophers*[5], C.S. Lewis writes:

Who does it? Earth could be fair, all men glad and wise. Instead we have prisons, smokestacks, asylums. What sphinx of cement and aluminum breaks open their skulls and eats up their imagination?

Alan Ginsberg answers the question: *Moloch does it* He gave a much longer answer, where he is howling at Moloch, but that's the drift. And even with that, I had a hard time figuring out what he meant.

Scott Alexander's Slate article *Meditations on Moloch*, which you can find here
`https://slatestarcodex.com/2014/07/30/meditations-on-moloch/`,
takes the question of why humankind is in such bad shape seriously. He is particularly interested in why, if nobody likes the current system, it persists

Scott's essay has a list of 14 real world phenomena which any rational person would want to change and yet nothing changes. They are mostly Prisoner's Dilemma, Tragedy of the Commons, Malthusian scenarios, but they are not abstract. They are real. He then proposes some ways out of these traps.

Zvi's follow-up essay says what Scott got right and what Scott got wrong. Hmmm, that sounds too shallow. Zvi's essay is an intelligent comment on Scott's essay. Read them both.

6 Failure

I quote the first paragraph:

The fourth book is about failure. It's what happens when a system behaves differently from how we expect it to, with adverse consequences for those who were relying on the success of that system. Failure is often as much about misunderstanding how a system works, as it is about the lack of effort or plan to bring the system into a successful configuration.

There were 13 essays of which 5 are excellent, 2 are good, and 6 are meh. I will discuss 2 excellent, 1 very good, and 1 meh.

Excellent

Blackmail by Zvi Mowshowitz

This essay discusses why blackmail should be illegal. You might think *of course it should be*. This essay gives good arguments for why it is illegal but also raises questions about the entire endeavor.

Why wasn't science invented in China? by Ruben Bloom

The title is not quite right: some science was done in China at about the same time as in Europe. But far less. This essay gives cogent reasons for this. I quote one here: *Unlike Europe, China's political, religious, legal, and educational systems did not afford the neutral spaces where novel ideas could be advanced and old ideas questioned.*

Good

AI Success Stories Wei Dai

This article discusses various AI success stories and gives criteria to tell if they really were successes. This is interesting; however, it was only 5 pages – I would have wanted more examples.

Meh

[5] I have not been able to find the book or article *Hierarchy for Philosophers by C.S. Lewis*. The only references to it are on the Lesswrong Forum. Conspiracy?

The Strategy Stealing Assumption by Paul Christiano

The strategy stealing assumption is that for any strategy an unaligned AI can use to influence the long-run future, there is an analogous strategy that a similarly-sized group of humans can use in order to capture a similar amount of flexible influence over the future. The article is speculative about this. I would prefer it to give concrete examples.

7 Newords that I Learned From These Books

The word *Newords* is not a misspelling. The best neologisms do not need to be explained. Oh well.

7.1 From the Book Trust

1. **Kaggle Competition**: From the website of Kaggle,

   `https://www.kaggle.com/docs/competitions`

   Kaggle Competitions are designed to provide challenges for competitors at all different stages of their machine learning careers. As a result, they are very diverse, with a range of broad types.

2. **The model contains no gears**: Machine learning models often work great but nobody knows why. But it's worse than that. There is no why, it's "just" pattern matching.

3. **Chesterson's Fence**: The theologian G.K. Chesterson said that if you see a fence that you want to knock down, *don't!* You must first understand why it is there. More generally, any proposed reform to a system must understand why the system is there in the first place. Economist-philosopher Edmond Burke had similar ideas and is considered one of the founders of a certain school of conservative thought. While this is often a wise policy, it can also be an excuse for doing nothing.

4. **The Toxoplasma of Rage**: Memes that are controversial and incite rage, even negative rage, are more effective at getting the message out. Scott Alexander has a great article about this that probably coined the term (for this context–it also has a medical meaning) here:

   `https://slatestarcodex.com/2014/12/17/the-toxoplasma-of-rage/`

5. **Zombie Theories**: A theory appears in a paper that is likely false. However, nobody bothers debunking it, so it keeps getting revived. This can even happen to theories that are debunked, like that vaccines cause autism.

