Algorithms In A Nutshell A Practical Guide

Understand, Manage, and Prevent Algorithmic BiasComputer Science Programming Basics in RubyAlgorithms In A NutshellAlgorithms UnlockedC++ In a NutshellR in a NutshellAlgorithms in a NutshellFoundations of Data ScienceSequence Analysis in a Nutshell: A Guide to ToolsEssential AlgorithmsLinux Kernel in a NutshellAlgorithms in a NutshellLinux in a NutshellIntroduction To AlgorithmsC in a NutshellThe Black Box SocietyAn Introduction to the Analysis of AlgorithmsGraph AlgorithmsBandit AlgorithmsData AlgorithmsUnderstanding Machine LearningData Mining and Knowledge Discovery with Evolutionary AlgorithmsComputer ScienceAppleScript in a NutshellSparse ModelingThe Algorithm Design ManualA Practical Introduction to Data Structures and Algorithm AnalysisAlgorithms and ComplexityAlgorithms in a NutshellStatistics in a NutshellPython in a NutshellCodeless Data Structures and AlgorithmsReal-World AlgorithmsMASTERING ALGORITHMS WITH C. Avec une disquetteVB.NET Language in a NutshellASP in a NutshellComponent-Based Software EngineeringC in a NutshellAllergy in a NutshellImplementing Service Level Objectives

Understand, Manage, and Prevent Algorithmic Bias

Despite growing interest, basic information on methods and models for mathematically analyzing algorithms has rarely been directly accessible to practitioners, researchers, or students. An Introduction to the Analysis of Algorithms, Second Edition, organizes and presents that knowledge, fully introducing primary techniques and results in the field. Robert Sedgewick and the late Philippe Flajolet have drawn from both classical mathematics and computer science, integrating discrete mathematics, elementary real analysis, combinatorics, algorithms, and data structures. They emphasize the mathematics needed to support scientific studies that can serve as the basis for predicting algorithm performance and for comparing different algorithms on the basis of performance. Techniques covered in the first half of the book include recurrences, generating functions, asymptotics, and analytic combinatorics. Structures studied in the second half of the book include permutations, trees, strings, tries, and mappings. Numerous examples are included throughout to illustrate applications to the analysis of algorithms that are playing a critical role in the evolution of our modern computational infrastructure. Improvements and additions in this new edition include Upgraded figures and code An all-new chapter introducing analytic combinatorics Simplified derivations via analytic combinatorics throughout The book’s thorough, self-contained coverage will help readers appreciate the field’s challenges, prepare them for advanced results—covered in their monograph Analytic Combinatorics and in Donald Knuth’s The Art of Computer Programming books—and provide the background they need to keep abreast of new research. "[Sedgewick and Flajolet] are not only worldwide leaders of the field, they also are masters of exposition. I am sure that every serious computer scientist will find this book rewarding in many ways." —From the Foreword by Donald E. Knuth

Computer Science Programming Basics in Ruby

This newly expanded and updated second edition of the best-selling classic continues to take the "mystery" out of designing algorithms, and analyzing their efficacy and efficiency. Expanding on the first edition, the book now serves as the primary textbook of choice for algorithm design courses while maintaining its status as the premier practical reference guide to algorithms for programmers, researchers, and students. The reader-friendly Algorithm Design Manual provides straightforward access to combinatorial algorithms technology, stressing design over analysis. The first part, Techniques, provides accessible instruction on methods for designing and analyzing computer algorithms. The second part, Resources, is intended for browsing and reference, and comprises the catalog of algorithmic resources, implementations and an extensive bibliography. NEW to the second edition: • Doubles the tutorial material and exercises over the first edition • Provides full online support for lecturers, and a completely updated and improved website component with lecture slides, audio and video • Contains a unique catalog identifying the 75 algorithmic problems that arise most often in practice, leading the reader down the right path to solve them • Includes several NEW "war stories" relating experiences from real-world applications • Provides up-to-date links leading to the very best algorithm implementations available in C, C++, and Java

Algorithms In A Nutshell

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 Learn Understand tree data structures without delving into unnecessary details or going into too much theory Get 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.

