Grokking Algorithms An Illustrated Guide For Programmers And Other Curious People

As recognized, adventure as competently as experience about lesson, amusement, as well as deal can be gotten by just checking out a book **Grokking Algorithms An Illustrated Guide For Programmers And Other Curious People** after that it is not directly done, you could recognize even more not far off from this life, on the order of the world.

We give you this proper as capably as simple pretentiousness to acquire those all. We present Grokking Algorithms An Illustrated Guide For Programmers And Other Curious People and numerous ebook collections from fictions to scientific research in any way. accompanied by them is this Grokking Algorithms An Illustrated Guide For Programmers And Other Curious People that can be your partner.

Java Concurrency in Practice - Tim Peierls 2006-05-09
Threads are a fundamental part of the Java platform. As multicore processors become the norm, using concurrency effectively becomes essential for building high-performance applications. Java SE 5 and 6 are a huge step forward for the development of concurrent applications, with improvements to the Java Virtual Machine to support high-performance, highly scalable concurrent classes and a rich set of new concurrency building blocks. In Java Concurrency in Practice , the creators of these new facilities explain not only how they work and how to use them, but also the motivation and design patterns behind them. However, developing, testing, and debugging multithreaded programs can still be very difficult; it is all too easy to create concurrent programs that appear to work, but fail when it matters most: in production, under heavy load. Java Concurrency in Practice arms readers with both the theoretical underpinnings and concrete techniques for building reliable, scalable, maintainable concurrent applications. Rather than simply offering an inventory of concurrency APIs and mechanisms, it provides design rules, patterns, and mental models that make it easier to build concurrent programs that are both correct and performant. This book covers: Basic concepts of concurrency and thread safety Techniques for building and composing thread-safe classes Using the concurrency building blocks in java.util.concurrent Performance optimization dos and don'ts Testing concurrent programs Advanced topics such as atomic variables, nonblocking algorithms, and the Java Memory Model

Python Algorithms - Magnus Lie Hetland 2014-09-17
Python Algorithms, Second Edition explains the Python approach to algorithm analysis and design. Written by Magnus Lie Hetland, author of Beginning Python, this book is sharply focused on classical algorithms, but it also gives a solid understanding of fundamental algorithmic problem-solving techniques. The book deals with some of the most important and challenging areas of programming and computer science in a highly readable manner. It covers both algorithmic theory and programming practice, demonstrating how theory is reflected in real Python programs. Well-known algorithms and data structures that are built into the Python language are explained, and the user is shown how to implement and evaluate others.

The Productive Programmer - Neal Ford 2008-07-03
Anyone who develops software for a living needs a proven way to produce it better, faster, and cheaper. The Productive Programmer offers critical timesaving and productivity tools that you can adopt right away, no matter what platform you use. Master developer Neal Ford not only offers advice on the mechanics of productivity-how to work smarter, spurn interruptions, get the most out your computer, and avoid repetition-he also details valuable practices that will help you elude common traps, improve your code, and become more valuable to your team. You'll learn to: Write the test before you write the code Manage the lifecycle of your objects fastidiously Build only what you need now, not what you might need later Apply ancient philosophies to software development Question authority, rather than blindly adhere to standards Make hard things easier and impossible things possible through meta-programming Be sure all code within a method is at the same level of abstraction Pick the right editor and assemble the best tools for the job This isn't theory, but the fruits of Ford's real-world experience as an Application Architect at the global IT consultancy ThoughtWorks. Whether you're a beginner or a pro with years of experience, you'll improve your work and your career with the simple and straightforward principles in The Productive Programmer.

A Common-Sense Guide to Data Structures and Algorithms - Jay Wengrow 2017-08-03
" 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 today's web and mobile apps. This book takes a practical approach to data structures and algorithms, with techniques and real-world scenarios that you can use in your daily production code. Graphics and examples make these computer science concepts understandable and relevant. You can use these techniques with any language; examples in the book are in JavaScript, Python, and Ruby. Use Big O notation, the primary tool for evaluating algorithms, 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. You'll even encounter a single keyword that can give your code a turbo boost. Jay Wengrow brings to this book the key teaching practices he developed as a web development bootcamp founder and educator. Use these techniques today to make your code faster and more scalable. "

Mazes for Programmers - Jamis Buck 2015-07-15
Unlock the secrets to creating random mazes! Whether you're a game developer, an algorithm connoisseur, or simply in search of a new puzzle, you're about to level up. Learn algorithms to randomly generate mazes in a variety of shapes, sizes, and dimensions. Bend them into Moebius strips, fold them into cubes, and wrap them around spheres. Stretch them into other dimensions, squeeze them into arbitrary outlines, and tile them in a dizzying variety of ways. From twelve little algorithms, you'll discover a vast reservoir of ideas and inspiration. From video games to movies, mazes are ubiquitous. Explore a dozen algorithms for generating these puzzles randomly, from Binary Tree to Eller's, each copiously illustrated and accompanied by working implementations in Ruby. You'll learn their pros and cons, and how to choose the right one for the job. You'll start by learning six maze algorithms and transition from making mazes on paper to writing programs that generate and draw them. You'll be introduced to Dijkstra's algorithm and see how it can help solve, analyze, and visualize mazes. Part 2 shows you how to constrain your mazes to different shapes and outlines, such as text, circles, hex and triangle grids, and more. You'll learn techniques for culling dead-ends, and for making your passages weave over and under each other. Part 3 looks at six more algorithms, taking it all to the next level. You'll learn how to build your mazes in multiple dimensions, and even on curved surfaces. Through it all, you'll discover yourself brimming with ideas, the best medicine for programmer's block, burn-out, and the grayest of days. By the time you're done, you'll be energized and full of maze-related possibilities! What You Need: The example code requires version 2 of the Ruby programming language. Some examples depend on the ChunkyPNG library to generate PNG images, and one chapter uses POV-Ray version 3.7 to render 3D graphics.