6. **FOOM Debate**: Robin Hanson and Eliezer Yudlowsky had a debate about the future of AI, called *AI-FOOM Debate*. You can read about it, and download it, here

   `https://intelligence.org/ai-foom-debate/`

   Why is it called FOOM? Because **FOOM** is a sudden increase in artificial intelligence such that an AI systems becomes extremely powerful. This may be the same or close to **The Singularity**.

7.2 From the Book Modularity

1. **Gear-level Models** vs **Black-Box Models**: A Gears-Level Model strives to understand what's really going on. A Black-Box Model is only concerned with input-output.

2. **metis and Metis**: The book says that metis is knowledge handed down for generations that might not make sense. Wikipedia says that the Metis are a group of indigenous peoples who inhabit parts of Canada. It is likely that the the Metis have metis. (lalaithion informs me that *metis* is an ancient Greek word which originally meant *magical cunning* but drifted to mean *wisdom* or *prudence* or *the je ne sais quoi of being able to solve practical problems*. He also points to James C Scott's book *Seeing Like a State* where it was used to mean implicit knowledge passed down through a culture.)

3. **Disputant**: The article *Coherent Decisions Imply Consistent Utilities*, in the section *Why not circular preferences?* begins as follows

   > *De gustibus non est disputandum* goes the proverb, matters of taste cannot be disputed.

   Okay, that's fine. But later in that section it says,

   > That (circular preferences) sound wrong. But can we disputandum that.

   Clearly the author just meant dispute and is either attempting to be funny (I don't think it is, but of course, *De gustibus non est disputandum*) or made a mistake. Even so, I am happy to know the quote.

4. **Utilon**: A unit of pleasure. You need to define how much it is for yourself as there is no standard. From *Map* I learned the word **Hedon** which is a unit of pleasure. I got 2 utilons and 3 hedons when I read gjm' comment reminding me that I had **Hedon** in my word list from my review of *Map*.

5. **The Allais Paradox**: Wikipedia does a good job on this one, so see

   `https://en.wikipedia.org/wiki/Allais_paradox`

6. **MIRI**: Machine Intelligent Research Institute. I quote their website:

   A non-profit research organization devoted to reduce the existential risk from unfriendly AI, and understanding problems related to friendly AI.

7.3 From the Book Incentives

1. **The Copenhagen Interpretation of Ethics**: See my description of the Incentives book for the definition.

2. **Trained in the Way**: I assumed this meant thinking rationally and objectively and all the good things that Lesswrong values. They reference Eliezar Yudkowsky's post *Twelve Virtues of Rationality*. I read that and he never quite defines the term but it seems to mean what I thought.

3. **Schelling Point**: Two people (or companies) want to communicate but for some reason can't. Even so, if they have the same social-cultural background they may be able to, without communication, get an agreement. The agreement is called the Schelling point. This was first introduced by economist Thomas Schelling in his book *The Strategy of Conflict (1960)*. I give three examples.

   (a) Two Americans know they need to meet but can't communicate where and when. They might both end up under the clock in Grand Central Station on New Years Eve at 12:01PM.

(b) Alice is writing a review of Bob's book, and Bob is writing a review of Alice's book. They *want* to say *I'll give you a good review if you give me a good review* but that would be unethical. Even so, they end up doing just that.

(c) Two companies sell the same product and the price fluctuates between 8 and 12 dollars. Eventually both will sell it for 10 dollars.

4. **Simulacrum**: A representation of something. Often thought to be inferior. In the book it's used for things like meaningless titles (e.g., Vice President in charge of Sorting). See next phrase. (The plural is *Simulacra*.)

5. **Baudrillard's Theory**: Baudrillard was a French philosopher who thought that society had become so saturated with simulacra and lives so saturated with the constructs of society that all meaning had become meaningless by being infinitely mutable. This theory applied to the Trump presidency explains the epidemic of akrasia in high office as well as our citizenry.

7.4 From the Book Failure

1. **Akrasia**: The article uses the term without saying what it means. A search on the Lesswrong website yielded other articles that use the term without saying what it means. I think they are using the following which I got from Wikipedia:

   Akrasia: A lack of self-control or the state of acting against one's better judgment.

2. **CFAR**: Center for Applied Rationality. Here is their website:

   `https://rationality.org/`

3. **Connectionism**: The belief that we can explain intellectual ability using artificial neural networks.