Algorithms Unlocked

This book is an introductory textbook on the design and analysis of algorithms. The author uses a careful selection of a few topics to illustrate the tools for algorithm analysis. Recursive algorithms are illustrated by Quicksort, FFT, fast matrix multiplications, and others. Algorithms associated with the network flow problem are fundamental in many areas of graph connectivity, matching theory, etc. Algorithms in number theory are discussed with some applications to public key encryption. This second edition will differ from the present edition mainly in that solutions to most of the exercises will be included.

C++ In a Nutshell

This book provides an introduction to the mathematical and algorithmic foundations of data science, including machine learning, high-dimensional geometry, and analysis of large networks. Topics include the counterintuitive nature of data in high dimensions, important linear algebraic techniques such as singular value decomposition, the theory of random walks and Markov chains, the fundamentals of and important algorithms for machine learning, algorithms and analysis for clustering, probabilistic models for large networks, representation learning including topic modelling and non-negative matrix factorization, wavelets and compressed sensing. Important probabilistic techniques are developed including the law of large numbers, tail inequalities, analysis of random projections, generalization guarantees in machine learning, and moment methods for analysis of phase transitions in large random graphs. Additionally, important structural and complexity measures are discussed such as matrix norms and VC-dimension. This book is suitable for both undergraduate and graduate courses in the design and analysis of algorithms for data.

R in a Nutshell

Explains how Visual BASIC has been altered to work within the .NET framework and provides information about topics such as syntax, keyword operations, accepted arguments, and undocumented behaviors of VB.NET.

Algorithms in a Nutshell

ASP in a Nutshell provides the high-quality reference documentation that web application developers really need to create effective Active Server Pages. It focuses on how features are used in a real application and highlights little-known or undocumented features.This book also includes an overview of the interaction between the latest release of Internet Information Server (version 5) and ASP 3.0, with an introduction to the IIS object model and the objects it comprises. The examples shown in this section and throughout the book are illustrated in VBScript.The main components of this book are: Active Server Pages Introduction. Brief overview of the ASP application paradigm with examples in VBScript. Also included is an introduction to Microsoft's Internet Information Server 5.0, the IIS object model, and the objects that it comprises. Object Reference. Each object is discussed in the following manner: descriptions, properties, collections, methods, events, accessory files/required DLLs, and remarks, including real-world uses, tips and tricks, and author's experience (where applicable). The objects--Application, Response, Request, Server, Session, ObjectContext, and ASPError, as well as ASP Directives, Global.ASA, and Server-Side Includes--all follow this paradigm. Component Reference. This section follows the same paradigm found in Object Reference. The discussion covers all of the additional components included with IIS, such as ActiveX Data Objects, the Ad Rotator, the Browser capabilities component, the File System Object, and more. Appendixes. Gives examples in one or two objects and components using Perl, REXX, and Python in ASP. Like other books in the "In a Nutshell" series this book offers the facts, including critical background information, in a no-nonsense manner that users will refer to again and again. It is a detailed reference that enables even experienced web developers to advance their ASP applications to new levels.

Foundations of Data Science

Efficient algorithms are at the heart of all but the most trivial programs. 'Algorithms in a Nutshell' helps programmers select, analyze, and implement the right algorithms for their particular needs, providing just enough mathematics to let the reader understand and analyze algorithm performance.

Sequence Analysis in a Nutshell: A Guide to Tools

An extensively revised edition of a mathematically rigorous yet accessible introduction to algorithms.

Essential Algorithms

Contains an introduction to the operating system with detailed documentation on commands, utilities, programs, system configuration, and networking.

Linux Kernel in a Nutshell

On behalf of the Organizing Committee I am pleased to present the proceedings of the 2005 Symposium on Component-Based Software Engineering (CBSE). CBSE is concerned with the development of software-intensive systems from reusable parts (components), the development of reusable parts, and system maintenance and improvement by means of component replacement and c- tomization. CBSE 2005, “Software Components at Work,” was the eighth in a series of events that promote a science and technology foundation for achieving predictable quality in software systems through the use of software component technology and its associated software engineering practices. We were fortunate to have a dedicated Program Committee comprised of 30 internationally recognized researchers and industrial practitioners. We received 91 submissions andeach paper wasreviewedby at least three ProgramComm- tee members (four for papers with an author on the Program Committee). The entirereviewingprocesswassupportedbyCyberChairPro,theWeb-basedpaper submissionandreviewsystemdevelopedandsupportedbyRichardvandeStadt of Borbala Online Conference Services. After a two-day virtual Program C- mittee meeting, 21 submissions were accepted as long papers and 2 submissions were accepted as short papers.