Graphic Go Algorithms - Yang Hu 2020-06-18
Discover how graph algorithms can help you leverage the relationships within your data to develop more intelligent solutions and enhance your algorithms. You'll learn how graph analytics are uniquely suited to unfold complex structures and reveal difficult-to-find patterns lurking in your data. this book illustrates how graph algorithms deliver data structure and algorithmsThis practical book walks you through hands-on examples of how to use graph algorithms in Go.Learn how graph analytics vary from conventional algorithms analysisUnderstand how classic graph algorithms work, and how they are appliedExplore algorithm examples with working code and sample dada.The complexity of life, because they do not understand to simplify the complex, simple is the beginning of

wisdom. From the essence of practice, this book to briefly explain the concept and vividly cultivate programming interest, you will learn it easy, fast and well

Grokking Machine Learning - Luis Serrano 2021-12-14
Grokking Machine Learning presents machine learning algorithms and techniques in a way that anyone can understand. This book skips the confused academic jargon and offers clear explanations that require only basic algebra. As you go, you'll build interesting projects with Python, including models for spam detection and image recognition. You'll also pick up practical skills for cleaning and preparing data.

Grokking Algorithms - Aditya Bhargava 2016-05-12
Summary Grokking Algorithms is a fully illustrated, friendly guide that teaches you how to apply common algorithms to the practical problems you face every day as a programmer. You'll start with sorting and searching and, as you build up your skills in thinking algorithmically, you'll tackle more complex concerns such as data compression and artificial intelligence. Each carefully presented example includes helpful diagrams and fully annotated code samples in Python. Learning about algorithms doesn't have to be boring! Get a sneak peek at the fun, illustrated, and friendly examples you'll find in Grokking Algorithms on Manning Publications' YouTube channel. Continue your journey into the world of algorithms with Algorithms in Motion, a practical, hands-on video course available exclusively at Manning.com (www.manning.com/livevideo/algorithms-in-motion). Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology An algorithm is nothing more than a step-by-step procedure for solving a problem. The algorithms you'll use most often as a programmer have already been discovered, tested, and proven. If you want to understand them but refuse to slog through dense multipage proofs, this is the book for you. This fully illustrated and engaging guide makes it easy to learn how to use the most important algorithms effectively in your own programs. About the Book Grokking Algorithms is a friendly take on this core computer science topic. In it, you'll learn how to apply common algorithms to the practical programming problems you face every day. You'll start with tasks like sorting and searching. As you build up your skills, you'll tackle more complex problems like data compression and artificial intelligence. Each carefully presented example includes helpful diagrams and fully annotated code samples in Python. By the end of this book, you will have mastered widely applicable algorithms as well as how and when to use them. What's Inside Covers search, sort, and graph algorithms Over 400 pictures with detailed walkthroughs Performance trade-offs between algorithms Python-based code samples About the Reader This easy-to-read, picture-heavy introduction is suitable for self-taught programmers, engineers, or anyone who wants to brush up on algorithms. About the Author Aditya Bhargava is a Software Engineer with a dual background in Computer Science and Fine Arts. He blogs on programming at adit.io. Table of Contents Introduction to algorithms Selection sort Recursion Quicksort Hash tables Breadth-first search Dijkstra's algorithm Greedy algorithms Dynamic programming K-nearest neighbors

Data Structure and Algorithmic Thinking with Python - Narasimha Karumanchi 2015-01-29
It is the Python version of "Data Structures and Algorithms Made Easy." Table of Contents: goo.gl/VLEUca Sample Chapter: goo.gl/8AEcYk Source Code: goo.gl/L8Xxdt The sample chapter should give you a very good idea of the quality and style of our book. In particular, be sure you are comfortable with the level and with our Python coding style. This book focuses on giving solutions for complex problems in data structures and algorithm. It even provides multiple solutions for a single problem, thus familiarizing readers with different possible approaches to the same problem. "Data Structure and Algorithmic Thinking with Python" is designed to give a jump-start to programmers, job hunters and those who are appearing for exams. All the code in this book are written in Python. It contains many programming puzzles that not only encourage analytical thinking, but also prepares readers for interviews. This book, with its focused and practical approach, can help readers quickly pick up the concepts and techniques for developing efficient and effective solutions to problems. Topics covered include: Organization of Chapters Introduction Recursion and Backtracking Linked Lists Stacks Queues Trees Priority Queues and Heaps Disjoint Sets ADT Graph Algorithms Sorting Searching Selection Algorithms [Medians] Symbol Tables Hashing String Algorithms Algorithms Design Techniques Greedy Algorithms Divide and Conquer Algorithms Dynamic Programming Complexity Classes Hacks on Bit-wise Programming Other Programming

Questions
Android Programming - Erik Hellman 2013-11-04
Unleash the power of the Android OS and build the kinds of brilliant, innovative apps users love to use If you already know your way around the Android OS and can build a simple Android app in under an hour, this book is for you. If you're itching to see just how far you can push it and discover what Android is really capable of, it's for you. And if you're ready to learn how to build advanced, intuitive, innovative apps that are a blast to use, this book is definitely for you. From custom views and advanced multi-touch gestures, to integrating online web services and exploiting the latest geofencing and activity recognition features, ace Android developer, Erik Hellman, delivers expert tips, tricks and little-known techniques for pushing the Android envelope so you can: Optimize your components for the smoothest user experience possible Create your own custom Views Push the boundaries of the Android SDK Master Android Studio and Gradle Make optimal use of the Android audio, video and graphics APIs Program in Text-To-Speech and Speech Recognition Make the most of the new Android maps and location API Use Android connectivity technologies to communicate with remote devices Perform background processing Use Android cryptography APIs Find and safely use hidden Android APIs Cloud-enable your applications with Google Play Services Distribute and sell your applications on Google Play Store Learn how to unleash the power of Android and transform your apps from good to great in Android Programming: Pushing the Limits.