4. **The Curse of the Counterfactual**: When you compare reality to what could have been (or what your rose-colored hindsight glasses see) you may get depressed. I've also heard this called **buyer's remorse**.

5. **Internal Family Systems (IFS) Therapy**: A therapy used to cure traumas. See the website

   `https://ifs-institute.com/`

6. **Litany of Gendlin**: This is a method to combat the curse of the counterfactual. This is a quote from Eugene Gendlin:

   > "What is true is already so.
   > Owning up to it doesn't make it worse
   > Not being open about it doesn't make it go away
   > And because it's true, it is what is there to be interacted with
   > Anything untrue isn't there to be lived
   > People can stand what is true
   > for they are already enduring it."

   (I got this from the Less Wrong Website. I don't think it's used beyond that.)

7. **Litany of Tarski**: This is a method to combat the curse of the counterfactual. A template to remind oneself that beliefs stem from reality, from what actually is as opposed to what we want or what would be convenient. Logically

   If X then I desire to believe X.

   (I got this from the Less Wrong Website. I don't think it's used beyond that.)

8. **The Unilateralist's Curse**: We give an example from the article *The Unilaterist's Curse and the Case for a Principle of Conformity* by Bostrom, Douglas & Sandberg.

   A group of scientists working on HIV accidentally create an air-transmissible variant. 19 out of the 20 scientists agree that this should not be published. But one disagrees since he thinks the world should know about the danger and prepare. He announces the result at a conference.

   The Unilaterist's Curse is that a small number of people can act against what the vast majority wants.

9. **The Great Divergence**: This refers to when Europe began to dominate other countries culturally, economically, and scientifically. This occurred in the 19th century.

8 Should You Read This Book?

Yes.

Okay, I will elaborate on that.

8.1 Should You Read This Book? The Numbers

I review my ratings E for Excellent, G for Good, or M for Meh (none were B for Bad):

1. *Trust* E-7, G-6, M-3.
2. *Modularity* E-6, G-2, M-1.
3. *Incentives* E-10, G-2, M-4.
4. *Failure* E-5, G-2, M-6.

What to do with this information?

1. There are 28 excellent articles! That's. . . excellent!
2. There are 12 good articles! That's. . . good?
3. There are 14 meh articles! That's. . . meh.

The ratio 28-12-14 is excellent and is better than that for *A Map that Reflects the Territory* which got 15-15-15. Since I don't really think *Map* was worse than *Engines* I may have been harsh on *Map*. Or perhaps I've drank the Kool Aid. In any case, yes, you should buy this book.

8.2 Should You Read This Book? Not the Numbers

Let's look at the extremes: the best and worst thing about the book.

Best things I got out of the book:

1. Many of the essays discuss how the world got the way it is, what's wrong with it, and possible solutions. While these are mostly in the books *Trust* and *Incentives* there are other articles that touch on these points. The discussions are refreshingly honest, objective, and do not have a bias ahead of time.

2. Some of the essays discuss gears-level vs black box. This is one of those concepts which you sort-of know ahead of time but is great to see written down and explained and explored at much greater length. While these were mostly in the book *Modularity* other articles touch on it.

3. This will sound like a back-handed compliment. Or a left-handed complimented. The book gives pointers to OTHER really good books and essays.

Worst thing about the book:

1. As noted earlier, the AI essays often had too much talk-talk-talk and not enough walk-walk-walk. I am reminded of how early (in the 1960s) people would talk about how great AI was going to be and then babble some incoherent philosophy about machines thinking. The current discussion is *not* how great it will be. It's about (a) is AI dangerous? and (b) if so how to tame it? Those are good questions, but the essays about it were talking in a vacuum. This reminds me of an old joke and a new joke. Both begin the same.

   How does someone in AI make love to their spouse?

   - Old Joke: They sit on the bed and tell them how great its going to be.
   - New Joke: They sit on the bed and talk about making sure to align what you really want with what you say you want, and be careful since if you get things wrong that could be dangerous.

Fortunately, as the numbers tell you, the meh essays were fairly few; however, as a book reviewer I had to read them. You can use a variant of Ebert's rule:

If you don't laugh in the first 15 minutes of a comedy, you won't laugh in the remaining 105 minutes.

I also hasten to point out, these are just my opinions, and as well known, *de gustibus non est disputandum*.

