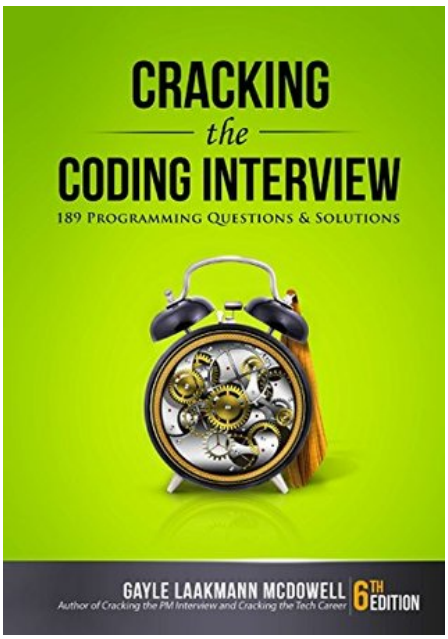

Cracking the Coding Interview: 189 Programming Questions and Solutions

Gayle Laakmann McDowell

Cracking the Coding Interview: 189 Programming Questions and Solutions

Gayle Laakmann McDowell

Cracking the Coding Interview: 189 Programming Questions and Solutions Gayle Laakmann McDowell I am not a recruiter. I am a software engineer. And as such, I know what it's like to be asked to whip up brilliant algorithms on the spot and then write flawless code on a whiteboard. I've been through this as a candidate and as an interviewer.

Cracking the Coding Interview, 6th Edition is here to help you through this process, teaching you what you need to know and enabling you to perform at your very best. I've coached and interviewed hundreds of software engineers. The result is this book.

Learn how to uncover the hints and hidden details in a question, discover how to break down a problem into manageable chunks, develop techniques to unstick yourself when stuck, learn (or re-learn) core computer science concepts, and practice on 189 interview questions and solutions.

These interview questions are real; they are not pulled out of computer science textbooks. They reflect what's truly being asked at the top companies, so that you can be as prepared as possible. **WHAT'S INSIDE?**
• 189 programming interview questions, ranging from the basics to the trickiest algorithm problems.
• A walk-through of how to derive each solution, so that you can learn how to get there yourself.
• Hints on how to solve each of the 189 questions, just like what you would get in a real interview.
• Five proven strategies to tackle algorithm questions, so that you can solve questions you haven't seen.
• Extensive coverage of essential topics, such as big O time, data structures, and core algorithms.
• A behind the scenes look at how top companies like Google and Facebook hire developers.
• Techniques to prepare for and ace the soft side of the interview: behavioral questions.
• For interviewers and companies: details on what makes a good interview question and hiring process.

Cracking the Coding Interview: 189 Programming Questions and Solutions Details

Date	:	Published July 1st 2015 by Careercup (first published October 14th 2008)
ISBN	:	9780984782857
Author	:	Gayle Laakmann McDowell
Format	:	Paperback 708 pages
Genre	:	Computer Science, Programming, Science, Technical, Nonfiction, Coding, Technology

From Reader Review Cracking the Coding Interview: 189 Programming Questions and Solutions for online ebook

Tim O'Hearn says

I've been through this book, front to back, many times, and, this time, I've shelved it with great optimism. So here's to you, Gayle. You've helped mediocre students like me to chase their wildest dreams without concession.

Reviewing CtCI has traditionally been considered the best way to test Software Engineering fundamentals among those entering the industry. With the popularization of interactive platforms like HackerRank and Leetcode, it has become a rite of passage among undergraduates to grind practice problems for hundreds of hours. The result of this leading to highly-desireable career outcomes is that 1. You have a lot of people studying Computer Science for the wrong reason (E.G. CS is the new pre-med) and 2. Most other skills that are important for those working in the industry are deemphasized. 3. There is increased homogeneity in an almost exclusively negative sense.

I don't place the blame on Gayle for what has happened nor do I blame companies that use permutations of these classic problems to screen tens of thousands of candidates. Many people who studied Computer Science did so because they saw it as the best way to satisfy their desire to solve interesting problems and hate it if you must, but this is just a book filled with fun little problems to solve. You signed up for it! Getting toasted in a Facebook interview doesn't mean that the candidate evaluation process is broken!

Anyway, it's a classic book. People probably recommend "better" things now but, as you can probably imagine, most of the problems you'll find elsewhere will be similar if not identical. I'm getting a bit old for this now, though, and I'm hoping I never have to open the book again.

Visit my blog for this and other book reviews

Morgane says

I suppose it's fair to say I'll never quite be "finished" reading this, but regardless, I have an opinion of it.

Laakmann (McDowell? Which last name do I choose?) does a thorough job of breaking down different kinds of interview questions, from brain teasers to OO questions, and there's a good breadth of difficulty as well. My only gripe is that it seems like this book is more for the Java-enthusiast programmer who is dying to interview at Microsoft, rather than... anyone else. It doesn't help that the solutions are all in Java. What is this, the '90s?

So, take this book with a grain of salt. I've noticed more companies moving towards a pair-programming style of interview, which focus on your skills as a *programmer*, not a *computer scientist*. Of course, it just depends where you interview.

Java is so gross to look at.

Venkat Pedapati says

The list of questions and solutions is not comprehensive, but I guess that is the point. Coding interviews are about judging your approach to problems rather than specific solutions. I found that some of the problems were quite simple compared to the difficulty level currently in force at various companies. In particular would like to see more dynamic programming problems.