Algorithms in a Nutshell

If you know basic high-school math, you can quickly learn and apply the core concepts of computer science with this concise, hands-on book. Led by a team of experts, you’ll quickly understand the difference between computer science and computer programming, and you’ll learn how algorithms help you solve computing problems. Each chapter builds on material introduced earlier in the book, so you can master one core building block before moving on to the next. You’ll explore fundamental topics such as loops, arrays, objects, and classes, using the easy-to-learn Ruby programming language. Then you’ll put everything together in the last chapter by programming a simple game of tic-tac-toe. Learn how to write algorithms to solve real-world problems Understand the basics of computer architecture Examine the basic tools of a programming language Explore sequential, conditional, and loop programming structures Understand how the array data structure organizes storage Use searching techniques and comparison-based sorting algorithms Learn about objects, including how to build your own Discover how objects can be created from other objects Manipulate files and use their data in your software

Linux in a Nutshell

This practical text contains fairly "traditional" coverage of data structures with a clear and complete use of algorithm analysis, and some emphasis on file processing techniques as relevant to modern programmers. It fully integrates OO programming with these topics, as part of the detailed presentation of OO programming itself.Chapter topics include lists, stacks, and queues; binary and general trees; graphs; file processing and external sorting; searching; indexing; and limits to computation.For programmers who need a good reference on data structures.

Introduction To Algorithms

If you are ready to dive into the MapReduce framework for processing large datasets, this practical book takes you step by step through the algorithms and tools you need to build distributed MapReduce applications with Apache Hadoop or Apache Spark. Each chapter provides a recipe for solving a massive computational problem, such as building a recommendation system. You'll learn how to implement the appropriate MapReduce solution with code that you can use in your projects. Dr. Mahmoud Parsian covers basic design patterns, optimization techniques, and data mining and machine learning solutions for problems in bioinformatics, genomics, statistics, and social network analysis. This book also includes an overview of MapReduce, Hadoop, and Spark. Topics include: Market basket analysis for a large set of transactions Data mining algorithms (K-means, KNN, and Naive Bayes) Using huge genomic data to sequence DNA and RNA Naive Bayes theorem and Markov chains for data and market prediction Recommendation algorithms and pairwise document similarity Linear regression, Cox regression, and Pearson correlation Allelic frequency and mining DNA Social network analysis (recommendation systems, counting triangles, sentiment analysis)

C in a Nutshell

Every day, corporations are connecting the dots about our personal behavior—silently scrutinizing clues left behind by our work habits and Internet use. But who connects the dots about what firms are doing with all this information? Frank Pasquale exposes how powerful interests abuse secrecy for profit and explains ways to rein them in.

The Black Box Society

This book is a comprehensive guide to the diagnosis and management of allergies. Beginning with an overview of allergy epidemiology, immunology and aerobiology, the next chapters cover different types of allergy – respiratory, gastrointestinal, dermatologic – and include a section on anaphylaxis. The following chapters explain allergy testing techniques as well as supportive investigations such as nasal endoscopy, bronchoscopy and spirometry. The book concludes with discussion on allergy management techniques including complementary medications, and provides appendices detailing algorithmic approaches and pharmacotherapy. Each chapter features explanatory diagrams and tables, international guidelines, and further reading suggestions. Key points Comprehensive guide to diagnosis and management of allergies Examines allergy testing techniques and supportive investigations Includes appendices detailing algorithmic approaches and pharmacotherapy Features numerous diagrams and tables, and further reading suggestions

An Introduction to the Analysis of Algorithms

A clear and concise introduction and reference for anyone new to the subject of statistics.

Graph Algorithms

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.