8.3 Should You Read This Book? The Elephant in the Room

(The next paragraph is almost word-for-word what I wrote in the review of *A Map that Reflects the Territory*.)

And now for the elephant in the room: Why buy a book if the essays are on the web for free? I have addressed this issue in the past:

1. I've reviewed 4 blog books. See the next four links for the reviews:

   `https://www.cs.umd.edu/~gasarch/BLOGPAPERS/lipton.pdf`

   `https://www.cs.umd.edu/~gasarch/BLOGPAPERS/liptonregan.pdf`

   `https://www.cs.umd.edu/~gasarch/BLOGPAPERS/tao.pdf`

   `https://www.cs.umd.edu/~gasarch/bookrev/FRED/Lesswrong.pdf`

2. I have written my own blog book: *Problems with a point: Explorations in Math and Computer Science by Gasarch and Kruskal*

   See here for its entry on amazon:

   `https://www.amazon.com/Problems-Point-Exploring-Computer-Science/dp/9813279974`

Here is an abbreviated quote from my book that applies to the book under review.

The Elephant in the Room

So why should you buy this book if it's available for free?

1. Trying to find which entries are worth reading would be hard. There are a lot of entries and it really is a mixed bag.

2. There is something about a book that makes you want to read it. Having words on a screen just doesn't do it. I used to think this was my inner-Luddite talking, but younger people agree, especially about math-on-the-screen.

9 Acknowledgments

I thank the entire Lesswrong community for existing and putting forth reason as a way of life (though be careful eating Manioc).

I thank Oliver Habryka for giving me this opportunity to review these books.

I thank Nathan Hayes and Ben Pace for proofreading and useful comments.

I thank Ruby Bloom and Justis-from-Lesswrong2.0 for helping me post this review to the LessWrong forum in text form (having a pointer to the review I learned was not as good).

I thank Fred Green, editor of SIGACT News for proofreading and not asking me to cut anything for length.

I thank lalaithon, gjm, and Richard Kennaway for comments they made when I posted this on lesswrong, which I incorporated into the review.

All of the people acknowledged help make this review LessWrong.

Review of[6]
Mathematical Muffin Morsels – Nobody wants a small piece
by William Gasarch, Erik Metz, Jacob Prinz, and Daniel Smolyak
World Scientific, 2021
210 pages, Softcover, $59.99.

Review by
Frederic Green (fgreen@clarku.edu)
Departments of Mathematics and Computer Science
Clark University, Worcester, MA

1 Introduction

You have 5 muffins and 3 (as usual) hungry students. They ask that you divide the muffins equitably between them. Do you cut every muffin into 3 parts, and give 5 separate pieces of size $\frac{1}{3}$ to each student? Well no, because that's too easy, and more importantly no student wants any piece that's less than a third of a muffin. Can *this* be done? Indeed it can! It turns out you can guarantee that everyone gets a piece of size at least $\frac{5}{12}$. Can everyone be guaranteed a bigger piece than that? Alas, the answer is no. That's the bad news. The good news is that one could write a book about it, and these guys wrote a great one.

The problem can be generalized in natural ways. For example, you could have 5 donuts and 3 secretaries, 5 cookies and 3 toddlers, 5 treats and 3 dogs, 5 grants and 3 professors, or 5 tasks and 3 processors. But wait a minute: You could have m muffins (etc.) and s students (etc., resp.)! Maybe that's a more interesting generalization...

Let m denote the number of muffins and s the number of students. Is there a procedure to find the largest minimum size piece that each student gets? And, if there is, what is that largest size, which is denoted $f(m, s)$? E.g., $f(5, 3) = \frac{5}{12}$ as described above. The book is dedicated to this problem, which is far more subtle and interesting than it may at first appear.

2 Contents

- Chapter 1, "Five Muffins, Three Students; Three Muffins, Five Students": How to solve the specific case mentioned above, and also what happens when you switch the numbers.

- Chapter 2, "One Student! Two Students! Some Basic Theorems!": A short chapter that gives more examples and sets down the basic notation (e.g., $f(m, s)$). From here on in the goal is to systematically generalize the ad hoc method given in Chapter 1. A few simply proved but useful tools appear. One of them gives criteria for when we can guarantee that every muffin is cut into two pieces. Another is an elegant duality between the m-muffin, s-student problem and the s-muffin, m-student problem, thus explaining the result at the end of Chapter 1.