But the way the solutions are laid out is amazing. Although you should try to solve it yourself before reading the solutions, once you have solved it, you should read the solutions anyway, because he explains the approach he has taken to solve the problems and that turns out to be very valuable.

Overall, very useful - not just for people looking for change, but for every one looking to keep their mind sharp and fresh.

Aaron Sun says

Did not help me crack the interview

A.N. Mignan says

Targeted to the software engineer, the data scientist will also find some valuable information in this book. The first chapters are general enough that the description of the interview process, difference between different tech companies, behavioural questions, etc, shall apply to both engineers and scientists. Then I really enjoyed the sections on math and logic puzzles and the advanced topics on math. Of course, the bulk of the book, on coding, will mainly interest the software engineer.

Felipe says

This book attempts to define *the* format for programming interviews, equating interviewing with competitive programming, which is the type of programming used at events like IEEE Xtreme or ACM ICPC (among others).

Competitive programming has a lot of merit, and provides a tangible demonstration of some of the programmer's abilities, but in many cases, high performance in these tasks will not be a predictor of on-the-job performance. In this regard, "Cracking the Coding Interview" is a bit of a disservice to the software engineering interviewing process.

If you wish however, to get better at this interviewing format, I would rather recommend the following resources:
- Learning: Sedgewick's algorithms book and video course.
- Exercises: InterviewBit, Topcoder.

In addition to that, a competitive programming book. I recommend "Competitive Programmer's Handbook" by Antti Laaksonen (https://cses.fi/book.html), which provides programmer to programmer advice, in a linear, easy to follow format, with code examples.

Jeff says

A useful guide to the interview process, followed by 150 exercises in a number of subfields relevant to programmers. The book straight-forwardly is what it says it is, and accomplishes what it sets out to accomplish; I do believe if you study the material therein, you will be better prepared for programming interviews. If you are already prepared to this level, you will quickly confirm that during your read of the book, which then makes you more confident in your abilities and what to expect at your upcoming interviews.

If you're paranoid/thorough, go ahead and work through more problems, such as those found in Elements of Programming Interviews, Programming Interviews Exposed, Project Euler, TopCoder, HackerRank, and so on.

---

afloatingpoint says

I have not finished all the exercises in here, but the book did give me a good idea what to expect in a tech interview. Not related to the book, but I found this complete, comprehensive solution set in other languages on github: https://github.com/gaylemcd/ctci

---

Zeyuan Hu says

This makes me not want to be a programmer or a developer.

David says

This is a must read book for all engineers looking for work. It goes over how to act as well as coding/behavioral questions. A related site for more examples can be found here: http://www.mycareerstack.com/questions/

Again, this is a must read book (and fast read). It will let you know what areas to improve on and which ones are normally asked (and they really do get asked!)

Mohamed Elsherif says

For better or for worse, this book is a must read for any developer, at least from pragmatic point of view, I personally don't see the current interview practices any good, but reality is reality.

Danail Nachev says

This book is a must have for any software engineer. It doesn't matter whether you already have the perfect job or you are a college graduate, who needs to find your first job - it provides a good base for how programming interviews are done in the majority of the software companies.

The book is structured in two parts: overview of how interviews for software engineers proceed and actual coding exercises with hints and complete answers.

The overview of the interview process focuses on how most of the interviews in the software companies are done and confirming the fact that majority of the time in these interviews is spent actually coding on whiteboard, which for bad or good is quite different than coding on a computer. Probably the main take aways are:

The goal of the coding in the interview process is not to find out what you know, but how you approach and think through a problem, which you've never encountered before. Observing what you know is a side-effect of facing unfamiliar problem, which often requires application of knowledge you already know.
No one expects you to solve a problem immediately or in one try. The goal is to see how you react to unfamiliar situation - ability to find information, analyze the problem looking for a starting point, etc.
On similar note, no one expects to be able to answer all the questions. You are graded not based on whether you answered all questions or not, but on how well you perform against other interview candidates.

Maxim Perepelitsyn says

I am very negative about hiring practices in the US software industry, where often for a pizza delivery kind of job they interview as if they need a space ship pilot. And this book and many similar ones are culmination of this flawed approach, that forces people to memorize tricky tasks and their solutions instead of developing strong CS and Math backgrounds.

Overall the book was OK to get a taste of what insanity to expect on coding interviews. But almost every topic in the book is covered much better elsewhere.

Ugur says

Cracking the Coding Interview is one of the best books to remember old topics. If you are still coding or like coding this book is a great resource to practice old topics. Most of the topics are not too detailed but it is good to check the abstracts.

After reading this book, you probably need more detailed books for each subject.

- Arrays and Strings
- Linked Lists
- Stacks and Queues

- Trees and Graphs
- Bit Manipulation
- Brain Teasers
- Mathematics and Probability
- Object-Oriented Design
- Recursion and Dynamic Programming
- Sorting and Searching
- Scalability and Memory Limits
- Testing
- C and C++
- Java
- Databases
- Threads and Locks

Reviewed on December 29, 2014
Overall: 4.0/5.0

Heather says

This is a great interview prep book, and I would recommend catering your focus to the types of companies you're interviewing with.

In general, most companies probably aren't going to give you questions from math/probability and brain teasers (the exception is the big-name companies the author mentions, Microsoft, Google, Amazon, etc.).

I actually brushed up on the java section last night before an interview today, and believe it or not, I was asked about a topic from the java overview (I was thankful I had the refresher!). For that reason alone, I'm giving this a 4 star rating :-)