Bandit Algorithms

This book integrates two areas of computer science, namely data mining and evolutionary algorithms. Both these areas have become increasingly popular in the last few years, and their integration is currently an active research area. In general, data mining consists of extracting knowledge from data. The motivation for applying evolutionary algorithms to data mining is that evolutionary algorithms are robust search methods which perform a global search in the space of candidate solutions. This book emphasizes the importance of discovering comprehensible, interesting knowledge, which is potentially useful for intelligent decision making. The text explains both basic concepts and advanced topics

Data Algorithms

For anyone who has ever wondered how computers solve problems, an engagingly written guide for nonexperts to the basics of computer algorithms. Have you ever wondered how your GPS can find the fastest way to your destination, selecting one route from seemingly countless possibilities in mere seconds? How your credit card account number is protected when you make a purchase over the Internet? The answer is algorithms. And how do these mathematical formulations translate themselves into your GPS, your laptop, or your smart phone? This book offers an engagingly written guide to the basics of computer algorithms. In Algorithms Unlocked, Thomas Cormen—coauthor of the leading college textbook on the subject—provides a general explanation, with limited mathematics, of how algorithms enable computers to solve problems. Readers will learn what computer algorithms are, how to describe them, and how to evaluate them. They will discover simple ways to search for information in a computer; methods for rearranging information in a computer into a prescribed order (“sorting”); how to solve basic problems that can be modeled in a computer with a mathematical structure called a “graph” (useful for modeling road networks, dependencies among tasks, and financial relationships); how to solve problems that ask questions about strings of characters such as DNA structures; the basic principles behind cryptography; fundamentals of data compression; and even that there are some problems that no one has figured out how to solve on a computer in a reasonable amount of time.

Understanding Machine Learning

Demonstrates the programming language's strength as a Web development tool, covering syntax, data types, built-ins, the Python standard module library, and real world examples.

Data Mining and Knowledge Discovery with Evolutionary Algorithms

Creating robust software requires the use of efficient algorithms, but programmers seldom think about them until a problem occurs. This updated edition of 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

Computer Science

Why learn R? Because it's rapidly becoming the standard for developing statistical software. R in a Nutshell provides a quick and practical way to learn this increasingly popular open source language and environment. You'll not only learn how to program in R, but also how to find the right user-contributed R packages for statistical modeling, visualization, and bioinformatics. The author introduces you to the R environment, including the R graphical user interface and console, and takes you through the fundamentals of the object-oriented R language. Then, through a variety of practical examples from medicine, business, and sports, you'll learn how you can use this remarkable tool to solve your own data analysis problems. Understand the basics of the language, including the nature of R objects Learn how to write R functions and build your own packages Work with data through visualization, statistical analysis, and other methods Explore the wealth of packages contributed by the R community Become familiar with the lattice graphics package for high-level data visualization Learn about bioinformatics packages provided by Bioconductor "I am excited about this book. R in a Nutshell is a great introduction to R, as well as a comprehensive reference for using R in data analytics and visualization. Adler provides 'real world' examples, practical advice, and scripts, making it accessible to anyone working with data, not just professional statisticians."

AppleScript in a Nutshell

To-the-point, authoritative, no-nonsense solutions have always been a trademark of O'Reilly books. The In a Nutshell books have earned a solid reputation in the field as the well-thumbed references that sit beside the knowledgeable developer's keyboard. C++ in a Nutshell lives up to the In a Nutshell promise. C++ in a Nutshell is a lean, focused reference that offers practical examples for the most important, most often used, aspects of C++.C++ in a Nutshell packs an enormous amount of information on C++ (and the many libraries used with it) in an indispensable quick reference for those who live in a deadline-driven world and need the facts but not the frills.The book's language reference is organized first by topic, followed by an alphabetical reference to the language's keywords, complete with syntax summaries and pointers to the topic references. The library reference is organized by header file, and each library chapter and class declaration presents the classes and types in alphabetical order, for easy lookup. Cross-references link related methods, classes, and other key features. This is an ideal resource for students as well as professional programmers.When you're programming, you need answers to questions about language syntax or parameters required by library routines quickly. What, for example, is the C++ syntax to define an alias for a namespace? Just how do you create and use an iterator to work with the contents of a standard library container? C++ in a Nutshell is a concise desktop reference that answers these questions, putting the full power of this flexible, adaptable (but somewhat difficult to master) language at every C++ programmer's fingertips.

