

Chapter 1 : Contents - CRACKING the CODING INTERVIEW

Cracking the Coding Interview: Programming Questions and Solutions () Now in the 4th edition, Cracking the Coding Interview gives you the interview preparation you need to get the top software developer jobs.

Now in the 5th edition, Cracking the Coding Interview gives you the interview preparation you need to get the top software developer jobs. This is a deeply technical book and focuses on the software engineering skills to ace your interview. The book is over pages and includes programming interview questions and answers, as well as other advice. The full list of topics are as follows:

- The Interview Process**This section offers an overview on questions are selected and how you will be evaluated. What happens when you get a question wrong? When should you start preparing, and how? What language should you use? All these questions and more are answered.
- Behind the Scenes**Learn what happens behind the scenes during your interview, how decisions really get made, who you interview with, and what they ask you. Learn what your interviewers are looking for and how much code you need to know.
- Before the Interview**In order to ace the interview, you first need to get an interview.
- Behavioral Preparation**Although most of a software engineering interview will be technical, behavioral questions matter too. This section covers how to prepare for behavioral questions and how to give strong, structured responses. It also teaches you what exactly good coding is when it comes to an interview.

Each section opens with a discussion of the core knowledge and strategies to tackle this type of question, diving into exactly how you break down and solve it. A link is provided with the book so that you can download, compile, and play with the solutions yourself.

Changes from the Fourth Edition: The fifth edition includes over pages of new content, bringing the book from pages to over pages. Major revisions were done to almost every solution, including a number of alternate solutions added. The introductory chapters were massively expanded, as were the opening of each of the chapters under Technical Questions. In addition, 24 new questions were added.

Chapter 2 : Cracking the Coding Interview - Gayle Laakmann - Google Books

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.

I moved to TA for other classes, and I finally launched my very own computer science class there, concentrated on hands-on abilities. As an engineer at Google, coaching and mentoring new engineers have been a number of the things that I enjoyed most. I used my time to instruct two computer science classes in the University of Washington. I have seen their struggles and mistakes, and I have developed strategies and techniques to help them fight these very difficulties. Even today, you may frequently find me hanging out in, assisting customers that stop by for assistance. At many of the best tech firms and a number of other businesses. Algorithm and coding issues form the largest component of this interview procedure. Consider them as work-related questions. The interviewer wants to evaluate your capability to solve algorithmic issues you have not seen previously. Frequently, you may undergo just 1 question in a meeting. You should do your very best to speak out loudly throughout the issue and clarify your thought procedure. Your interviewer may jump in occasionally to assist you; allow them. It is normal and does not mean that you are doing badly. Nevertheless, obviously not needing signs is better. At the close of the interview, the interviewer will walk off with a gut feel for the way you did. A numerical score may be delegated to your own performance, but it is not really a qualitative assessment. There is no graph that states the number of things you get for various items. It simply does not function like that. Download this Cracking the coding interview 6th Edition pdf book for free below. If you see error "Failed to load PDF document or blank page". Please bear in mind that we do not own copyrights to this book. We highly encourage our visitors to purchase original books from the respected publishers. If someone with copyrights wants us to remove this content, please contact us immediately. All books on the cibook. If you feel that we have violated your copyrights, then please contact us immediately [click here](#).

*Now in the 4th edition, Cracking the Coding Interview gives you the interview preparation you need to get the top software developer jobs. This book provides: This book provides: * Programming Interview Questions and Solutions: From binary trees to binary search, this list of questions includes the most common and most useful questions in data structures, algorithms, and knowledge based questions.*

