Elements Of Programming Interviews The Insiders

Creating Reality in Factual Television analyzes the uneasy interaction between economics, culture, and professional ethics in reality and documentary television storytelling. Through the "frankenbite," an editorial tool that extracts and re-orders the salient elements or single words of a statement, interview, or exchange into a revealing confession or argument, the book explores how and why editors manipulate truth in factual television. The author considers how the editing of documentary television is increasingly following reality television's dictate to entertain instead of inform, how the "real" and the "truth" fall victim to the demand to "tell entertaining stories," and how editors must compromise their professional ethics as a result. Drawing on interviews with 75 North American and European editors that explore their experiences and opinions of reality and documentary television practices, and their views on their responsibilities and loyalties in the field, Creating Reality in Factual Television illuminates the real and potential ethical dilemmas of editorial decision making, the context in which decisions are made, and how editors themselves validate the editing choices to themselves and others. Addressing a dramatic development in contemporary media ecology – the age of "alternative facts" – this book is a useful research tool for scholars and students of documentary film, media literacy, genre studies, media ethics, affect theory, and audience perception.

Ace technical interviews with smart preparation Programming Interviews Exposed is the programmer's ideal first choice for technical interview preparation. Updated to reflect changing techniques and trends, this new fourth edition provides insider guidance on the unique interview process that today's programmers face. Online coding contests are being used to screen candidate pools of thousands, take-home projects have become commonplace, and employers are even evaluating a candidate's public code repositories at GitHub—and with competition becoming increasingly fierce, programmers need to shape themselves into the ideal candidate well in advance of the interview. This book doesn't just give you a collection of questions and answers, it walks you through the process of coming up with the solution so you learn the skills and techniques to shine on whatever problems you're given. This edition combines a thoroughly revised basis in classic questions involving fundamental data structures and algorithms with problems and step-by-step procedures for new topics including probability, data science, statistics, and machine learning which will help you fully prepare for whatever comes your way. Learn what the interviewer needs to hear to move you forward in the process Adopt an effective approach to phone screens with non-technical recruiters Examine common interview problems and tests with expert explanations Be ready to demonstrate your skills verbally, in contests, on GitHub, and more Technical jobs require the skillset, but you won't get hired unless you are able to effectively and efficiently demonstrate that skillset under pressure, in competition with hundreds of others with the same background. Programming Interviews Exposed teaches you the interview skills you need to stand out as the best applicant to help you get the job you want.

Quick solutions to frequently asked algorithms and data structure questions. KEY FEATURES ? Learn how to crack the Data structure and Algorithms Code test using the top 75 questions/solutions discussed in the book. ? Refresher on Python data structures and writing clean, actionable python codes. ? Simplified solutions on translating business problems into executable programs and applications. DESCRIPTION Python is the most popular programming language, and hence, there is a huge demand for Python programmers. Even if you have learnt Python or have done projects on AI, you cannot enter the top companies unless you have cleared the Algorithms and data Structure coding test. This book presents 75 most frequently asked coding questions by top companies of the world. It not only focuses on the solution strategy, but also provides you with the working code. This book will equip you with the skills required for developing and analyzing algorithms for various situations. This book teaches you how to measure Time Complexity, it then provides solutions to questions on the Linked list, Stack, Hash table, and Math. Then you can review questions and solutions based on graph theory and application techniques. Towards the end, you will come across coding questions on advanced topics such as Backtracking, Greedy, Divide and Conquer, and Dynamic Programming. After reading this book, you will successfully pass the python interview with high confidence and passion for exploring python in future. WHAT YOU WILL LEARN ? Design an efficient algorithm to solve the problem. ? Learn to use python tricks to make your program competitive. ? Learn to understand and measure time and space complexity. ? Get solutions to questions based on Searching, Sorting, Graphs, DFS, BFS, Backtracking, Dynamic programming. WHO THIS BOOK IS FOR This book will help professionals and beginners clear the Data structures and Algorithms coding test. Basic knowledge of Python and Data Structures is a must. TABLE OF CONTENTS 1. Lists, binary search and strings 2. Linked lists and stacks 3. Hash table and maths 4. Trees and graphs 5. Depth first search 6. Breadth first search 7. Backtracking 8. Greedy and divide and conquer algorithms 9. Dynamic programming

Daily Coding Problem contains a wide variety of questions inspired by real programming interviews, with in-depth solutions that clearly take you through each core concept. You'll learn about: * Linked Lists * Arrays * Heaps * Trees * Graphs * Randomized Algorithms * Backtracking * Dynamic Programming * Stacks and Queues * Bit Manipulation * System Design

