



The Art of Computer Programming, Volumes 1-3 Boxed Set

Donald Ervin Knuth

[Download now](#)

[Read Online ➔](#)

The Art of Computer Programming, Volumes 1-3 Boxed Set

Donald Ervin Knuth

The Art of Computer Programming, Volumes 1-3 Boxed Set Donald Ervin Knuth

Knuth's classic work has been widely acclaimed as one of the most influential works in the field of computer science. For the first time, these books are available as a boxed, three-volume set. The handsome slipcase makes this set an ideal gift for the recent computer science graduate or professional programmer. Offering a description of classical computer science, this multi-volume work is a useful resource in programming theory and practice for students, researchers, and practitioners alike. For programmers, it offers cookbook solutions to their day-to-day problems.

The Art of Computer Programming, Volumes 1-3 Boxed Set Details

Date : Published October 15th 1998 by Addison-Wesley Professional

ISBN : 9780201485417

Author : Donald Ervin Knuth

Format : Boxed Set - Hardcover 896 pages

Genre : Computer Science, Programming, Science, Technology, Computers, Technical, Reference, Nonfiction



[Download The Art of Computer Programming, Volumes 1-3 Boxed Set ...pdf](#)



[Read Online The Art of Computer Programming, Volumes 1-3 Boxed Set ...pdf](#)

Download and Read Free Online The Art of Computer Programming, Volumes 1-3 Boxed Set Donald Ervin Knuth

From Reader Review The Art of Computer Programming, Volumes 1-3 Boxed Set for online ebook

Steven Raspudic says

Outdated. Get something more modern if you wish to learn algorithms.

Mohammad Huwaidi says

These three volumes are okay to follow, but the following volumes that Knuth did not publish yet are very very hard to grasp.

Charles says

I haven't read these all the way through mind you, but these books along with the five Fascicles make up the greatest programing reference available, not for any language mind you, but programing in general, meta-programing if you well. Algorithms galore, sorting, and searching, and etc. If your looking for a solution to a programing problem its probibly here. Required for intermediate level skill and up, not for beginners.

Appu Shaji says

I got the first edition of this book more than 10 years back. I have read (and re-read) the series in book in parts over the years, and every time I was impressed by the scholarly mastery and precision of the author. This arguably, is `the' most important text in computer science.

AOCP, along with Computer Algorithms by Cormen, Leiserson and Rivest are the first books I turn to whenever I have an upcoming challenge or interview. Pinnacle of Precision!

Carl Thuysbaert says

A classic

Will says

Knuth cannot be compared to anything else. It's both the beginning and the end of computer science -- you could spend months just trying to understand a single chapter. I love it, but I barely understand a single math equation.

John says

Say hello to *the* monograph on computer programming. Don't roll your own without it.

I expect this'll be on my "currently reading" shelf for a few years. It's not exactly quick going.

Curtis Schofield says

I'm going to order this to hold up my monitor and then hold the inner world and dream a dream with you.

Dan Sutton says

As programmers, this is our bible, along with "A method of programming" by E. W. Dijkstra, which somehow manages to condense into its few pages most of what Knuth expresses in these three (now four) large volumes. Nevertheless, "The Art of Computer Programming" represents an absolute in terms of exposition of the process: no programmer should be without the knowledge contained within. Where Dijkstra, the European, represents a terse, quick method of thinking with huge leaps of intuition, fierce intelligence and sparse, elegant beauty, Knuth's approach is American: methodical and complete, moving gradually from one step to another in a procedural fashion. But there is room for beauty here, too, as Knuth slowly unfolds what has been percolating in his mind these last fifty years, during which he has gradually honed these volumes to perfection. Reading them, one feels as though one has become part of that journey.

Ayush Bhat says

just even not being able to fathom these volumes feel so good :)

Ean Schuessler says

This is the Harry Potter series for REAL wizards.

Matt says

Some people consider these books a sort of *bible* for computer programmers. I wouldn't go so far. A bible always assumes some form of religion. And that would make the author a kind of deity, a pedestal upon

which he certainly would not want to be put. If you like to read a "bible-ish" book by Donald Knuth read 3:16.

I learned a great deal from these books. As a poor student I could only afford to buy Volume 3 (*Sorting and Searching*). The other two I borrowed or read directly at the library (while listening to all the Ook's around me). I bought my copy of *S&S* almost exactly 30 years ago*. Back then algorithms were toys for me. It was fun to play around with them, and Knuth's books helped a lot in understanding algorithms, and improving my own. Later on, after my studies, those toys became tools in problem-solving. And that's what they still are – at least to me.

Other people now use algorithms as weapons, and I'm not even talking about *hackers*. With the advent of "BIG DATA" the possibilities become seemingly infinite. A brave new world awaits us. A world ruled by algorithms. Blessed be Amazon, Google, and the NSA — Hallelujah!

I recommend this set of books to anyone seriously interested in the subject. It's not intended as a pop-science read for laymen. If you get this joke on the *traveling salesman problem* you probably qualify:

*To give you an idea of how long ago this was: In my book I found a punch-card – *pauses* – a *punch card!* – that I used as a bookmark. We actually had to write programs on punch-cards back then. Those were the days.

Carl says

The definitive work on programming; without a doubt there is no more important book on Computer Science. However, it's almost totally impenetrable. I haven't read even a quarter of this, and fully understood much less, but that's nothing to be ashamed of, as probably no one else has either. All the examples are in a made up assembly language, and Knuth invented his own typesetting system to publish it, which became widespread and famous.