Where did the numbers come from? They are the ages of customers. And how many customers are there? We now have a pretty different problem: How do we solve this? Just create an array with elements and count the number of ages at each value. Design an Algorithm Designing an algorithm can be tough, but our five approaches to algorithms can help you out see pg Pseudo-Code Writing pseudo-code first can help you outline your thoughts clearly and reduce the number of mistakes you commit. Just go at a nice, slow methodical pace. Also, remember this advice: Where relevant, use a good data structure or define your own. This is a minor thing, but it can really help. This will give you plenty of space to write your answer. Test Yes, you need to test your code! What happens if the user passes in null or a negative value? Test the normal case. Also, when you find mistakes which you will , carefully think through why the bug is occurring. For example, imagine a candidate writes a function that returns a number. Not only does this look bad, but it also sends the candidate on an endless string of bugs and bug fixes. When you notice problems in your code, really think deeply about why your code failed before fixing the mistake. Keep in mind that the more problems you practice, the easier it will to identify which approach to use. Write out specific examples of the problem, and see if you can figure out a general rule. Given a time, calculate the angle between the hour and minute hands. Start with an example like 3: We can draw a picture of a clock by selecting where the 3 hour hand is and where the 27 minute hand is. By playing around with these examples, we can develop a rule: Consider what problems the algorithm is similar to, and figure out if you can modify the solution to develop an algorithm for this problem. A sorted array has been rotated so that the elements might appear in the order 3 4 5 6 7 1 2. How would you find the minimum element? However, binary search is very applicable. You know that the array is sorted, but rotated. So, it must proceed in an increasing order, then reset and increase again. Use a recursive algorithm. Then, insert s[n] into every location of the string. Base Case and Build Algorithms often lead to natural recursive algorithms. This is hacky, but it often works. Simply run through a list of data structures and try to apply each one. Numbers are randomly generated and stored into an expanding array. How would you keep track of the median? Probably not

” linked lists tend not to do very well with accessing and sorting numbers. Maybe, but you already have an array. Could you somehow keep the elements sorted? This is possible, since binary trees do fairly well with ordering. In fact, if the binary search tree is perfectly balanced, the top might be the median. A heap is really good at basic ordering and keeping track of max and mins. This is actually interesting

” if you had two heaps, you could keep track of the biggest half and the smallest half of the elements. The biggest half is kept in a min heap, such that the smallest element in the biggest half is at the root. The smallest half is kept in a max heap, such that the biggest element of the smallest half is at the root. Now, with these data structures, you have the potential median elements at the roots. Note that the more problems you do, the better instinct you will develop about which data structure to apply. You got the offer! We have some other advice for you. Ok, maybe not always, but usually an offer is negotiable even if a recruiter tells you otherwise. It helps if you have a competing offer. Also, technology is a small world, and people talk. Think about the full offer package. Many companies will have impressive salaries, but small annual bonuses. Other companies will have huge annual bonuses, but lower salaries. Make sure you look at the full package salary, signing bonus, health care benefits, raises, annual bonus, relocation, stock, promotions, etc. What about your career options? Even if money is all that matters, think about the long term career. Make a plan for your career. What do you want to do 5, 10 and 15 years out? What skills will you need to develop? Which company or position will help you get there? Do you prefer to move into management, or would you prefer to become an increasingly senior developer? As mentioned above, make sure you look at the full package. Did you like the people? Some company names will open doors, while others will not as much. What

about company stability? Personally, I think it matters much less than most people think. There are so many software companies out there. If you get laid off and need to find a new job, will it be difficult to find a new one? Only you can answer that. On the job, and beyond Before starting at a company, make a career plan. What would you like your career to look like? What will it take to get there? Make sure you check in on your career plan regularly and are on track. If what you want is to stay an engineer for life, then there is absolutely nothing wrong with that. However, if you want to run a company one day, or move up into management, you should stop and check your career plan. Is another year at your job going to help you get there? Or is it time to move? You, and only you, can decide. The air is different. The terrain is different. Put away the compiler and get out the old pen and paper. Use a compiler only to verify your solutions. Your interviewer is judging those too! Behavioral prep is relatively easy and well-worth your time. Looking over your projects and positions and think of the key stories. See pg 29 for more details. Your whole school, or company, or whatever will be there. Your future depends on this. And all you do to prepare is read the speech to yourself. Not doing a mock interview to prepare for your real interview is just like this. You can even return the favor! Real preparation is about learning how to approach new problems. Speak up often, and try to talk your way through a solution.

Chapter 4 : Cracking the Coding Interview, 4th Edition - PDF Free Download - Fox eBook

Cracking the Coding Interview is the result of my first-hand experience interviewing at top companies. It is the result of hundreds of conversations with candidates. It is the result of the thousands of candidate- and interviewer- contributed questions.