Cracking Programming Interviews

Element of Programming Interview in Java

A Revolutionary Approach to Time and Energy Management for Ambitious Women

Grokking the System Design Interview

The Frankenbite and Other Fakes

Kubernetes Operators

This is the Python version of our book. See the website for links to the C++ and Java version.Have you ever... Wanted to work at an exciting futuristic company? Struggled with an interview problem thatcould have been solved in 15 minutes? Wished you could study real-world computing problems? If so, you need to read Elements of Programming Interviews (EPI). EPI is your comprehensive guide to interviewing for software development roles. The core of EPI is a collection of over 250 problems with detailed solutions. The problems are representative of interview questions asked at leading software companies. The problems are illustrated with 200 figures, 300 tested programs, and 150 additional variants. The book begins with a summary of the nontechnical aspects of interviewing, such as strategies for a great interview, common mistakes, perspectives from the other side of the table, tips on negotiating the best offer, and a guide to the best ways to use EPI. We also provide a summary of data structures, algorithms, and problem solving patterns. Coding problems are presented through a series of chapters on basic and advanced data structures, searching, sorting, algorithm design principles, and concurrency. Each chapter stars with a brief introduction, a case study, top tips, and a review of the most important library methods. This is followed by a broad and thought-provoking set of problems. A practical, fun approach to computer science fundamentals, as seen through the lens of common programming interview questions. Jeff Atwood/Co-founder, Stack Overflow and Discourse

Covers the methodology and state-of-the-art techniques of constrained verification, which is new and popular. It relates constrained verification with the also-hot technology called assertion-based design. Discussed and clarifies language issues, critical to both the above, which will help the implementation of these languages.

"Brimming with ideas. . . . The Origins of Creativity approach[es] creativity scientifically but sensitively, feeling its roots without pulling them out."—Economist In a stirring exploration of human nature recalling his foundational work Consilience, Edward O. Wilson offers a "luminous" (Kirkus Reviews) reflection on the humanities and their integral relationship to science. Both endeavors, Wilson argues, have their roots in human creativity—the defining trait of our species. By studying fields as diverse as paleontology, evolution, and neurobiology, Wilson demonstrates that creative expression began not 10,000 years ago, as we have long assumed, but more than 100,000 years ago in the Paleolithic Age. A provocative investigation into what it means to be human, The Origins of Creativity reveals how the humanities have played an unexamined role in defining our species. With the eloquence, optimism, and pioneering inquiry we have come to expect from our leading biologist, Wilson proposes a transformational "Third Enlightenment" in which the blending of science and humanities will enable a deeper understanding of our human condition, and how it ultimately originated.

The system design interview is considered to be the most complex and most difficult technical job interview by many. Those questions are intimidating, but don't worry. It's just that nobody has taken the time to prepare you systematically. We take the time. We go slow. We draw lots of diagrams and use lots of examples. You'll learn step-by-step, one question at a time.Don't miss out.What's inside?- An insider's take on what interviewers really look for and why.- A 4-step framework for solving any system design interview question.- 16 real system design interview questions with detailed solutions.- 188 diagrams to visually explain how different systems work.