Sparse Modeling

Learning a language--any language--involves a process wherein you learn to rely less and less on instruction and more increasingly on the aspects of the language you've mastered. Whether you're learning French, Java, or C, at some point you'll set aside the tutorial and attempt to converse on your own. It's not necessary to know every subtle facet of French in order to speak it well, especially if there's a good dictionary available. Likewise, C programmers don't need to memorize every detail of C in order to write good programs. What they need instead is a reliable, comprehensive reference that they can keep nearby. C in a Nutshell is that reference. This long-awaited book is a complete reference to the C programming language and C runtime library. Its purpose is to serve as a convenient, reliable companion in your day-to-day work as a C programmer. C in a Nutshell covers virtually everything you need to program in C, describing all the elements of the language and illustrating their use with numerous examples. The book is divided into three distinct parts. The first part is a fast-paced description, reminiscent of the classic Kernighan & Ritchie text on which many C programmers cut their teeth. It focuses specifically on the C language and preprocessor directives, including extensions introduced to the ANSI standard in 1999. These topics and others are covered: Numeric constants Implicit and explicit type conversions Expressions and operators Functions Fixed-length and variable-length arrays Pointers Dynamic memory management Input and output The second part of the book is a comprehensive reference to the C runtime library; it includes an overview of the contents of the standard headers and a description of each standard library function. Part III provides the necessary knowledge of the C programmer's basic tools: the compiler, the make utility, and the debugger. The tools described here are those in the GNU software collection. C in a Nutshell is the perfect companion to K&R, and destined to be the most reached-for reference on your desk.

The Algorithm Design Manual

A comprehensive guide to understanding the language of C offers solutions for everyday programming tasks and provides all the necessary information to understand and use common programming techniques. Original. (Intermediate).

A Practical Introduction to Data Structures and Algorithm Analysis

Named a Notable Book in the 21st Annual Best of Computing list by the ACM! Robert Sedgewick and Kevin Wayne’s Computer Science: An Interdisciplinary Approach is the ideal modern introduction to computer science with Java programming for both students and professionals. Taking a broad, applications-based approach, Sedgewick and Wayne teach through important examples from science, mathematics, engineering, finance, and commercial computing. The book demystifies computation, explains its intellectual underpinnings, and covers the essential elements of programming and computational problem solving in today’s environments. The authors begin by introducing basic programming elements such as variables, conditionals, loops, arrays, and I/O. Next, they turn to functions, introducing key modular programming concepts, including components and reuse. They present a modern introduction to object-oriented programming, covering current programming paradigms and approaches to data abstraction. Building on this foundation, Sedgewick and Wayne widen their focus to the broader discipline of computer science. They introduce classical sorting and searching algorithms, fundamental data structures and their application, and scientific techniques for assessing an implementation’s performance. Using abstract models, readers learn to answer basic questions about computation, gaining insight for practical application. Finally, the authors show how machine architecture links the theory of computing to real computers, and to the field’s history and evolution. For each concept, the authors present all the information readers need to build confidence, together with examples that solve intriguing problems. Each chapter contains question-and-answer sections, self-study drills, and challenging problems that demand creative solutions. Companion web site (introcs.cs.princeton.edu/java) contains Extensive supplementary information, including suggested approaches to programming assignments, checklists, and FAQs Graphics and sound libraries Links to program code and test data Solutions to selected exercises Chapter summaries Detailed instructions for installing a Java programming environment Detailed problem sets and projects Companion 20-part series of video lectures is available at informit.com/title/9780134493831

Algorithms and Complexity

An introduction to algorithms for readers with no background in advanced mathematics or computer science, emphasizing examples and real-world problems. Algorithms are what we do in order not to have to do something. Algorithms consist of instructions to carry out tasks—usually dull, repetitive ones. Starting from simple building blocks, computer algorithms enable machines to recognize and produce speech, translate texts, categorize and summarize documents, describe images, and predict the weather. A task that would take hours can be completed in virtually no time by using a few lines of code in a modern scripting program. This book offers an introduction to algorithms through the real-world problems they solve. The algorithms are presented in pseudocode and can readily be implemented in a computer language. The book presents algorithms simply and accessibly, without overwhelming readers or insulting their intelligence. Readers should be comfortable with mathematical fundamentals and have a basic understanding of how computers work; all other necessary concepts are explained in the text. After presenting background in pseudocode conventions, basic terminology, and data structures, chapters cover compression, cryptography, graphs, searching and sorting, hashing, classification, strings, and chance. Each chapter describes real problems and then presents algorithms to solve them. Examples illustrate the wide range of applications, including shortest paths as a solution to paragraph line breaks, strongest paths in elections systems, hashes for song recognition, voting power Monte Carlo methods, and entropy for machine learning. Real-World Algorithms can be used by students in disciplines from economics to applied sciences. Computer science majors can read it before using a more technical text.