This is a deeply technical book and focuses on the software engineering skills to ace your interview. The book is over pages and includes programming interview questions and answers, as well as other advice. Changes from the Fourth Edition: The fifth edition includes over pages of new content, bringing the book from pages to over pages. Major revisions were done to almost every solution, including a number of alternate solutions added. The introductory chapters were massively expanded, as were the opening of each of the chapters under Technical Questions. In addition, 24 new questions were added. Cracking the PM Interview: How do you design an alarm clock for the blind? What is your favorite piece of software and why? How would you launch a video rental service in India? This book will teach you how to answer these questions and more. Cracking the PM Interview is a comprehensive book about landing a product management role in a startup or bigger tech company. Scoring those coveted positions requires brains, drive, a minimum of luck, and an abundance of smart moves. Get crucial advice on choosing a major, minor, internship, and part-time job that will make you a superior candidate. Identify your weaknesses, and start developing a track record of achievement now. The hiring processes and requirements of the tech industry are unique and the competition, stiff. No matter your tech savvy, there are infinite ways to improve your chances of landing your dream job, from picking relevant extracurricular activities to striking the right tone during an interview. But if you want to stand out among the roiling mass of tech strivers, you have to do everything right. She interviewed over candidates in the U. She assessed over candidates in that role, and evaluated hundreds more resumes. In , McDowell founded CareerCup. Launched first as a free forum for interview questions, CareerCup now offers a book, a video and mock interviews.

Chapter 5 : Cracking the Coding Interview Solutions | CareerCup

"Cracking the coding interview" (4th ed,) by Gayle Laakman is a good book, but so is the very similar book "Programming Interviews Exposed: Secrets to Landing Your Next Job" (2nd edn), John Mongan, Noah Suojanen and Eric Giguère (2nd edn,).

From binary trees to binary search, this list of questions includes the most common and most useful questions in data structures, algorithms, and knowledge based questions. Learn what many candidates do wrong, and how to avoid these issues. Stop meandering through an endless set of questions, while missing some of the most important preparation techniques. Follow these steps to more thoroughly prepare in less time. Humorous but instructive stories from our interviewers show you how some candidates really flopped on the most important question - and how you can avoid doing the same. Customer Book Reviews Nice collection of problems, but unpolished and full of errors, lacks explanation. I decided to interview at a promising early-stage startup and thus bought two interview books this one and Programming Interview Exposed to brush up on things and get back that interviewing magic. I had been at my last job for over 5 years, so it has been a while. This book is good in some ways. The best thing about it is the set of problems in it. Even the knowledge-based chapters e. So why only two stars? The biggest issue is that there are tons of errors in this book. Some are egregious -- the smart pointer implementation, which is compressed to one page in the solution section, is utterly broken. That is a completely wrong solution to the problem! What if an inexperienced reader reads this section and becomes confused? Books like this should not have mistakes, especially not a 4th edition. Flipping to the solution reveals what the problem was, but then what was the point? In general, when reading this book, once gets the sense that it was thrown together. Very cheap and kind of unprofessional. So, there you go. Having conducted dev interviews before, I wanted to be as prepared as possible. I used this book as a practice guide for my interviews with Microsoft, Amazon, and Google. Got offered to two of the companies through practicing the questions in this book over and over again. I was only caught off guard by one or two questions during my interviews through at least 15 interview loops. Most of the interview questions at the above companies follow the same two dozen or so coding patterns and the questions in this book thoroughly covers them. Which should you choose? Here are the main differences: Another nice thing is that "Exposed" presents the solution in stages, so you can read the first part of the solution to get a hint, and then try again. They are sorted in order of increasing difficulty, which is helpful. In addition, many explanations would be much clearer with an example. There is also the good book "Algorithms For Interviews" by Adnan Aziz and Amit Prakash , which contains much harder questions, focusing on clever algorithms and not on implementation details. When I started my preparation, the entire process seemed really daunting. There is so much to study, and there is vast amount of study-material available out there. A friend of mine suggested me this book and so I read it. To summarize, this book is an awesome compilation of problems, strategies and approaches you must know for doing well in a Tech interview. It starts with chapters focusing on non-coding aspects company-info, resume, behavioral etc. After that, it contains questions grouped by categories, covering all the relevant areas. And finally it has useful information, solutions and discussions for all the questions. Rather, if you slog through these questions, learning techniques; I am sure you will be able to solve questions on that day. Solving problems through the book, I learnt different [ingenious] approaches. Because they contain some really neat algorithms: Hope this review helps.. And all the best for your interview: Still have many errors even in 4th edition By Laozi on Sep 23, In general this is a good book to cover technical interview questions and some very useful tips for how to prepare and act for the on-site interview on technical questions. However, there are still several obvious errors in the books, especially codes and some formulas. For example, in page 47 Question: What is the running time of this code? The two sample codes are exactly the same and the first one should be something not using StringBuffer. Jones on Oct 03, Getting into one of the top companies in the software industry, is not normally an easy task. This book will prepare you for what to expect from companies like Microsoft, Google, Amazon, Yahoo and Apple. Then it will give you the tools to solve the problems that they will throw at you during an interview.