Elements of Programming Interviews

Cracking the Coding Interview, 6th Edition

The Insider's Guide

Get Exceptionally Good at Coding Interviews by Solving One Problem Every Day

Automating the Container Orchestration Platform

Ace the Programming Interview

The pressure is on during the interview process but with the right preparation, you can walk away with your dream job. This classic book uncovers what interviews are really like at America's top software and computer companies and provides you with the tools to succeed in any situation. The authors take you step-by-step through new problems and complex brainteasers they were asked during recent technical interviews. 50 interview scenarios are presented along with in-depth analysis of the possible solutions. The problem-solving p
you'll be able to easily apply what you've learned during crunch time. You'll also find expert tips on what questions to ask, how to approach a problem, and how to recover if you become stuck. All of this will help you ace the interview and get the job you want. What you will learn from this book Tips for effectively completing the job application Ways to prepare for the entire programming interview process How to find the kind of programming job that fits you best Strategies for choosing a solution and what your approach says about you h
interviewing skills so that you can respond to any question or situation Techniques for solving knowledge-based problems, logic puzzles, and programming problems Who this book is for This book is for programmers and developers applying for jobs in the software industry or in IT departments of major corporations. Wrox Beginning guides are crafted to make learning programming languages and technologies easier than you think, providing a structured, tutorial format that will guide you through all the techniques involved.
Algorithms and data structures are much more than abstract concepts. Mastering them enables you to write code that runs faster and more efficiently, which is particularly important for todayâ€™s web and mobile apps. Take a practical approach to data structures and algorithms, with techniques and real-world scenarios that you can use in your daily production code, with examples in JavaScript, Python, and Ruby. This new and revised second edition features new chapters on recursion, dynamic programming, and using Big O in your
notation to measure and articulate the efficiency of your code, and modify your algorithm to make it faster. Find out how your choice of arrays, linked lists, and hash tables can dramatically affect the code you write. Use recursion to solve tricky problems and create algorithms that run exponentially faster than the alternatives. Dig into advanced data structures such as binary trees and graphs to help scale specialized applications such as social networks and mapping software. Youâ€™ll even encounter a single keyword that can give yo
your new skills with exercises in every chapter, along with detailed solutions. Use these techniques today to make your code faster and more scalable.
Searching & sorting algorithms form the back bone of coding acumen of developers. This book comprehensively covers In-depth tutorial & analysis of all major algorithms and techniques used to search and sort across data structures. All major variations of each algorithm (e.g. Ternary, Jump, Exponential, Interpolation are variations of Binary search). 110 real coding interview questions as solved examples and unsolved problems. Case studies of implementation of searching and sorting in language libraries. Introduction to how questio
to answer on online competitive coding and hiring platforms like hackerrank.com, codechef.com, etc. Introduction to data structures.
Data is at the center of many challenges in system design today. Difficult issues need to be figured out, such as scalability, consistency, reliability, efficiency, and maintainability. In addition, we have an overwhelming variety of tools, including relational databases, NoSQL datastores, stream or batch processors, and message brokers. What are the right choices for your application? How do you make sense of all these buzzwords? In this practical and comprehensive guide, author Martin Kleppmann helps you navigate this diverse lands
and cons of various technologies for processing and storing data. Software keeps changing, but the fundamental principles remain the same. With this book, software engineers and architects will learn how to apply those ideas in practice, and how to make full use of data in modern applications. Peer under the hood of the systems you already use, and learn how to use and operate them more effectively Make informed decisions by identifying the strengths and weaknesses of different tools Navigate the trade-offs around consistency, s
complexity Understand the distributed systems research upon which modern databases are built Peek behind the scenes of major online services, and learn from their architectures

How to Prepare for a Career and Land a Job at Apple, Microsoft, Google, or any Top Tech Company

A Developer's Guide to Using Soft Skills to Get Hired

Secrets to Landing Your Next Job

The Elements of Programming Style

Daily Coding Problem

CPT® 2021 Professional Edition is the definitive AMA-authored resource to help health care professionals correctly report and bill medical procedures and services. Providers want accurate reimbursement. Payers want efficient claims processing. Since the CPT® code set is a dynamic, everchanging standard, an outdated codebook does not suffice. Correct reporting and billing of medical procedures and services begins with CPT® 2021 Professional Edition.
Only the AMA, with the help of physicians and other experts in the health care community, creates and maintains the CPT code set. No other publisher can claim that. No other codebook can provide the official guidelines to code medical services and procedures properly. FEATURES AND BENEFITS The CPT® 2021 Professional Edition codebook covers hundreds of code, guideline and text changes and features: CPT® Changes, CPT® Assistant, and Clinical Examples in
Radiology citations -- provides cross-referenced information in popular AMA resources that can enhance your understanding of the CPT code set E/M 2021 code changes - gives guidelines on the updated codes for office or other outpatient and prolonged services section incorporated A comprehensive index -- aids you in locating codes related to a specific procedure, service, anatomic site, condition, synonym, eponym or abbreviation to allow for a clearer,
quicker search Anatomical and procedural illustrations -- help improve coding accuracy and understanding of the anatomy and procedures being discussed Coding tips throughout each section -- improve your understanding of the nuances of the code set Enhanced codebook table of contents -- allows users to perform a quick search of the codebook's entire content without being in a specific section Section-specific table of contents -- provides users with a
tool to navigate more effectively through each section's codes Summary of additions, deletions and revisions -- provides a quick reference to 2020 changes without having to refer to previous editions Multiple appendices -- offer quick reference to additional information and resources that cover such topics as modifiers, clinical examples, add-on codes, vascular families, multianalyte assays and telemedicine services Comprehensive E/M code selection
tables -- aid physicians and coders in assigning the most appropriate evaluation and management codes Adhesive section tabs -- allow you to flag those sections and pages most relevant to your work More full color procedural illustrations Notes pages at the end of every code set section and subsection
Elements of Programming Interviews (EPI) aims to help engineers interviewing for software development positions. The primary focus of EPI is data structures, algorithms, system design, and problem solving. The material is largely presented through questions.
Covers Expression, Structure, Common Blunders, Documentation, & Structured Programming Techniques
Now in the 5th edition, Cracking the Coding Interview gives you the interview preparation you need to get the top software developer jobs. This book provides: 150 Programming Interview Questions and Solutions: From binary trees to binary search, this list of 150 questions includes the most common and most useful questions in data structures, algorithms, and knowledge based questions. 5 Algorithm Approaches: Stop being blind-sided by tough algorithm
questions, and learn these five approaches to tackle the trickiest problems. Behind the Scenes of the interview processes at Google, Amazon, Microsoft, Facebook, Yahoo, and Apple: Learn what really goes on during your interview day and how decisions get made. Ten Mistakes Candidates Make -- And How to Avoid Them: Don't lose your dream job by making these common mistakes. Learn what many candidates do wrong, and how to avoid these issues. Steps to
Prepare for Behavioral and Technical Questions: 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.
Dynamic Programming (with Solutions in Python)
With 100+ Interview questions
Programming Interviews Exposed
Surviving the Whiteboard Interview
Searching & Sorting for Coding Interviews
Constraint-Based Verification