Algorithms in a Nutshell

A friendly introduction to the most useful algorithms written in simple, intuitive English The revised and updated second edition of Essential Algorithms, offers an accessible introduction to computer algorithms. The book contains a description of important classical algorithms and explains when each is appropriate. The author shows how to analyze algorithms in order to understand their behavior and teaches techniques that the can be used to create new algorithms to meet future needs. The text includes useful algorithms such as: methods for manipulating common data structures, advanced data structures, network algorithms, and numerical algorithms. It also offers a variety of general problem-solving techniques. In addition to describing algorithms and approaches, the author offers details on how to analyze the performance of algorithms. The book is filled with exercises that can be used to explore ways to modify the algorithms in order to apply them to new situations. This updated edition of Essential Algorithms: Contains explanations of algorithms in simple terms, rather than complicated math Steps through powerful algorithms that can be used to solve difficult programming problems Helps prepare for programming job interviews that typically include algorithmic questions Offers methods can be applied to any programming language Includes exercises and solutions useful to both professionals and students Provides code examples updated and written in Python and C# Essential Algorithms has been updated and revised and offers professionals and students a hands-on guide to analyzing algorithms as well as the techniques and applications. The book also includes a collection of questions that may appear in a job interview. The book’s website will include reference implementations in Python and C# (which can be easily applied to Java and C++).

Statistics in a Nutshell

The new edition of this classic O’Reilly reference provides clear, detailed explanations of every feature in the C language and runtime library, including multithreading, type-generic macros, and library functions that are new in the 2011 C standard (C11). If you want to understand the effects of an unfamiliar function, and how the standard library requires it to behave, you’ll find it here, along with a typical example. Ideal for experienced C and C++ programmers, this book also includes popular tools in the GNU software collection. You’ll learn how to build C programs with GNU Make, compile executable programs from C source code, and test and debug your programs with the GNU debugger. In three sections, this authoritative book covers: C language concepts and language elements, with separate chapters on types, statements, pointers, memory management, I/O, and more The C standard library, including an overview of standard headers and a detailed function reference Basic C programming tools in the GNU software collection, with instructions on how use them with the Eclipse IDE

Python in a Nutshell

Although service-level objectives (SLOs) continue to grow in importance, there’s a distinct lack of information about how to implement them. Practical advice that does exist usually assumes that your team already has the infrastructure, tooling, and culture in place. In this book, recognized SLO expert Alex Hidalgo explains how to build an SLO culture from the ground up. Ideal as a primer and daily reference for anyone creating both the culture and tooling necessary for SLO-based approaches to reliability, this guide provides detailed analysis of advanced SLO and service-level indicator (SLI) techniques. Armed with mathematical models and statistical knowledge to help you get the most out of an SLO-based approach, you'll learn how to build systems capable of measuring meaningful SLIs with buy-in across all departments of your organization. Define SLIs that meaningfully measure the reliability of a service from a user's perspective Choose appropriate SLO targets, including how to perform statistical and probabilistic analysis Use error budgets to help your team have better discussions and make better data-driven decisions Build supportive tooling and resources required for an SLO-based approach Use SLO data to present meaningful reports to leadership and your users

Codeless Data Structures and Algorithms

Presents an overview of kernel configuration and building for version 2.6 of the Linux kernel.

Real-World Algorithms

This work pulls together all of the vital information about the most commonly used databases, analytical tools, and tables used in sequence analysis.