- Chapter 3, "Our Plan," lays out the agenda of the rest of the book in a couple of pages. In brief, it lists the various methods to be considered. These methods can be feasibly applied to relatively prime m, s in the range $3 \leq s \leq 100$ and $s \leq m \leq 110$, of which there are 3520 pairs. As the book progresses,

[6]

for more values of (m, s), upper bounds on $f(m, s)$ can be determined that were not found before. In the introduction to later chapters, cases are given for which the earlier methods fail. Still, each method builds on the previous ones. The one exception is in Chapter 12, which see below.

- Chapter 4, "Three Students! Four Students! The Floor-Ceiling Theorem!": A useful formula for an upper bound for $f(m, s)$ is given, assuming that s does not divide m (the nontrivial case of interest). The name of the theorem derives from the fact that the upper bound is expressed in terms of floors and ceilings of $\frac{2m}{s}$ (a concept that is not assumed to be known to the reader). That theorem is then used to determine $f(m, s)$ for (you guessed it) $s = 3$ and $s = 4$, and all m.

- Chapter 5, "Finding Procedures" is a more descriptive title than it at first appears, since these are actually procedures for finding procedures. Chapter 4 gave an upper bound for $f(m, s)$ via the Floor-Ceiling Theorem. Note that in this context, the usual role of upper and lower bounds are reversed: Here an upper bound tells us what we *cannot* do (i.e., not everyone is guaranteed a large enough piece), while lower bounds show what we *can* (ensure that everyone gets a large piece). This chapter gives a procedure ("FINDPROC") that tries to find procedures to prove lower bounds: $f(m, s) \geq \alpha$ for a given rational α. The problem reduces to linear algebra, in which one must find integer solutions. For this and other reasons, the chapter is labeled as requiring more mathematical background than others. The linear equation themselves need to be found, and this can be done either by linear algebra, or memoization (denoted here by "recursion"... indeed, in quotes). That in turn is motivated via quick introductions to recursion (not in quotes) and dynamic programming. FINDPROC "tries" to find a procedure in the sense that it can output "Don't Know" ("DK"). But assuming a conjecture, given in this chapter, it will never output DK if a procedure actually exists. This method is used for lower bounds in the subsequent chapters.

- Chapter 6, "The Half Method": I find it intriguing how a proof by contradiction can lead to an algorithm. The proof proceeds by assuming that an upper bound on $f(m, s)$, say $f(m, s) \leq \alpha$, doesn't hold. Thus assume $f(m, s) > \alpha$. Then (to consider one scenario), we can use previous results (Theorem 2.6) to put an upper bound on the number of pieces of size $> \frac{1}{2}$, and a contradicting lower bound on the number of pieces $> \frac{1}{2}$ in any procedure where $f(m, s) > \alpha$. The algorithm emerges by *deriving* an α such that this proof by contradiction works.

- Chapter 7, "A formula for $f(m, 5)$": After having taken care of 3 and 4 in Chapter 4, we now graduate to 5. The chapter consists almost entirely of an exercise and its solution, in which the formula (which is a very complicated reduction into cases) is derived via exhaustive applications of the Floor-Ceiling Theorem, for *all* m (not just those in the range specified in Chapter 3). A brief discussion at the end mentions generalizations (e.g., 6, 7, etc.), that leads to formulas too complicated for the book, but are on the associated website.

- Chapter 8, "The Interval Method": Here we consider the number of shares each student can have, and place them into intervals, each bracketing the size of the shares. An analysis of these intervals leads to an upper bound on $f(m, s)$. This kind of reasoning grew of the methods of Chapter 6, but succeeds in some cases where the Half Method failed. By the end of the chapter, 80% of the 3520 cases set down in Chapter 3 have been solved, either by the Floor-Ceiling Theorem, the Half Method, or the Interval Method.

- Chapter 9, "The Midpoint Method": This is similar to the Interval Method, but with a special twist in which $\frac{1}{2}$ is in the exact middle of an interval. In this and the preceding chapter, exercises ask the reader

to write the method him- or herself, given very helpful hints. By now, combined with the territory gained in the previous chapter, just over 83% of the Chapter 3 cases have been covered.