Are you preparing for a programming interview? Would you like to work at one of the Internet giants, such as Google, Facebook, Amazon, Apple, Microsoft or Netflix? Are you looking for a software engineer position? Are you studying computer science or programming? Would you like to improve your programming skills? If the answer to any of these questions is yes, this book is for you! The book contains very detailed answers and explanations for the most common dynamic programming problems asked in programming interviews. The solutions consist of cleanly written co
plenty of comments, accompanied by verbal explanations, hundreds of drawings, diagrams and detailed examples, to help you get a good understanding of even the toughest problems. The goal is for you to learn the patterns and principles needed to solve even dynamic programming problems that you have never seen before. Here is what you will get: A 180-page book presenting dynamic programming problems that are often asked in interviews. Multiple solutions for each problem, starting from simple but naive answers that are gradually improved until reaching the optima
Plenty of detailed examples and walkthroughs, so that you can see right away how the solution works. 350+ drawings and diagrams which cater towards visual learners. Clear and detailed verbal explanations of how to approach the problems and how the code works. Analysis of time and space complexity. Discussion of other variants of the same problem, with solutions. Unit tests, including the reasoning behind choosing each one (edge case identification, performance evaluation etc.). Suggestions regarding what clarification questions you should ask, for each problem. Mult
to the problems, where appropriate. General Python implementation tips. Wishing you the best of luck with your interviews!

Elements of Programming Interviews in PythonEPI

This is a larger-format version of Elements of Programming Interviews in Java. Specifically, the font size is larger, and the page size is 7"x10" (the regular format uses 6"x9"). The content is identical. This is the Java version of our book. See our website for links to the C++ version.Have you ever... Wanted to work at an exciting futuristic company? Struggled with an interview problem thatcould have been solved in 15 minutes? Wished you could study real-world computing problems? If so, you need to read Elements of Programming Interviews (EPI). EPI is your comprehensive
interviewing for software development roles. The core of EPI is a collection of over 250 problems with detailed solutions. The problems are representative of interview questions asked at leading software companies. The problems are illustrated with 200 figures, 300 tested programs, and 150 additional variants. The book begins with a summary of the nontechnical aspects of interviewing, such as strategies for a great interview, common mistakes, perspectives from the other side of the table, tips on negotiating the best offer, and a guide to the best ways to use EPI. We also pr
summary of data structures, algorithms, and problem solving patterns. Coding problems are presented through a series of chapters on basic and advanced data structures, searching, sorting, algorithm design principles, and concurrency. Each chapter stars with a brief introduction, a case study, top tips, and a review of the most important library methods. This is followed by a broad and thought-provoking set of problems. A practical, fun approach to computer science fundamentals, as seen through the lens of common programming interview questions.
I wanted to compute 80th term of the Fibonacci series. I wrote the rampant recursive function, int fib(int n){ return (1==n || 2==n) ? 1 : fib(n-1) + fib(n-2); } and waited for the result. I wait... and wait... and wait... With an 8GB RAM and an Intel i5 CPU, why is it taking so long? I terminated the process and tried computing the 40th term. It took about a second. I put a check and was shocked to find that the above recursive function was called 204,668,309 times while computing the 40th term. More than 200 million times? Is it reporting function calls or scam of some government? :
Dynamic Programming solution computes 100th Fibonacci term in less than fraction of a second, with a single function call, taking linear time and constant extra memory. A recursive solution, usually, neither pass all test cases in a coding competition, nor does it impress the interviewer in an interview of company like Google, Microsoft, etc. The most difficult questions asked in competitions and interviews, are from dynamic programming. This book takes Dynamic Programming head-on. It first explain the concepts with simple examples and then deep dives into complex DP pr

The Google Resume

A Common-Sense Guide to Data Structures and Algorithms, Second Edition

Top Expert-Led Coding Interview Question Bank for Python Aspirants (English Edition)

Elements of Programming Interviews in Python

189 Programming Questions and Solutions (Indian Edition)

Creating Reality in Factual Television

Operators are a way of packaging, deploying, and managing Kubernetes applications. A Kubernetes application doesn't just run on Kubernetes; it's composed and managed in Kubernetes terms. Operators add application-specific operational knowledge to a Kubernetes cluster, making it easier to automate complex, stateful applications and to augment the platform. Operators can coordinate application upgrades seamlessly, react to failures automatically, and streamline repetitive maintenance like backups. Think of Operators as site reliability engineers in software. They work by extending the Kubernetes control plane and API, helping systems integrators, cluster administrators, and application developers reliably deploy and manage key services and components. Using real-world examples, authors Jason Dobies and Joshua Wood demonstrate how to use Operators today and how to create Operators for your applications with the Operator Framework and SDK. Learn how to establish a Kubernetes cluster and deploy an Operator Examine a range of Operators from usage to implementation Explore the three pillars of the Operator Framework: the Operator SDK, the Operator Lifecycle Manager, and Operator Metering Build Operators from the ground up using the Operator SDK Build, package, and run an Operator in development, testing, and production phases Learn how to distribute your Operator for installation on Kubernetes clusters

Be prepared to answer the most relevant interview questionsand land the job Programmers are in demand, but to land the job, you mustdemonstrate knowledge of those things expected by today'semployers. This guide sets you up for success. Not only doesit provide 160 of the most commonly asked interview questions andmodel answers, but it also offers insight into the context andmotivation of hiring managers in today's marketplace. Written by aveteran hiring manager, this book is a comprehensive guide forexperienced and first-time programmers alike. Provides insight into what drives the recruitment process andhow hiring managers think Covers both practical knowledge and recommendations forhandling the interview process Features 160 actual interview questions, including some relatedto code samples that are available for download on a companionwebsite Includes information on landing an interview, preparing acheat-sheet for a phone interview, how to demonstrate yourprogramming wisdom, and more Ace the Programming Interview, like the earlier Wileybestseller Programming Interviews Exposed, helps youapproach the job interview with the confidence that comes frombeing prepared.

The Complete Coding Interview Guide in Java is an all-inclusive solution guide with meticulously crafted questions and answers that will help you crack any Java Developer job. This book will help you build a strong foundation and the skill-set required to confidently appear in the toughest coding interviews.

This book (also available online at www.designgurus.org) by Design Gurus has helped 60k+ readers to crack their system design interview (SDI). System design questions have become a standard part of the software engineering interview process. These interviews determine your ability to work with complex systems and the position and salary you will be offered by the interviewing company. Unfortunately, SDI is difficult for most engineers, partly because they lack experience developing large-scale systems and partly because SDIs are unstructured in nature. Even engineers who've some experience building such systems aren't comfortable with these interviews, mainly due to the open-ended nature of design problems that don't have a standard answer. This book is a comprehensive guide to master SDIs. It was created by hiring managers who have worked for Google, Facebook, Microsoft, and Amazon. The book contains a carefully chosen set of questions that have been repeatedly asked at top companies. What's inside? This book is divided into two parts. The first part includes a step-by-step guide on how to answer a system design question in an interview, followed by famous system design case studies. The second part of the book includes a glossary of system design concepts. Table of Contents First Part: System Design Interviews: A step-by-step guide. Designing a URL Shortening service like TinyURL. Designing Pastebin. Designing Instagram. Designing Dropbox. Designing Facebook Messenger. Designing Twitter. Designing YouTube or Netflix. Designing Typeahead Suggestion. Designing an API Rate Limiter. Designing Twitter Search. Designing a Web Crawler. Designing Facebook's Newsfeed. Designing Yelp or Nearby Friends. Designing Uber backend. Designing Ticketmaster. Second Part: Key Characteristics of Distributed Systems. Load Balancing. Caching. Data Partitioning. Indexes. Proxies. Redundancy and Replication. SQL vs. NoSQL. CAP Theorem. PACELC Theorem. Consistent Hashing. Long-Polling vs. WebSockets vs. Server-Sent Events. Bloom Filters. Quorum. Leader and Follower. Heartbeat. Checksum. About the Authors Designed Gurus is a platform that offers online courses to help software engineers prepare for coding and system design interviews. Learn more about our courses at www.designgurus.org.