MASTERING ALGORITHMS WITH C. Avec une disquette

Discover how graph algorithms can help you leverage the relationships within your data to develop more intelligent solutions and enhance your machine learning models. You'll learn how graph analytics are uniquely suited to unfold complex structures and reveal difficult-to-find patterns lurking in your data. Whether you are trying to build dynamic network models or forecast real-world behavior, this book illustrates how graph algorithms deliver value—from finding vulnerabilities and bottlenecks to detecting communities and improving machine learning predictions. This practical book walks you through hands-on examples of how to use graph algorithms in Apache Spark and Neo4j—two of the most common choices for graph analytics. Also included: sample code and tips for over 20 practical graph algorithms that cover optimal pathfinding, importance through centrality, and community detection. Learn how graph analytics vary from conventional statistical analysis Understand how classic graph algorithms work, and how they are applied Get guidance on which algorithms to use for different types of questions Explore algorithm examples with working code and sample datasets from Spark and Neo4j See how connected feature extraction can increase machine learning accuracy and precision Walk through creating an ML workflow for link prediction combining Neo4j and Spark

VB.NET Language in a Nutshell

Are algorithms friend or foe? The human mind is evolutionarily designed to take shortcuts in order to survive. We jump to conclusions because our brains want to keep us safe. A majority of our biases work in our favor, such as when we feel a car speeding in our direction is dangerous and we instantly move, or when we decide not take a bite of food that appears to have gone bad. However, inherent bias negatively affects work environments and the decision-making surrounding our communities. While the creation of algorithms and machine learning attempts to eliminate bias, they are, after all, created by human beings, and thus are susceptible to what we call algorithmic bias. In Understand, Manage, and Prevent Algorithmic Bias, author Tobias Baer helps you understand where algorithmic bias comes from, how to manage it as a business user or regulator, and how data science can prevent bias from entering statistical algorithms. Baer expertly addresses some of the 100+ varieties of natural bias such as confirmation bias, stability bias, pattern-recognition bias, and many others. Algorithmic bias mirrors—and originates in—these human tendencies. Baer dives into topics as diverse as anomaly detection, hybrid model structures, and self-improving machine learning. While most writings on algorithmic bias focus on the dangers, the core of this positive, fun book points toward a path where bias is kept at bay and even eliminated. You’ll come away with managerial techniques to develop unbiased algorithms, the ability to detect bias more quickly, and knowledge to create unbiased data. Understand, Manage, and Prevent Algorithmic Bias is an innovative, timely, and important book that belongs on your shelf. Whether you are a seasoned business executive, a data scientist, or simply an enthusiast, now is a crucial time to be educated about the impact of algorithmic bias on society and take an active role in fighting bias. What You'll Learn Study the many sources of algorithmic bias, including cognitive biases in the real world, biased data, and statistical artifact Understand the risks of algorithmic biases, how to detect them, and managerial techniques to prevent or manage them Appreciate how machine learning both introduces new sources of algorithmic bias and can be a part of a solution Be familiar with specific statistical techniques a data scientist can use to detect and overcome algorithmic bias Who This Book is For Business executives of companies using algorithms in daily operations; data scientists (from students to seasoned practitioners) developing algorithms; compliance officials concerned about algorithmic bias; politicians, journalists, and philosophers thinking about algorithmic bias in terms of its impact on society and possible regulatory responses; and consumers concerned about how they might be affected by algorithmic bias

ASP in a Nutshell

A comprehensive and rigorous introduction for graduate students and researchers, with applications in sequential decision-making problems.

Component-Based Software Engineering

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.

C in a Nutshell

Introduces machine learning and its algorithmic paradigms, explaining the principles behind automated learning approaches and the considerations underlying their usage.

Allergy in a Nutshell

AppleScript in a Nutshell is the first complete reference to AppleScript, the popular programming language that gives both power users and sophisticated enterprise customers the important ability to automate repetitive tasks and customize applications. As the Macintosh continues to expand and solidify its base in the multimedia and publishing industries, AppleScript is the tool of choice on this platform for creating sophisticated time- and money-saving workflow applications (applets). These applets automate the processing and management of digital video, imaging, print, and web-based material. AppleScript is also gaining a foothold in scientific programming, as technical organizations adopt G4 CPU-based systems for advanced computing and scientific analysis. Finally, "power users" and script novices will find that AppleScript is a great everyday Mac programming tool, similar to Perl on Windows NT or Unix.In this well-organized and concise reference, AppleScript programmers will find: Detailed coverage of AppleScript Version 1.4 and beyond on Mac OS 9 and Mac OS X. Complete descriptions of AppleScript language features, such as data types, flow-control statements, functions, object-oriented features (script objects and libraries), and other syntactical elements. Descriptions and hundreds of code samples on programming the various "scriptable" system components, such as the Finder, File Sharing, File Exchange, Network scripting, