Algorithms Unlocked

Thomas H. Cormen

The MIT Press
Cambridge, Massachusetts London, England

This book was set in Times Roman and Mathtime Pro 2 by the author and was printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Cormen, Thomas H.
Algorithms Unlocked / Thomas H. Cormen.
p. cm
Includes bibliographical references and index.
ISBN 978-0-262-51880-2 (pbk. : alk. paper)
1. Computer algorithms. I. Title.
QA76.9.A43C685 2013
005.1—dc23

2012036810

10 9 8 7 6 5 4 3 2

Contents

In loving memory of my mother, Renee Cormen.

Preface

How do computers solve problems? How can your little GPS find, out of the gazillions of possible routes, the fastest way to your destination, and do so in mere seconds? When you purchase something on the Internet, how is your credit-card number protected from someone who intercepts it? The answer to these, and a ton of other questions, is ***algorithms***. I wrote this book to unlock the mystery of algorithms for you.

I coauthored the textbook *Introduction to Algorithms*. It's a wonderful book (of course, I'm biased), but it gets pretty technical in spots.

This book is not *Introduction to Algorithms*. It's not even a textbook. It goes neither broadly nor deeply into the field of computer algorithms, it doesn't prescriptively teach techniques for designing computer algorithms, and it contains nary a problem or exercise for the reader to solve.

So just what is this book? It's a place for you to start, if

- you're interested in how computers solve problems,
- you want to know how to evaluate the quality of these solutions,
- you'd like to see how problems in computing and approaches to solving them relate to the non-computer world,
- you can handle a little mathematics, and
- you have not necessarily ever written a computer program (though it doesn't hurt to have programmed).

Some books about computer algorithms are conceptual, with little technical detail. Some are chock full of technical precision. Some are in between. Each type of book has its place. I'd place this book in the in-between category. Yes, it has some math, and it gets rather precise in some places, but I've avoided getting deep into details (except perhaps toward the end of the book, where I just couldn't control myself).

I think of this book as a bit like an antipasto. Suppose you go to an Italian restaurant and order an antipasto, holding off on deciding whether to order the rest of the meal until you've had the antipasto. It arrives, and you eat it. Maybe you don't like the antipasto, and you decide to not order anything else. Maybe you like it, but it fills you up,

so that you don't need to order anything else. Or maybe you like the antipasto, it does not fill you up, and you're looking forward to the rest of the meal. Thinking of this book as the antipasto, I'm hoping for one of the latter two outcomes: either you read this book, you're satisfied, and you feel no need to delve deeper into the world of algorithms; or you like what you read here so much that you want to learn more. Each chapter ends with a section titled "Further reading," which will guide you to books and articles that go deeper into the topics.

What will you learn from this book?

I can't tell you what you will learn from this book. Here's what I *intend* for you to learn from this book:

- What computer algorithms are, one way to describe them, and how to evaluate them.
- Simple ways to search for information in a computer.
- Methods to rearrange information in a computer so that it's in a prescribed order. (We call this task "sorting.")
- How to solve basic problems that we can model in a computer with a mathematical structure known as a "graph." Among many applications, graphs are great for modeling road networks (which intersections have direct roads to which other intersections, and how long are these roads?), dependencies among tasks (which task must precede which other tasks?), financial relationships (what are the exchange rates among all world currencies?), or interactions among people (who knows whom? who hates whom? which actor appeared in a movie with which other actor?).
- How to solve problems that ask questions about strings of textual characters. Some of these problems have applications in areas such as biology, where the characters represent base molecules and the strings of characters represent DNA structure.
- The basic principles behind cryptography. Even if you have never encrypted a message yourself, your computer probably has (such as when you purchase goods online).
- Fundamental ideas of data compression, going well beyond "f u cn rd ths u cn gt a gd jb n gd pay."

- That some problems are hard to solve on a computer in any reasonable amount of time, or at least that nobody has ever figured out how to do so.

What do you already need to know to understand the material in this book?

As I said earlier, there's some math in this book. If math scares you, then you can try skipping over it, or you can try a less technical book. But I've done my best to make the math accessible.