Coding Interview Questions

Coding Interviews

Idiomatic Programming for Experienced Developers

300 Questions and Solutions

An effective guide for aspiring Java developers to ace their programming interviews

Questions, Analysis & Solutions

Peeling Data Structures and Algorithms for (Java, Second Edition): * Programming puzzles for interviews * Campus Preparation * Degree/Masters Course Preparation * Instructor's * GATE Preparation * Big job hunters: Microsoft, Google, Amazon, Yahoo, Flip Kart, Adobe, IBM Labs, Citrix, Mentor Graphics, NetApp, Oracle, Webaroo, De-Shaw, Success Factors, Face book, McAfee and many more * Reference Manual for working people

The industry standard whiteboard interview can be daunting for developers. Let's face it: it combines the worst aspects of a typical interview, on-the-spot public speaking, a quiz show, and a dinner party full of strangers judging you—all at once. Brilliant developers can let their nerves get the best of them and completely bomb a whiteboard interview, while inexperienced developers who excel in soft skills can breeze through them. In Surviving the Whiteboard Interview, author William Gant uses his real-world knowledge and expertise to guide you through the psychological roadblocks of a coding test while also providing you with a sample coding challenge. With enough preparation, information, and assured confidence, you can survive a whiteboard interview at any organization. In addition to the benefits listed above, Gant helps you explore how you can create a good soft skills impression that will last beyond the whiteboard test by showing your work ethic, positive attitude, and ability to take and implement criticism effectively. These assets will unequivocally serve other parts of your life outside of an interview context, as well. While Gant does not promise that you will ever truly enjoy interviewing, he does promise to arm you with the proper preparation techniques and knowledge needed to tame the common fears and dread that come along with it. Maximize your career potential and get inspired with Surviving the Whiteboard Interview. The steps to your dream role just might be closer than you think. What You Will Learn Practice both hard and soft skills required to succeed at a whiteboard interview, covering coding tests as well as psychological preparationLearn how to make other aspects of your interview stronger, so you can create a great impressionMaster solving common whiteboard problems in different programming languages Who This Book is For This book is primarily for aspiring software developers who are looking for a job in the field. However, it will also be helpful for more seasoned developers who find interviewing painful and want to improve their skills.

A gargantuan, mind-altering comedy about the Pursuit of Happiness in America Set in an addicts' halfway house and a tennis academy, and featuring the most endearingly screwed-up family to come along in recent fiction, Infinite Jest explores essential questions about what entertainment is and why it has come to so dominate our lives; about how our desire for entertainment affects our need to connect with other people; and about what the pleasures we choose say about who we are. Equal parts philosophical quest and screwball comedy, Infinite Jest bends every rule of fiction without sacrificing for a moment its own entertainment value. It is an exuberant, uniquely American exploration of the passions that make us human - and one of those rare books that renew the idea of what a novel can do. "The next step in fiction...Edgy, accurate, and darkly witty...Think Beckett, think Pynchon, think Gaddis. Think." --Sven Birkerts, The Atlantic

A practical, expert-reviewed guide to growing software engineering teams effectively, written by and for hiring managers, recruiters, interviewers, and candidates.

Rust for Rustaceans

Game Research Methods: An Overview

Cracking the Coding Interview

Elements of Programming

Infinite Jest

150 Programming Interview Questions and Solutions

Have you ever... - Wanted to work at an exciting futuristic company? - Struggled with an interview problem that could have been solved in 15 minutes? - Wished you could study real-world computing problems? If so, you need to read Elements of Programming Interviews (EPI). EPI is your comprehensive guide to interviewing for software development roles. The core of EPI is a collection of over 250 problems with detailed solutions. The problems are representative of interview questions asked at leading software companies. The problems are illustrated with 200 figures, 300 tested programs, and 150 additional variants. The book begins with a summary of the nontechnical aspects of interviewing, such as strategies for a great interview, common mistakes, perspectives from the other side of the table, tips on negotiating the best offer, and a guide to the best ways to use EPI. We also provide a summary of data structures, algorithms, and problem solving patterns. Coding problems are presented through a series of chapters on basic and advanced data structures, searching, sorting, algorithm design principles, and concurrency. Each chapter stars with a brief introduction, a case study, top tips, and a review of the most important library methods. This is followed by a broad and thought-provoking set of problems. A practical, fun approach to computer science fundamentals, as seen through the lens of common programming interview questions. Jeff Atwood/Co-founder, Stack Overflow and Discourse

A practical and spiritual guide for working moms to learn how to have more by doing less. This is a book for working women and mothers who are ready to release the culturally inherited belief that their worth is equal to their productivity, and instead create a personal and professional life that's based on presence, meaning, and joy. As opposed to focusing on "fitting it all in," time management, and leaning in, as so many books geared at ambitious women do, this book embraces the notion that through doing less women can have--and be--more. The addiction to busyness and the obsession with always trying to do more leads women, especially working mothers, to feel like they're always failing their families, their careers, their spouses, and themselves. This book will give women the permission and tools to change the way they approach their lives and allow them to embrace living in tune with the cyclical nature of the feminine, cutting out the extraneous busyness from their lives so they have more satisfaction and joy, and letting themselves be more often instead of doing all the time. Do Less offers the reader a series of 14 experiments to try to see what would happen if she did less in one specific way. So, rather than approaching doing less as an entire life overhaul (which is overwhelming in and of itself), this book gives the reader bite-sized steps to try incorporating over 2 weeks!