- Chapter 10, "The Easy Buddy-Match Method": The "buddy" of a piece of muffin of size x is the remaining portion of size $1 - x$. In the case that some students receive 2 shares and possibly some 3, this is relaxed to the idea of a "buddy match": The buddy match of a piece of size x is the other piece a student with 2 shares would receive, which is of size $\frac{m}{s} - x$. A procedure for finding buddy matches leads to an algorithm for finding yet another upper bound for $f(m, s)$, covering cases that were not covered before.

- Chapter 11, "The Hard Buddy-Match Method": Continues the idea of the previous chapter. This method is called "hard" because it involves linear algebra and an analysis of the types of students (how many have 2 shares, again a requirement, and how many 3). By the end of this chapter, over 92% of the Chapter 3 cases have been covered. On the home stretch!

- Chapter 12, "The Gap and Train Methods": The Gap Method starts out like the Midpoint Method, but changes course when it discovers (by finding that a set of linear equations *does* have a solution) that it will fail, outputting "DK." The Gap Method succeeds by identifying gaps between intervals like those studied in Chapter 8 and later. The Gap Method leave all of 13 out of the 3520 target cases unsolved. These are taken care of by the Train Method, which is omitted from the book but can be found on the muffin site. All the methods can be combined into one algorithm which applies each one in turn and finds the minimum.

- Chapter 13, "Scott Huddleston's Method": The previous chapter reached 100% of 3520. What about the remaining infinite number of cases? For one, the authors note the distinct possibility that the techniques of the first 12 chapters can be adapted to handle all muffin problems, but this has not yet been done, and there are cases for which those methods are known not to work. However, these problems can be solved by the method in this chapter, which is separate in technique and spirit from the earlier ones, much more sophisticated mathematically, and more general. Scott Huddleston first discovered the algorithm, which was independently found by Richard Chatwin, who also proved that it is correct (the proof is beyond the scope of the book). Scott's algorithm finds $f(m, s)$ in $O(m^2)$ time. It generalizes the original problem by associating a value with each muffin and reducing the problem to a graph problem. The vertices of the (multi-)graph are identified with muffins or students, the degree of a "muffin-vertex" being the number of pieces into which the muffin is divided, and the degree of a "student-vertex" the number of pieces that student gets. The algorithm is first illustrated with two non-trivial examples and then described in general.

3 Opinion

One word you'll find embedded in the word "muffin" is "fun." And this book has a great deal of fun in it (to say nothing of humor), as it is obvious the authors also had in doing the work. Another title might have been "Muffin Fun," but maybe that has a slightly off-color ring to it. The authors certainly "go wild" with the topic, and to be honest, another title that kept popping into my mind replaced "Morsels" with "Madness."

But though this be madness, yet there is method in it. There is a very deliberate and clear structure to the first 12 chapters, in which an increasingly large swath of the feasible mathematical muffin landscape is systematically and ruthlessly tamed. Each technique builds on and adds insight to earlier ones, a clear illustration of how algorithms and mathematical proofs can be constructed.

Furthermore, all algorithms and proofs are motivated by examples. Lots of them. This is one major feature that makes it so comprehensible, without being dry or boring. It is also loaded with exercises, making it one of those books where the reader may feel like a participant in the writing; sound pedagogy. Thus a high school or college student who is relatively new to math or CS will find numerous important concepts effectively and concretely illustrated here: Proofs of various kinds (notably contradiction, but also induction), algorithm design (iteration, recursion, dynamic programming), combinatorics, significant applications of linear algebra, and even a smattering of complexity. That the motivation is driven by examples does not mean that rigor is sacrificed. After all, there *are* real definitions and proofs, given informally but still carefully.

The authors point out that the book serves multiple purposes, and I agree. One would be to introduce math to motivated students. Another would be to motivate them to begin with. I would add that yet another would be to motivate seasoned researchers to get interested in this type of problem, and in that I think it will succeed admirably.

The richness of examples serves another purpose: You can amaze your friends. If you ever find yourself with a few muffins (or donuts, etc.) to divide amongst a few students (or secretaries, etc.), this is the book to turn to. It's not unlikely you'll find the exact case you're dealing with.