Elegant Objects - Yegor Bugayenko 2017-04-18
TL;DR Compound variable names, validators, private static literals, configurable objects, inheritance, annotations, MVC, dependency injection containers, reflection, ORM and even algorithms are our enemies.

A Common-Sense Guide to Data Structures and Algorithms, Second Edition - Jay Wengrow 2020-08-10
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 today'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 daily work. Use Big O 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. You'll even encounter a single keyword that can give your code a turbo boost. Practice your new skills with exercises in every chapter, along with detailed solutions. Use these techniques today to make your code faster and more scalable.

Algorithms C - Yang Hu 2020-08-10
mmers better use the energy of algorithms in daily projects.1. Classic reference book in the field of algorithms: reflects the core knowledge system of algorithms2. Comprehensive content: Comprehensive discussion of sorting, linked list, search, hash, graph and tree algorithms and data structures, covering the algorithms commonly used by every programmer3. The C implementation code, using a modular programming style, gives the actual code of the algorithm.Simple is the beginning of wisdom. From the essence of practice, this book to briefly explain the concept and vividly cultivate programming interest, you will learn it easy, fast and well

Grokking Algorithms - Aditya Y. Bhargava 2016-05-25
Summary Grokking Algorithms is a fully illustrated, friendly guide that teaches you how to apply common algorithms to the practical problems you face every day as a programmer. You'll start with sorting and searching and, as you build up your skills in thinking algorithmically, you'll tackle more complex concerns such as data compression and artificial intelligence. Each carefully presented example includes helpful diagrams and fully annotated code samples in Python. Learning about algorithms doesn't have to be boring! Get a sneak peek at the fun, illustrated, and friendly examples you'll find in Grokking Algorithms on Manning Publications' YouTube channel. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology An algorithm is nothing more than a step-by-step procedure for solving a problem. The algorithms you'll use

most often as a programmer have already been discovered, tested, and proven. If you want to understand them but refuse to slog through dense multipage proofs, this is the book for you. This fully illustrated and engaging guide makes it easy to learn how to use the most important algorithms effectively in your own programs. About the Book Grokking Algorithms is a friendly take on this core computer science topic. In it, you'll learn how to apply common algorithms to the practical programming problems you face every day. You'll start with tasks like sorting and searching. As you build up your skills, you'll tackle more complex problems like data compression and artificial intelligence. Each carefully presented example includes helpful diagrams and fully annotated code samples in Python. By the end of this book, you will have mastered widely applicable algorithms as well as how and when to use them. What's Inside Covers search, sort, and graph algorithms Over 400 pictures with detailed walkthroughs Performance trade-offs between algorithms Python-based code samples About the Reader This easy-to-read, picture-heavy introduction is suitable for self-taught programmers, engineers, or anyone who wants to brush up on algorithms. About the Author Aditya Bhargava is a Software Engineer with a dual background in Computer Science and Fine Arts. He blogs on programming at adit.io. Table of Contents Introduction to algorithms Selection sort Recursion Quicksort Hash tables Breadth-first search Dijkstra's algorithm Greedy algorithms Dynamic programming K-nearest neighbors

Computer Science Distilled - Wladston Ferreira Filho 2017-01-17
A foolproof walkthrough of must-know computer science concepts. A fast guide for those who don't need the academic formality, it goes straight to what differentiates pros from amateurs. First introducing discrete mathematics, then exposing the most common algorithm and data structure design elements, and finally the working principles of computers and programming languages, the book is indicated to all programmers.

JavaScript on Things - Lyza Gardner 2018-02-22
Summary JavaScript on Things is your first step into the exciting and downright entertaining world of programming for small electronics. If you know enough JavaScript to hack a website together, you'll be making things go bleep, blink, and spin faster than you can say "nodebot." Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Are you ready to make things move? If you can build a web app, you can create robots, weather stations, and other funky gadgets! In this incredibly fun, project-based guide, JavaScript hardware hacker Lyza Danger Gardner takes you on an incredible journey from your first flashing LED through atmospheric sensors, motorized rovers, Bluetooth doorbells, and more. With JavaScript, some easy-to-get hardware, and a bit of creativity, you'll be beeping, spinning, and glowing in no time. About the Book JavaScript on Things introduces the exciting world of programming small electronics! You'll start building things immediately, beginning with basic blinking on Arduino. This fully illustrated, hands-on book surveys JavaScript toolkits like Johnny-Five along with platforms including Raspberry Pi, Tessel, and BeagleBone. As you build project after interesting project, you'll learn to wire in sensors, hook up motors, transmit data, and handle user input. So be warned: once you start, you won't want to stop. What's Inside Controlling hardware with JavaScripti Designing and assembling robots and gadgets A crash course in electronics Over a dozen hands-on projects! About the Reader Written for readers with intermediate JavaScript and Node.js skills. No experience with electronics required. About the Author Lyza Danger Gardner has been a web developer for over 20 years. She's part of the NodeBots community and a contributor to the Johnny-Five Node.js library. Table of Contents PART 1 - A JAVASCRIPTER'S INTRODUCTION TO HARDWARE Bringing JavaScript and hardware together Embarking on hardware with Arduino How to build circuits PART 2 - PROJECT BASICS: INPUT AND OUTPUT WITH JOHNNY-FIVE Sensors and input Output: making things happen Output: making things move PART 3 - MORE SOPHISTICATED PROJECTS Serial communication Projects without wires Building your own thing PART 4 - USING JAVASCRIPT WITH HARDWARE IN OTHER ENVIRONMENTS JavaScript and constrained hardware Building with Node.js and tiny computers In the cloud, in the browser, and beyond