The perfect Coffee table book.

Weloytty says

This is one of those sets of books you put on your shelf so that people will recognize you as a Serious Programmer. I don't know anyone who has actually READ them, other than college students who were forced to do it. I actually used it as a reference once when I was writing a sort, but then I tossed my code and used library code, because when it comes down to it, who wants to write a sort by yourself???

I've read some other stuff Knuth has written, and he's actually a pretty good author. I wouldn't start with this, though, unless you're a masochist or you want someone to think that you're a Serious Programmer.

Manel says

I remember finding this book at a university library and, having heard of it, picked it to see why was it special. The book had a profound and lasting effect on the way I had to write computer programs, both academically and professionally. A must for computer science lovers.

Carlo says

Best algorithms reference out there, though it's often quite dense & it's useful to cross-reference with something simpler, like *Algorithms in a Nutshell*.

Jon says

Jesus christ, this is some heavy duty stuff. As a calculus 2 dropout, I might hit the wall early, but I really want to fight through this as much as I can before my head explodes.

Nick Black says

Perfection itself. Every few months I have to go back to my Knuth for some forgotten analysis or modeling, and it's always a savory treat -- I know no other books so dense and overflowing with rare and obscenely useful tricks, so immense in their scope and successful in coverage thereof; there's really nothing approachable in computer science or, so far as I know, in any field (the Hilbert-Courant volumes or Thorne's Gravitation might compare, if their subjects weren't so much vaster than computing, now in only its seventh or eighth decade as a proper discipline).

Don't bother with editions other than the magisterial Third (I've collected three editions, although I've only studied the most recent) -- remember that Knuth took ten years off to develop TeX as a response to the misery of preparing and printing the First. The Third is a beauty, some of the best-bound books in my library and practically indestructible to boot (note that Volume 3 is in only its Second Edition).

Don't get me wrong: TAOCP is not an introductory guide to programming by any means, and it'd have been fantastically difficult (although possible, in a Ramanujan-like way) to learn from these books alone (try *Structure and Interpretation of Computer Programs*, or *A Discipline of Programming*, for that). Indeed, I purchased them before I could put them to full use (and while mathematically, abstractly and computationally underexperienced at 18, I wasn't a tremendous dumbass or anything). Knuth demands, requires, and with epic results justifies a fair level of mathematical sophistication (indeed, the greatest value of early exposure to these classics might be the immediate and forceful demonstration of the absolutely critical role mathematics, abjured by all too many undergraduates brought up on cheesy web programming and mind-rotting video games, plays as a Real Programmer's cynosure and primary tool). Both continuous and discrete analysis are required for following along with the text or solving most of the elegant, absolutely stunning problem sets (Knuth has put more effort into his problems than any author I know, hands down). The sedulous reader will be rewarded many times over.

I've heard complaints about Knuth's use of assembly language; I'm bewildered by these objections and find them fatuous in the extreme. A well-defined assembly language (especially on the MIX/MMIX machines, intended as they are for abstraction) is surely much simpler than the majority of high-level languages, if rather more obscure in its mnemonics. I learned assembly languages for the MosTek 6502 (used in my ATARI 400) and Intel 8086 at the tender ages of 7 and 10, respectively, and found the former a far more sensible and reasonable (not to mention powerful) ontology than an ATARI BASIC that served primarily to eat from your precious 16K of RAM. Let's face a hard truth, gentlefolk: if you can't figure out freakin' assembly language, you'd be better giving up those dreams of code and sticking to blue shirts, smiles and Best Buy's stereo department.

Just as two quick examples of Knuth's power and reach: about a year ago, I needed to set up a multisite scanning platform driven by modular FSM's, designed to perform varied scans of the IPv4 space with an eye towards isotropy. I sought to offer several different targeting schemes, and opened volume 2 wondering semi-absentmindedly "how might one enumerate 2^{32} elements in a pseudorandom fashion on a 32 bit architecture?" Simply, and in $O(1)$ state -- iterate through a linear congruence having period 2^{32} . The necessary constraint equations were there on page 20 or so. I coded up the generating function before lunch that day, turned it into my (awestruck) boss, and it continues even now to form the heart of our active search operations. Secondly, for a recent analysis of a multicore sort viz various cache setups, I was struggling to properly model the transactions -- I opened up Volume 3, and was blinded by 5.3.2 "Minimum-Comparison Sorting", a section I swear I've read multiple times but never thought to apply. My analysis, much finer than it could have been otherwise, was completed within the hour.

Finally, they're uproariously funny at times. Knuth's is an understated and erudite humor, but every few dozen pages he'll toss a riotous throwaway, often two or three levels of reference deep, into the middle of a pile of equations.

notgettingenough says

The mouseover to Randall's cartoon says:

'Dear Reader: Enclosed is a cheque for 98 cents. Using your work, I have proved this equals the amount you requested.'

Nick provides a proper review...<http://www.goodreads.com/review/show/...>

Oh. It's full of jokes, by the way. Just as well I didn't have my cup of tea in hand when I came upon this one:

4. [M50] Prove that when n is an integer, $n > 2$, the equation $x^n + y^n = z^n$ has no solution in positive integers x, y, z . (NB the book was published in the 1960s.)

Ali Kalantari says

Reading it as a book for Advanced programming reference. But I think I'll read it completely when it's last day of my life