Review[7] of
The Algorithm Design Manual, 3^{rd} ed.[8]
by Steven S. Skiena
Springer, 2020
793 pages, Hardcover, $99.99

Review by

Nicholas Tran (ntran@scu.edu)
Department of Mathematics & Computer Science
Santa Clara University

1 Overview

The Algorithm Design Manual is an aptly named undergraduate-level textbook that takes a utilitarian approach to presenting algorithms as a collection of off-the-shelf technologies that can be adapted to obtain practical solutions to real-world problems. Its intended audience is aspiring and practicing professional programmers, and its pedagogical emphasis is on facility in application rather than rigorous implementation or analysis of these technologies.

The book consists of two parts. Part I (Practical Algorithm Design) covers standard topics in basic data structures as well as in algorithm design and analysis techniques. The highly engaging and intuitive presentation of these topics is leavened with actual "war stories" that together bring to the fore real-world issues in algorithm design often not addressed in traditional textbooks, e.g., problem modeling, output validation, evaluating efficiency of different data structures for the same problem, and dealing with large data sets. Each chapter ends with a short bibliography and a substantial set of exercises, implementation challenges, interview questions, and relevant competitive programming problems on LeetCode, HackerRank, and Online Judge.

Part II (The Hitchhiker's Guide to Algorithms) is a catalog of seventy-five fundamental algorithmic problems (both tractable and intractable) à la Garey-Johnson. Each problem entry begins with its input/output description, followed by a list of practical applications that give rise to the problem and its variations; a discussion of its computational complexity and known efficient algorithms or heuristics; existing software implementations of these solutions in software libraries and code repositories; a brief bibliography; and a list of related problems.

Major changes in the third edition include expanded coverage of randomized algorithms, divide and conquer, approximation algorithms and quantum computing in Part I, updated reference material in Part II, and color illustrations throughout. Online lecture notes, video, and problem solutions have also been updated on the author's website.

2 Chapter Highlights

Chapter 1 introduces algorithmic problems and their solutions and goes to great lengths to emphasize the importance of demonstrating correctness of algorithms with examples of different proof techniques. This chapter also reviews combinatorial and recursive objects commonly used to model problems and presents

[8] Second edition reviewed on these pages, SIGACT News **42**(4), December 2011, pp. 29-31.

a war story that stresses the importance of model validation, i.e., making sure that the model accurately reflects the problem to be solved.

Chapter 2 defines formally best-case, worst-case, and average-case running times on the RAM model, common function families, and summation formulas. A remarkably effective table comparing the growth rates of $\log n$, n, n^2, 2^n, and $n!$ is used to motivate asymptotic notation. Staying true to the "manual" spirit, this chapter provides for easy lookup a list of prototypical algorithms and their running times. It ends with a brief discussion of using the formal limit definition to compare growth rates of more esoteric functions and a list of other texts for more in-depth treatment of algorithm analysis.

Chapter 3 reviews basic data structures such as arrays, linked lists, stacks, queues, priority queues, dictionaries and hashing, and the complexity of their operations. Complete C code for linked lists and binary search trees (except `delete`) is provided. A war story shows how an application is sped up by identifying the operation performed most often by the underlying algorithm and selecting the fastest data structure for that operation.

Chapter 4 motivates the need for sorting by showing that many problems can be solved easily if the input data is ordered. Heap, merge, and (randomized) quicksort algorithms are then presented and analyzed informally, followed by an argument that $\Omega(\log n!)$ comparisons are needed by any comparison-based sorting algorithm. A war story highlights the need to consider external memory performance when sorting very large data sets.

Chapter 5 discusses divide-conquer design strategy in general, including the master theorem, but focuses mainly on binary search and its variants. Other classic divide-conquer algorithms are briefly but lucidly sketched such as fast integer and polynomial multiplication, closest pair, as well as convolution and applications. A war story notes that divide-conquer algorithms could be parallelized, but care must be taken to ensure that the load is balanced among the processors.

Chapter 6 gives a high-level overview of randomized algorithms. Although a review of probability theory, balls and bins, the coupon collector's problem is included, linearity of expected values (a main tool for randomized algorithm analysis) is not. Various randomized hashing and related methods are discussed (simplified universal hashing, Bloom filters, minwise hashing), as well as Fermat primality test and the Rabin-Karp string matching algorithm. The chapter ends with an inspiring war story about the author's successful search for a counter-example to a conjecture about binomial coefficients using a short Mathematica program.