Classic Computer Science Problems in Java - David Kopec 2020-12-21
Sharpen your coding skills by exploring established computer science problems! Classic Computer Science Problems in Java challenges you with time-tested scenarios and algorithms. Summary Sharpen your coding skills by exploring established computer science problems! Classic Computer Science Problems in Java challenges you with time-

tested scenarios and algorithms. You'll work through a series of exercises based in computer science fundamentals that are designed to improve your software development abilities, improve your understanding of artificial intelligence, and even prepare you to ace an interview. As you work through examples in search, clustering, graphs, and more, you'll remember important things you've forgotten and discover classic solutions to your "new" problems! Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the technology Whatever software development problem you're facing, odds are someone has already uncovered a solution. This book collects the most useful solutions devised, guiding you through a variety of challenges and tried-and-true problem-solving techniques. The principles and algorithms presented here are guaranteed to save you countless hours in project after project. About the book Classic Computer Science Problems in Java is a master class in computer programming designed around 55 exercises that have been used in computer science classrooms for years. You'll work through hands-on examples as you explore core algorithms, constraint problems, AI applications, and much more. What's inside Recursion, memoization, and bit manipulation Search, graph, and genetic algorithms Constraint-satisfaction problems K-means clustering, neural networks, and adversarial search About the reader For intermediate Java programmers. About the author David Kopec is an assistant professor of Computer Science and Innovation at Champlain College in Burlington, Vermont. Table of Contents 1 Small problems 2 Search problems 3 Constraint-satisfaction problems 4 Graph problems 5 Genetic algorithms 6 K-means clustering 7 Fairly simple neural networks 8 Adversarial search 9 Miscellaneous problems 10 Interview with Brian Goetz

Elements of Programming Interviews - Adnan Aziz 2012-10-11
The core of EPI is a collection of over 300 problems with detailed solutions, including 100 figures, 250 tested programs, and 150 variants. The problems are representative of questions asked at the leading software companies. The book begins with a summary of the nontechnical aspects of interviewing, such as common mistakes, strategies for a great interview, perspectives from the other side of the table, tips on negotiating the best offer, and a guide to the best ways to use EPI. The technical core of EPI is a sequence of chapters on basic and advanced data structures, searching, sorting, broad algorithmic principles, concurrency, and system design. Each chapter consists of a brief review, followed by a broad and thought-provoking series of problems. We include a summary of data structure, algorithm, and problem solving patterns.

Grokking Artificial Intelligence Algorithms - Rishal Hurbans 2020-07-20
"From start to finish, the best book to help you learn AI algorithms and recall why and how you use them." - Linda Ristevski, York Region District School Board "This book takes an impossibly broad area of computer science and communicates what working developers need to understand in a clear and thorough way." - David Jacobs, Product Advance Local Key Features Master the core algorithms of deep learning and AI Build an intuitive understanding of AI problems and solutions Written in simple language, with lots of illustrations and hands-on examples Creative coding exercises, including building a maze puzzle game and exploring drone optimization About The Book "Artificial intelligence" requires teaching a computer how to approach different types of problems in a systematic way. The core of AI is the algorithms that the system uses to do things like identifying objects in an image, interpreting the meaning of text, or looking for patterns in data to spot fraud and other anomalies. Mastering the core algorithms for search, image recognition, and other common tasks is essential to building good AI applications Grokking Artificial Intelligence Algorithms uses illustrations, exercises, and jargon-free explanations to teach fundamental AI concepts.You'll explore coding challenges like detecting bank fraud, creating artistic masterpieces, and setting a self-driving car in motion. All you need is the algebra you remember from high school math class and beginning programming skills. What You Will Learn Use cases for different AI algorithms Intelligent search for decision making Biologically inspired algorithms Machine learning and neural networks Reinforcement learning to build a better robot This Book Is Written For For software developers with high school–level math skills. About the Author Rishal Hurbans is a technologist, startup and AI group founder, and international speaker. Table of Contents 1 Intuition of artificial intelligence 2 Search fundamentals 3 Intelligent search 4 Evolutionary algorithms 5 Advanced evolutionary approaches 6 Swarm intelligence: Ants 7 Swarm intelligence: Particles 8 Machine learning 9 Artificial neural networks 10 Reinforcement learning with Q-learning