This book helped me get into Microsoft. Cover Theory along with Questions By Anuradha Uduwage on Oct 27, This is a great book regardless you plan to go for an interview or not. Every chapter of the book starts with basic theory which will allow the reader to wake up the brain before going for the questions. Each chapter has great interview questions and answers are very well explained. Not only that some questions are given with two answers where the second answer is optimized version of the code. Also first chapter of this book has very good information about the nature of the big software companies such as Amazon, Google, Microsoft. I highly recommend this book regardless you prepare for an interview or not. Cracking the Coding Interview, Fourth Edition: If you read this book and get comfortable with the material that it contains, it should make you feel a lot more confident about getting grilled at your next interview. My main complaint is with the number of mistakes: For instance, problem 5. There are many others. The solution starts if e. Well, it turns out to be harmless, because the code as written returns null and returning null is the right thing to do in this situation. But it certainly fails the test of "say what you mean. Another example is right next door in 4. It looks as though at some point, the code was updated and the comments were not! I felt nimbler in my thinking and coding as a result and learned a few tricks along the way. But I was disappointed at the slipshod approach to quality, especially in a 4th edition. Excellent resource for interviewing By A. Stillman on May 09, Even rock star coders need to prepare for coding interviews, and this book is THE way to do that. The author has interviewed with virtually every tech company out there, worked for what are in my opinion the three coolest tech companies Google, Microsoft and Apple , and spent three years as an interviewer for Google. According to her bio, she interviewed for people at Google alone! The book covers every aspect of the interview process: The advice is very concrete, applicable and useful. I can definitely say that it made a huge difference in my interview process and my subsequent job offers. In addition to all that stuff, the book offers programming interview questions and answers , from the most common basic questions, to advanced algorithm questions. Regardless of how much interviewing experience you have, this book will make you perform substantially better. I found myself more easily able to think through problems, and less likely to make mistakes. I would recommend that any interviewee prep with this book. Must Buy By P. Cortes on May 09, Though I personally have mixed feelings on the value of so-called "coding interviews," the harsh reality is that all the top companies ask programmers to code on the spot. DO NOT walk into one of those interviews without reading this book first. Do I have to debug through the code to understand how it works?! In my opinion, this is unacceptable. Barely started running the examples, and it took me a lot of time. For the first example 1. No comments or description of the code. For the second example 1. Much more than memorizing answers, I found that interview prep is about getting in the right mentality. The mentality is a combination of being able to deeply communicate your experience with technical details , proving your passion for the field, and being able to have an engaging technical discussion with your interviewer that tells her something about your aptitude. Certainly work through the practice problems on the whiteboard and spend some time on glassdoor. If a candidate has unfeigned enthusiasm, it will help to build rapport with the interviewer and enhance the partnership. A partnership is the way you want to enter the interview Your interviewer may be your future colleague, after all. This kind of exchange is more valuable than whipping out the correct answer in a hurry, which gives your interviewer no information about your personality or your skills. Relaxing and being sociable is one of the best things you can do.

Chapter 6 : CRACKING the CODING INTERVIEW - Home

Welcome to the 4th edition of Cracking the Coding Interview This volume updates the 3rd edition with new content and refreshed information Be sure to check out our website, www.careercup.com, to connect with other candidates and to discover new resources.

Chapter 7 : Cracking The Coding Interview 6th Edition Pdf Ebook Download | dogsselfremchiochild

Download Cracking the Coding Interview PDF eBook by Gayle Laakmann McDowell online absolutely free and start

DOWNLOAD PDF CRACKING THE CODING INTERVIEW FOURTH EDITION

reading Cracking the Coding Interview 6th Edition PDF & 5th Edition PDF now as it is the perfect book for any software engineer out there be it a beginner or the experienced one.

Chapter 8 : Cracking the Coding Interview | PDF Free Download

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.

Chapter 9 : [PDF] Cracking the Coding Interview PDF (5th and 6th edition)

Now in the 5th edition, Cracking the Coding Interview gives you the interview preparation you need to get the top software developer jobs. This is a deeply technical book and focuses on the software engineering skills to ace your interview.