Chapter 7 defines various flavors of graphs, proposes adjacency matrix and adjacency lists as two basic choices for representing graphs, and implements breadth-first search and depth-first search. Applications of BFS (shortest paths, connected components, bipartiteness) and DFS (cycle detection, articulation vertices, topological sorting, strongly connected components) are also given in C code. I find the explanations given in this chapter less intuitive; adding mathematical properties about BFS and DFS (e.g., pre-order and post-order visit times) would enhance understanding of these traversal methods and their applications.

Chapter 8 implements various classic algorithms on weighted graphs: Prim's and Kruskal's algorithm for minimum spanning trees, Dijkstra's and Floyd-Warshall's algorithm for shortest paths, Edmonds-Karp's algorithm for maximum flow, and Karger's randomized algorithm for minimum cut. A war story shows how reconstructing English words as typed in a telephone keypad (a less familiar technology to smartphone users today) can be modeled as a shortest path problem.

Chapter 9 covers in-depth how to perform exhaustive search enhanced with pruning techniques to achieve reasonable running times. A general template in C for performing backtracking is given and adapted to produce all subsets and permutations of n objects and all paths in a given graph. Pruning techniques such as branch-and-bound and A* are applied to achieve good results in solving Sudoku and a problem in covering

a chessboard is detailed in a war story.

Chapter 10 is a comprehensive treatment of dynamic programming, starting with computing Fibonacci numbers and binomial coefficients, edit distances and variants, subset sums, ordered partitions, and the CYK algorithm for parsing context-free languages. Three war stories show how dynamic programming is used in bar codes, power grid load balancing, and unification in logic programming. In the chapter introduction, it is noted that dynamic programming works well for optimization problems on combinatorial objects that have an inherent left-to-right order among components. I believe another way to state this is that dynamic programming works well for problems that can be modeled using a DAG, which would amplify the message of "design graphs, not algorithms" stated in Chapter 8.

Chapter 11 is an informal (and probably fuzzy to novices) treatment of NP-completeness. It defines reductions between decision problems (in terms of the unhelpfully named problems of *Bandersnatch* and *Bo-billy*), introduces 3-SAT, integer partition, vertex cover, and Hamiltonian Path as acknowledged hard problems, and shows how to establish NP-hardness via reduction using a couple of examples. Semi-formal definitions of P, NP, NP-hardness, and NP-completeness appear at the end of the chapter. Two war stories on when things go very well or very wrong in a lecture are likely to resonate with instructors everywhere.

Chapter 12 gives examples of approximation algorithms with constant and logarithmic factors. It also gives a readable description of heuristic methods such as random sampling, hill climbing, and simulated annealing. The chapter ends with an idealized description of quantum computers, which is used to explain Grover's and Shor's algorithms.

Chapter 13 contains a short checklist for things to try when designing an algorithm and some advice on preparation for tech company interviews.

Chapters 14–21 provide detailed notes for 75 fundamental problems that arise often in practice. They are grouped under data structures, numerical problems, combinatorial problems, tractable graph problems, intractable graph problems, computational geometry, and set and string problems. Entries that do not usually appear in an algorithms textbook include arbitrary-precision arithmetic, calendrical calculations, and "drawing graphs nicely." A spot check on the graph isomorphism entry shows a reference to Babai's 2016 quasi-polynomial time algorithm.

Chapter 22 enumerates algorithm libraries, data sources, online bibliographic resources, and professional consulting services.

3 My opinion

This unique book focuses on facilitating the process of producing efficient algorithmic solutions to real-world problems by professional programmers. It does so by presenting a collection of ready-made solutions along with information about their efficiency and common applications to be adapted for the problem at hand. If no ready-made solution can be made to fit, it provides a framework for exhaustive search that has been proven convincingly effective in solving Sudoku, studying chessboard positions, and discovering counter-examples to mathematical conjectures. It also draws attention to problem modeling, result validation, and hardware concerns that are not often addressed in a traditional algorithms textbook.

I enjoy especially the chapters on combinatorial search, dynamic programming, and dealing with hardness. Although the informal but engaging style works for the most part, the presentation of graphs, randomized analysis, and NP-completeness could be improved with a bit more rigor.

This book serves very well its intended audience, which probably includes a majority of CS students, and should be on the bookshelf of every serious student of algorithms. Its pedagogical approach should be considered as a practical alternative to the standard treatment of algorithms in an undergraduate course.