Grokking Deep Learning - Andrew W. Trask 2019-01-23
Summary Grokking Deep Learning teaches you to build deep learning neural networks from scratch! In his engaging style, seasoned deep learning expert Andrew Trask shows you the science under the hood, so you grok for yourself every detail of training neural networks. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Deep learning, a branch of artificial intelligence, teaches computers to learn by using neural networks, technology inspired by the human brain. Online text translation, self-driving cars, personalized product recommendations, and virtual voice assistants are just a few of the exciting modern advancements possible thanks to deep learning. About the Book Grokking Deep Learning teaches you to build deep learning neural networks from scratch! In his engaging style, seasoned deep learning expert Andrew Trask shows you the science under the hood, so you grok for yourself every detail of training neural networks. Using only Python and its math-supporting library, NumPy, you'll train your own neural networks to see and understand images, translate text into different languages, and even write like Shakespeare! When you're done, you'll be fully prepared to move on to mastering deep learning frameworks. What's inside The science behind deep learning Building and training your own neural networks Privacy concepts, including federated learning Tips for continuing your pursuit of deep learning About the Reader For readers with high school-level math and intermediate programming skills. About the Author Andrew Trask is a PhD student at Oxford University and a research scientist at DeepMind. Previously, Andrew was a researcher and analytics product manager at Digital Reasoning, where he trained the world's largest artificial neural network and helped guide the analytics roadmap for the Synthesys cognitive computing platform. Table of Contents Introducing deep learning: why you should learn it Fundamental concepts: how do machines learn? Introduction to neural prediction: forward propagation Introduction to neural learning: gradient descent Learning multiple weights at a time: generalizing gradient descent Building your first deep neural network: introduction to backpropagation How to picture neural networks: in your head and on paper Learning signal and ignoring noise:introduction to regularization and batching Modeling probabilities and nonlinearities: activation functions Neural learning about edges and corners: intro to convolutional neural networks Neural networks that understand language: king - man + woman == ? Neural networks that write like Shakespeare: recurrent layers for variable-length data Introducing automatic optimization: let's build a deep learning framework Learning to write like Shakespeare: long short-term memory Deep learning on unseen data: introducing federated learning Where to go from here: a brief guide

Competitive Programming in Python - Christoph Dürr 2020-11-30
Want to kill it at your job interview in the tech industry? Want to win that coding competition? Learn all the algorithmic techniques and programming skills you need from two experienced coaches, problem setters, and jurors for coding competitions. The authors highlight the versatility of each algorithm by considering a variety of problems and show how to implement algorithms in simple and efficient code. Readers can expect to master 128 algorithms in Python and discover the right way to tackle a problem and quickly implement a solution of low complexity. Classic problems like Dijkstra's shortest path algorithm and Knuth-Morris-Pratt's string matching algorithm are featured alongside lesser known data structures like Fenwick trees and Knuth's dancing links. The book provides a framework to tackle algorithmic problem solving, including: Definition, Complexity, Applications, Algorithm, Key Information, Implementation, Variants, In Practice, and Problems. Python code included in the book and on the companion website.

Advanced Algorithms and Data Structures - Marcello La Rocca 2021-06-29
This book introduces a collection of algorithms for complex programming challenges in data analysis, machine learning, and graph computing. Youll discover cutting-edge approaches to a variety of tricky scenarios. --

Understanding Distributed Systems - Roberto Vitillo 2021
Learning to build distributed systems is hard, especially if they are large scale. It's not that there is a lack of information out there. You can find academic papers, engineering blogs, and even books on the subject. The problem is that the available information is spread out all over the place, and if you were to put it on a spectrum from theory to practice, you would find a lot of material at the two ends, but not much in the middle. That is why I decided to write a book to teach the fundamentals of distributed systems so that you don't have to spend countless hours scratching your head to understand how everything fits together. This is the guide I wished existed when I first started out, and it's based on my experience building large distributed systems that scale to millions of requests per second and billions of devices. If you develop the back-end of web or mobile applications (or would like to!), this book is for you. When building distributed systems, you need to be familiar with the network stack, data consistency models, scalability and reliability patterns, and much more. Although you can build applications without knowing any of that, you will end up spending hours debugging and re-designing their architecture, learning lessons that you could have acquired in a much faster and less painful way.

Algorithms in a Nutshell - George T. Heineman 2008-10-14
Creating robust software requires the use of efficient algorithms, but programmers seldom think about them until a problem occurs. Algorithms in a Nutshell describes a large number of existing algorithms for solving a variety of problems, and helps you select and implement the right algorithm for your needs -- with just enough math to let you understand and analyze algorithm performance. With its focus on application, rather than theory, this book provides efficient code solutions in several programming languages that you can easily adapt to a specific project. Each major algorithm is presented in the style of a design pattern that includes information to help you understand why and when the algorithm is appropriate. With this book, you will: Solve a particular coding problem or improve on the performance of an existing solution Quickly locate algorithms that relate to the problems you want to solve, and determine why a particular algorithm is the right one to use Get algorithmic solutions in C, C++, Java, and Ruby with implementation tips Learn the expected performance of an algorithm, and the conditions it needs to perform at its best Discover the impact that similar design decisions have on different algorithms Learn advanced data structures to improve the efficiency of algorithms With Algorithms in a Nutshell, you'll learn how to improve the performance of key algorithms essential for the success of your software applications.

Coders at Work - Peter Seibel 2009-12-21
Peter Seibel interviews 15 of the most interesting computer programmers alive today in Coders at Work, offering a companion volume to Apress's highly acclaimed best-seller Founders at Work by Jessica Livingston. As the words "at work" suggest, Peter Seibel focuses on how his interviewees tackle the day-to-day work of programming, while revealing much more, like how they became great programmers, how they recognize programming talent in others, and what kinds of problems they find most interesting. Hundreds of people have suggested names of programmers to interview on the Coders at Work web site: www.codersatwork.com. The complete list was 284 names. Having digested everyone's feedback, we selected 15 folks who've been kind enough to agree to be interviewed: Frances Allen: Pioneer in optimizing compilers, first woman to win the Turing Award (2006) and first female IBM fellow Joe Armstrong: Inventor of Erlang Joshua Bloch: Author of the Java collections framework, now at Google Bernie Cosell: One of the main software guys behind the original ARPANET IMPs and a master debugger Douglas Crockford: JSON founder, JavaScript architect at Yahoo! L. Peter Deutsch: Author of Ghostscript, implementer of Smalltalk-80 at Xerox PARC and Lisp 1.5 on PDP-1 Brendan Eich: Inventor of JavaScript, CTO of the Mozilla Corporation Brad Fitzpatrick: Writer of LiveJournal, OpenID, memcached, and Perlbal Dan Ingalls: Smalltalk implementor and designer Simon Peyton Jones: Coinventor of Haskell and lead designer of Glasgow Haskell Compiler Donald Knuth: Author of The Art of Computer Programming and creator of TeX Peter Norvig: Director of Research at Google and author of the standard text on AI Guy Steele: Coinventor of Scheme and part of the Common Lisp Gang of Five, currently working on Fortress Ken Thompson: Inventor of UNIX Jamie Zawinski: Author of XEmacs and early Netscape/Mozilla hacker

Algorithmic Puzzles - Anany Levitin 2011-10-14
Algorithmic puzzles are puzzles involving well-defined procedures for solving problems. This book will provide an enjoyable and accessible introduction to algorithmic puzzles that will develop the reader's algorithmic thinking. The first part of this book is a tutorial on algorithm design strategies and analysis techniques. Algorithm design strategies — exhaustive search, backtracking, divide-and-conquer and a few others — are general approaches to designing step-by-step instructions for solving problems. Analysis techniques are methods for investigating such procedures to answer questions about the ultimate result of the procedure or how many steps are executed before the procedure stops. The discussion is an elementary level, with puzzle examples, and requires neither programming nor mathematics beyond a secondary

school level. Thus, the tutorial provides a gentle and entertaining introduction to main ideas in high-level algorithmic problem solving. The second and main part of the book contains 150 puzzles, from centuries-old classics to newcomers often asked during job interviews at computing, engineering, and financial companies. The puzzles are divided into three groups by their difficulty levels. The first fifty puzzles in the Easier Puzzles section require only middle school mathematics. The sixty puzzle of average difficulty and forty harder puzzles require just high school mathematics plus a few topics such as binary numbers and simple recurrences, which are reviewed in the tutorial. All the puzzles are provided with hints, detailed solutions, and brief comments. The comments deal with the puzzle origins and design or analysis techniques used in the solution. The book should be of interest to puzzle lovers, students and teachers of algorithm courses, and persons expecting to be given puzzles during job interviews.

Algorithmic Thinking - Daniel Zingaro 2020-12-15
A hands-on, problem-based introduction to building algorithms and data structures to solve problems with a computer. Algorithmic Thinking will teach you how to solve challenging programming problems and design your own algorithms. Daniel Zingaro, a master teacher, draws his examples from world-class programming competitions like USACO and IOI. You'll learn how to classify problems, choose data structures, and identify appropriate algorithms. You'll also learn how your choice of data structure, whether a hash table, heap, or tree, can affect runtime and speed up your algorithms; and how to adopt powerful strategies like recursion, dynamic programming, and binary search to solve challenging problems. Line-by-line breakdowns of the code will teach you how to use algorithms and data structures like: • The breadth-first search algorithm to find the optimal way to play a board game or find the best way to translate a book • Dijkstra's algorithm to determine how many mice can exit a maze or the number of fastest routes between two locations • The union-find data structure to answer questions about connections in a social network or determine who are friends or enemies • The heap data structure to determine the amount of money given away in a promotion • The hash-table data structure to determine whether snowflakes are unique or identify compound words in a dictionary NOTE: Each problem in this book is available on a programming-judge website. You'll find the site's URL and problem ID in the description. What's better than a free correctness check?

GANs in Action - Vladimir Bok 2019-09-09
Deep learning systems have gotten really great at identifying patterns in text, images, and video. But applications that create realistic images, natural sentences and paragraphs, or native-quality translations have proven elusive. Generative Adversarial Networks, or GANs, offer a promising solution to these challenges by pairing two competing neural networks' one that generates content and the other that rejects samples that are of poor quality. GANs in Action: Deep learning with Generative Adversarial Networks teaches you how to build and train your own generative adversarial networks. First, you'll get an introduction to generative modelling and how GANs work, along with an overview of their potential uses. Then, you'll start building your own simple adversarial system, as you explore the foundation of GAN architecture: the generator and discriminator networks. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications.

Codeless Data Structures and Algorithms - Armstrong Subero 2020-02-13
In the era of self-taught developers and programmers, essential topics in the industry are frequently learned without a formal academic foundation. A solid grasp of data structures and algorithms (DSA) is imperative for anyone looking to do professional software development and engineering, but classes in the subject can be dry or spend too much time on theory and unnecessary readings. Regardless of your programming language background, Codeless Data Structures and Algorithms has you covered. In this book, author Armstrong Subero will help you learn DSAs without writing a single line of code. Straightforward explanations and diagrams give you a confident handle on the topic while ensuring you never have to open your code editor, use a compiler, or look at an integrated development environment. Subero introduces you to linear, tree, and hash data structures and gives you important insights behind the most common algorithms that you can directly apply to your own programs. Codeless Data Structures and Algorithms provides you with the knowledge about DSAs that you will need in the professional programming world, without using any complex mathematics or irrelevant information. Whether you are a new developer seeking a basic understanding of the subject or a decision-maker wanting

a grasp of algorithms to apply to your projects, this book belongs on your shelf. Quite often, a new, refreshing, and unpretentious approach to a topic is all you need to get inspired. What You'll LearnUnderstand tree data structures without delving into unnecessary details or going into too much theoryGet started learning linear data structures with a basic discussion on computer memory Study an overview of arrays, linked lists, stacks and queues Who This Book Is ForThis book is for beginners, self-taught developers and programmers, and anyone who wants to understand data structures and algorithms but don't want to wade through unnecessary details about quirks of a programming language or don't have time to sit and read a massive book on the subject. This book is also useful for non-technical decision-makers who are curious about how algorithms work.