This book is about coding interview questions from software and Internet companies. It covers five key factors which determine performance of candidates: (1) the basics of programming languages, data structures and algorithms, (2) approaches to writing code with high quality, (3) tips to solve difficult problems, (4) methods to optimize code, (5) soft skills required in interviews. The basics of languages, algorithms and data structures are discussed as well as questions that explore how to write robust solutions after breaking down problems into manageable pieces. It also includes examples to focus on modeling and creative problem solving. Interview questions from the most popular companies in the IT industry are taken as examples to illustrate the five factors above. Besides solutions, it contains detailed analysis, how interviewers evaluate solutions, as well as why they like or dislike them. The author makes clever use of the fact that interviewees will have limited time to program meaningful solutions which in turn, limits the options an interviewer has. So the author covers those bases. Readers will improve their interview performance after reading this book. It will be beneficial for them even after they get offers, because its topics, such as approaches to analyzing difficult problems, writing robust code and optimizing, are all essential for high-performing coders.

If you are a skilled Java programmer but are concerned about theJava coding interview process, this real-world guide can help youland your next position Java is a popular and powerful language that is a virtualrequirement for businesses making use of IT in their dailyoperations. For Java programmers, this reality offers job securityand a wealth of employment opportunities. But that perfect Javacoding job won't be available if you can't ace the interview. Ifyou are a Java programmer concerned about interviewing, JavaProgramming Interviews Exposed is a great resource to preparefor your next opportunity. Author Noel Markham is both anexperienced Java developer and interviewer, and has loaded his bookwith real examples from interviews he has conducted. Review over 150 real-world Java interview questions you arelikely to encounter Prepare for personality-based interviews as well as highlytechnical interviews Explore related topics, such as middleware frameworks andserver technologies Make use of chapters individually for topic-specific help Use the appendix for tips on Scala and Groovy, two otherlanguages that run on JVMs Veterans of the IT employment space know that interviewing for aJava programming position isn't as simple as sitting down andanswering questions. The technical coding portion of the interviewcan be akin to a difficult puzzle or an interrogation. With JavaProgramming Interviews Exposed, skilled Java coders can preparethemselves for this daunting process and better arm themselves withthe knowledge and interviewing skills necessary to succeed.

Java Programming Interviews Exposed

System Design Interview - An Insider's Guide

Dynamic Programming for Coding Interviews

Designing Data-Intensive Applications

Programming Interview Problems

The Truth is Funny, shift happens...

Master professional-level coding in Rust. For developers who've mastered the basics, this book is the next step on your way to professional-level programming in Rust. It covers everything you need to build and maintain larger code bases, write powerful and flexible applications and libraries, and confidently expand the scope and complexity of your projects. Author Jon Gjengset takes you deep into the Rust programming language, dissecting core topics like ownership, traits, concurrency, and unsafe code. You'll explore key concepts like type layout and trait coherence, delve into the inner workings of concurrent programming and asynchrony with async/await, and take a tour of the world of no_std programming. Gjengset also provides expert guidance on API design, testing strategies, and error handling, and will help develop your understanding of foreign function interfaces, object safety, procedural macros, and much more. You'll Learn: • How to design reliable, idiomatic, and ergonomic Rust programs based on best principles • Effective use of declarative and procedural macros, and the difference between them • How asynchrony works in Rust – all the way from the Pin and Waker types used in manual implementations of Futures, to how async/await saves you from thinking about most of those words • What it means for code to be unsafe, and best practices for writing and interacting with unsafe functions and traits • How to organize and configure more complex Rust projects so that they integrate nicely with the rest of the ecosystem • How to write Rust code that can interoperate with non-Rust libraries and systems, or run in constrained and embedded environments Brimming with practical, pragmatic insights that you can