I don't assume that you've ever written or even read a computer program. If you can follow instructions in outline format, you should be able to understand how I express the steps that, together, form an algorithm. If you get the following joke, you're already part of the way there:

> Did you hear about the computer scientist who got stuck in the shower? He[1] was washing his hair and following the instructions on the shampoo bottle. They read "Lather. Rinse. Repeat."

I've used a fairly informal writing style in this book, hoping that a personal approach will help make the material accessible. Some chapters depend on material in previous chapters, but such dependencies are few. Some chapters start off in a nontechnical manner and become progressively more technical. Even if you find that you're getting in over your head in one chapter, you can probably benefit from reading at least the beginning of the next chapter.

Reporting errors

If you find an error in this book, please let me know about it by sending email to algorithms-unlocked@mit.edu.

Acknowledgments

Much of the material in this book draws from *Introduction to Algorithms*, and so I owe a great deal to my coauthors on that book, Charles Leiserson, Ron Rivest, and Cliff Stein. You'll find that throughout this

---

[1] Or she. Given the unfortunate gender ratio in computer science, chances are it was he.

book, I shamelessly refer to (read: plug) *Introduction to Algorithms*, known far and wide by the initials CLRS of the four authors. Writing this book on my own makes me realize how much I miss collaborating with Charles, Ron, and Cliff. I also transitively thank everyone we thanked in the preface of CLRS.

I also drew on material from courses that I've taught at Dartmouth, especially Computer Science 1, 5, and 25. Thanks to my students for letting me know, by their insightful questions, which pedagogical approaches worked and, by their stony silence, which did not.

This book came to be at the suggestion of Ada Brunstein, who was our editor at the MIT Press when we prepared the third edition of CLRS. Ada has since moved on, and Jim DeWolf took her place. Originally, this book was slated to be part of the MIT Press "Essential Knowledge" series, but the MIT Press deemed it too technical for the series. (Imagine that—I wrote a book too technical for MIT!) Jim handled this potentially awkward situation smoothly, allowing me to write the book that I wanted to write rather than the book that the MIT Press originally thought I was writing. I also appreciate the support of Ellen Faran and Gita Devi Manaktala of the MIT Press.

Julie Sussman, P.P.A., was our technical copyeditor for the second and third editions of CLRS, and I am once again thrilled to have her copyedit this book. Best. Technical. Copyeditor. Ever. She let me get away with nothing. Here's evidence, in the form of part of an email that Julie sent me about an early draft of Chapter 5:

> Dear Mr. Cormen,
>
> Authorities have apprehended an escaped chapter, which has been found hiding in your book. We are unable to determine what book it has escaped from, but we cannot imagine how it could have been lodging in your book for these many months without your knowledge, so we have no option but to hold you responsible. We hope that you will take on the task of reforming this chapter and will give it an opportunity to become a productive citizen of your book. A report from the arresting officer, Julie Sussman, is appended.

In case you're wondering what "P.P.A." stands for, the first two letters are for "Professional Pain." You can probably guess what the "A" stands for, but I want to point out that Julie takes pride in this title, and rightly so. Thanks a googol, Julie!

I am no cryptographer, and the chapter on principles of cryptography benefited tremendously from comments and suggestions by Ron Rivest, Sean Smith, Rachel Miller, and Huijia Rachel Lin. That chapter has a footnote on baseball signs, and I thank Bob Whalen, the baseball coach at Dartmouth, for patiently explaining to me some of the signing systems in baseball. Ilana Arbisser verified that computational biologists align DNA sequences in the way that I explain in Chapter 7. Jim DeWolf and I went through several iterations of titles for this book, but it was an undergraduate student at Dartmouth, Chander Ramesh, who came up with *Algorithms Unlocked*.

The Dartmouth College Department of Computer Science is an awesome place to work. My colleagues are brilliant and collegial, and our professional staff is second to none. If you're looking for a computer science program at the undergraduate or graduate level, or if you seek a faculty position in computer science, I encourage you to apply to Dartmouth.

Finally, I thank my wife, Nicole Cormen; my parents, Renee and Perry Cormen; my sister, Jane Maslin; and Nicole's parents, Colette and Paul Sage, for their love and support. My father is sure that the figure on page 2 is a 5, not an S.

TOM CORMEN

Hanover, New Hampshire
November 2012