System Design Interview - An Insider's Guide - Alex Xu 2020-06-12
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.

Introduction To Algorithms - Thomas H Cormen 2001
The first edition won the award for Best 1990 Professional and Scholarly Book in Computer Science and Data Processing by the Association of American Publishers. There are books on algorithms that are rigorous but incomplete and others that cover masses of material but lack rigor. Introduction to Algorithms combines rigor and comprehensiveness. The book covers a broad range of algorithms in depth, yet makes their design and analysis accessible to all levels of readers. Each chapter is relatively self-contained and can be used as a unit of study. The algorithms are described in English and in a pseudocode designed to be readable by anyone who has done a little programming. The explanations have been kept elementary without sacrificing depth of coverage or mathematical rigor. The first edition became the standard reference for professionals and a widely used text in universities worldwide. The second edition features new chapters on the role of algorithms, probabilistic analysis and randomized algorithms, and linear programming, as well as extensive revisions to virtually every section of the book. In a subtle but important change, loop invariants are introduced early and used throughout the text to prove algorithm correctness. Without changing the mathematical and analytic focus, the authors have moved much of the mathematical foundations material from Part I to an appendix and have included additional motivational material at the beginning.

Grokking Bitcoin - Kalle Rosenbaum 2019-04-17
Summary If you think Bitcoin is just an alternative currency for geeks, it's time to think again. Grokking Bitcoin opens up this powerful distributed ledger system, exploring the technology that enables applications both for Bitcoin-based financial transactions and using the blockchain for registering physical property ownership. With this fully illustrated, easy-to-read guide, you'll finally understand how Bitcoin works, how you can use it, and why you can trust the blockchain. Foreword by David A. Harding, Contributor to Bitcoin documentation. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology Inflation, depressed economies, debased currencies ... these are just a few of the problems centralized banking has caused throughout history. Bitcoin, a digital currency created with the ambition to shift control away from change-prone governments, has the potential to bring an end to those problems once and for all. It's time to find out how it can help you. About the Book Grokking Bitcoin explains why Bitcoin's supporters trust it so deeply, and why you can too. This approachable book will introduce you to Bitcoin's groundbreaking technology, which is the key to this world-changing system. This illustrated, easy-to-read guide prepares you for a new way of thinking with easy-to-follow diagrams and exercises. You'll discover how Bitcoin mining works, how to accept Bitcoin, how to participate in the Bitcoin network, and how to set up a digital wallet. What's inside Bitcoin transactions The blockchain Bitcoin mining Bitcoin wallets About the Reader Intended for anyone interested in learning about Bitcoin technology. While a basic understanding of technical concepts is beneficial, no programming skills are necessary. About the Author Kalle Rosenbaum is a computer scientist, an avid Bitcoin supporter, and the founder of Propeller, a Bitcoin consultancy. Table of

Contents Introduction to Bitcoin Cryptographic hash functions and digital signatures Addresses Wallets Transactions The blockchain Proof of work Peer-to-peer network Transactions revisited Segregated witness Bitcoin upgrades

Advanced Data Structures - Peter Brass 2019-05-16
Advanced Data Structures presents a comprehensive look at the ideas, analysis, and implementation details of data structures as a specialized topic in applied algorithms. Data structures are how data is stored within a computer, and how one can go about searching for data within. This text examines efficient ways to search and update sets of numbers, intervals, or strings by various data structures, such as search trees, structures for sets of intervals or piece-wise constant functions, orthogonal range search structures, heaps, union-find structures, dynamization and persistence of structures, structures for strings, and hash tables. This is the first volume to show data structures as a crucial algorithmic topic, rather than relegating them as trivial material used to illustrate object-oriented programming methodology, filling a void in the ever-increasing computer science market. Numerous code examples in C and more than 500 references make Advanced Data Structures an indispensable text. topic. Numerous code examples in C and more than 500 references make Advanced Data Structures an indispensable text.

Programming F# 3.0 - Chris Smith 2012-10-09
Why learn F#? With this guide, you'll learn how this multi-paradigm language not only offers you an enormous productivity boost through functional programming, but also lets you develop applications using your existing object-oriented and imperative programming skills. You'll quickly discover the many advantages of the language, including access to all the great tools and libraries of the .NET platform. Reap the benefits of functional programming for your next project, whether you're writing concurrent code, or building data- or math-intensive applications. With this comprehensive book, former F# team member Chris Smith gives you a head start on the fundamentals and walks you through advanced concepts of the F# language. Learn F#'s unique characteristics for building applications Gain a solid understanding of F#'s core syntax, including object-oriented and imperative styles Make your object-oriented code better by applying functional programming patterns Use advanced functional techniques, such as tail-recursion and computation expressions Take advantage of multi-core processors with asynchronous workflows and parallel programming Use new type providers for interacting with web services and information-rich environments Learn how well F# works as a scripting language

Python Algorithms - Magnus Lie Hetland 2011-02-27
Python Algorithms explains the Python approach to algorithm analysis and design. Written by Magnus Lie Hetland, author of Beginning Python, this book is sharply focused on classical algorithms, but it also gives a solid understanding of fundamental algorithmic problem-solving techniques. The book deals with some of the most important and challenging areas of programming and computer science, but in a highly pedagogic and readable manner. The book covers both algorithmic theory and programming practice, demonstrating how theory is reflected in real Python programs. Well-known algorithms and data structures that are built into the Python language are explained, and the user is shown how to implement and evaluate others himself.