immediately apply, Rust for Rustaceans helps you do more with Rust, while also teaching you its underlying mechanisms.
Games are increasingly becoming the focus for research due to their cultural and economic impact on modern society. However, there are many different types of approaches and methods than can be applied to understanding games or those that play games. This book provides an introduction to various game research methods that are useful to students in all levels of higher education covering both quantitative, qualitative and mixed methods. In addition, approaches using game development for research is described. Each method is described in its own chapter by a researcher with practical experience of applying the method to topic of games. Through this, the book provides an overview of research methods that enable us to better our understanding on games.
"Coding Interview Questions" is a book that presents interview questions in simple and straightforward manner with a clear-cut explanation. This book will provide an introduction to the basics. It comes handy as an interview and exam guide for computer scientists. Programming puzzles for interviews Campus Preparation Degree/Masters Course Preparation Big job hunters: Apple, Microsoft, Google, Amazon, Yahoo, Flip Kart, Adobe, IBM Labs, Citrix, Mentor Graphics, NetApp, Oracle, Webaroo, De-Shaw, Success Factors, Face book, McAfee and many more Reference Manual for working people Topics Covered: Programming BasicsIntroductionRecursion and BacktrackingLinked Lists Stacks Queues Trees Priority Queue and HeapsGraph AlgorithmsSortingSearching Selection Algorithms [Medians] Symbol TablesHashing String Algorithms Algorithms Design Techniques Greedy Algorithms Divide and Conquer Algorithms Dynamic Programming Complexity Classes Design Interview Questions Operating System Concepts Computer Networking Basics Database Concepts Brain Teasers NonTechnical Help Miscellaneous Concepts Note: If you already have "Data Structures and Algorithms Made Easy" no need to buy this.
Part I Algorithms and Data Structures 1 Fundamentals Approximating the square root of a number Generating Permutation Efficiently Unique 5-bit Sequences Select Kth Smallest Element The Non-Crooks Problem Is this (almost) sorted? Sorting an almost sorted list The Longest Upsequence Problem Fixed size generic array in C++ Seating Problem Segment Problems Exponentiation Searching two-dimensional sorted array Hamming Problem Constant Time Range Query Linear Time Sorting Writing a Value as the Sum of Squares The Celebrity Problem Transport Problem Find Length of the rope Switch Bulb Problem In, On or Out The problem of the balanced seg The problem of the most isolated villages 2 Arrays The Plateau Problem Searching in Two Dimensional Sequence The Welfare Crook Problem 2D Array Rotation A Queuing Problem in A Post Office Interpolation Search Robot Walk Linear Time Sorting Write as sum of consecutive positive numbers Print 2D Array in Spiral Order The Problem of the Circular Racecourse Sparse Array Trick Bulterman's Reshuffling Problem Finding the majority Mode of a Multiset Circular Array Find Median of two sorted arrays Finding the missing integer Finding the missing number with sorted columns Re-arranging an array Switch and Bulb Problem Compute sum of sub-array Find a number not sum of subsets of array Kth Smallest Element in Two Sorted Arrays Sort a sequence of sub-sequences Find missing integer Inplace Reversing Find the number not occurring twice in an array 3 Trees Lowest Common Ancestor(LCA) Problem Spying Campaign 4 Dynamic Programming Stage Coach Problem Matrix Multiplication TSP Problem A Simple Path Problem String Edit Distance Music recognition Max Sub-Array Problem 5 Graphs Reliable distribution Independent Set Party Problem 6 Miscellaneous Compute Next Higher Number Searching in Possibly Empty Two Dimensional Sequence Matching Nuts and Bolts Optimally Random-number generation Weighted Median Compute a^n Compute a^n revisited Compute the product a × b Compute the quotient and remainder Compute GCD Computed Constrained GCD Alternative Euclid' Algorithm Revisit Constrained GCD Compute Square using only addition and subtraction Factorization Factorization Revisited Decimal Representation Reverse Decimal Representation Solve Inequality Solve Inequality Revisited Print Decimal Representation Decimal Period Length Sequence Periodicity Problem Compute Function Emulate Division and Modulus Operations Sorting Array of Strings : Linear Time LRU data structure Exchange Prefix and Suffix 7 Parallel Algorithms Parallel Addition Find Maximum Parallel Prefix Problem Finding Ranks in Linked Lists Finding the k th Smallest Element 8 Low Level Algorithms Manipulating Rightmost Bits Counting 1-Bits Counting the 1-bits in an Array Computing Parity of a word Counting Leading/Trailing 0's Bit Reversal Bit Shuffling Integer Square Root Newton's Method Integer Exponentiation LRU Algorithm Shortest String of 1-Bits Fibonacci words Computation of Power of 2 Round to a known power of 2 Round to Next Power of 2 Efficient Multiplication by Constants Bit-wise Rotation Gray Code Conversion Average of Integers without Overflow Least/Most Significant 1 Bit Next bit Permutation Modulus Division Part II C++ 8 General 9 Constant Expression 10 Type Specifier 11 Namespaces 12 Misc 13 Classes 14 Templates 15 Standard Library
Coding Your Way Through the Interview
The Big Ideas Behind Reliable, Scalable, and Maintainable Systems
The The Complete Coding Interview Guide in Java
The Holloway Guide to Technical Recruiting and Hiring
The Origins of Creativity
Do Less