Exercises for Programmers - Brian P. Hogan 2015-09-04
When you write software, you need to be at the top of your game. Great programmers practice to keep their skills sharp. Get sharp and stay sharp with more than fifty practice exercises rooted in real-world scenarios. If you're a new programmer, these challenges will help you learn what you need to break into the field, and if you're a seasoned pro, you can use these exercises to learn that hot new language for your next gig. One of the best ways to learn a programming language is to use it to solve problems. That's what this book is all about. Instead of questions rooted in theory, this book presents problems you'll encounter in everyday software development. These problems are designed for people learning their first programming language, and they also provide a learning path for experienced developers to learn a new language quickly. Start with simple input and output programs. Do some currency conversion and figure out how many months it takes to pay off a credit card. Calculate blood alcohol content and determine if it's safe to drive. Replace words in files and filter records, and use web services to display the weather, store data, and show how many people are in space right now. At the end you'll tackle a few larger programs that will help you bring everything together. Each problem includes constraints and challenges to push you further, but it's up to you to come up with the solutions. And next year, when you want to learn a new programming

language or style of programming (perhaps OOP vs. functional), you can work through this book again, using new approaches to solve familiar problems. What You Need: You need access to a computer, a programming language reference, and the programming language you want to use.

Essential Algorithms - Rod Stephens 2013-07-25
A friendly and accessible introduction to the most useful algorithms Computer algorithms are the basic recipes for programming. Professional programmers need to know how to use algorithms to solve difficult programming problems. Written in simple, intuitive English, this book describes how and when to use the most practical classic algorithms, and even how to create new algorithms to meet future needs. The book also includes a collection of questions that can help readers prepare for a programming job interview. Reveals methods for manipulating common data structures such as arrays, linked lists, trees, and networks Addresses advanced data structures such as heaps, 2-3 trees, B-trees Addresses general problem-solving techniques such as branch and bound, divide and conquer, recursion, backtracking, heuristics, and more Reviews sorting and searching, network algorithms, and numerical algorithms Includes general problem-solving techniques such as brute force and exhaustive search, divide and conquer, backtracking, recursion, branch and bound, and more In addition, Essential Algorithms features a companion website that includes full instructor materials to support training or higher ed adoptions.

Learning Algorithms - George Heineman 2021-07-20
When it comes to writing efficient code, every software professional needs to have an effective working knowledge of algorithms. In this practical book, author George Heineman (Algorithms in a Nutshell) provides concise and informative descriptions of key algorithms that improve coding in multiple languages. Software developers, testers, and maintainers will discover how algorithms solve computational problems creatively. Each chapter builds on earlier chapters through eye-catching visuals and a steady rollout of essential concepts, including an algorithm analysis to classify the performance of every algorithm presented in the book. At the end of each chapter, youâ??ll get to apply what youâ??ve learned to a novel challenge problemâ??simulating the experience you might find in a technical code interview. With this book, you will: Examine fundamental algorithms central to computer science and software engineering Learn common strategies for efficient problem solvingâ??such as divide and conquer, dynamic programming, and greedy approaches Analyze code to evaluate time complexity using big O notation Use existing Python libraries and data structures to solve problems using algorithms Understand the main steps of important algorithms

Grokking Algorithms - Aditya Bhargava 2016

40 Algorithms Every Programmer Should Know - Imran Ahmad 2020-06-12
Learn algorithms for solving classic computer science problems with this concise guide covering everything from fundamental algorithms, such as sorting and searching, to modern algorithms used in machine learning and cryptography Key FeaturesLearn the techniques you need to know to design algorithms for solving complex problemsBecome familiar with neural networks and deep learning techniquesExplore different types of algorithms and choose the right data structures for their optimal implementationBook Description Algorithms have always played an important role in both the science and practice of computing. Beyond traditional computing, the ability to use algorithms to solve real-world problems is an important skill that any developer or programmer must have. This book will help you not only to develop the skills to select and use an algorithm to solve real-world problems but also to understand how it works. You'll start with an introduction to algorithms and discover various algorithm design techniques, before exploring how to implement different types of algorithms, such as searching and sorting, with the help of practical examples. As you advance to a more complex set of algorithms, you'll learn about linear programming, page ranking, and graphs, and even work with machine learning algorithms, understanding the math and logic behind them. Further on, case studies such as weather prediction, tweet clustering, and movie recommendation engines will show you how to apply these algorithms optimally. Finally, you'll become well versed in techniques that enable parallel processing, giving you the ability to use these algorithms for compute-intensive tasks. By the end of this book, you'll have become adept at solving real-world computational problems by using a wide range of algorithms. What you will learnExplore existing data structures and algorithms found in

Python librariesImplement graph algorithms for fraud detection using network analysisWork with machine learning algorithms to cluster similar tweets and process Twitter data in real timePredict the weather using supervised learning algorithmsUse neural networks for object detectionCreate a recommendation engine that suggests relevant movies to subscribersImplement foolproof security using symmetric and asymmetric encryption on Google Cloud Platform (GCP)Who this book is for This book is for programmers or developers who want to understand the use of algorithms for problem-solving and writing efficient code. Whether you are a beginner looking to learn the most commonly used algorithms in a clear and concise way or an experienced programmer looking to explore cutting-edge algorithms in data science, machine learning, and cryptography, you'll find this book useful. Although Python programming experience is a must, knowledge of data science will be helpful but not necessary.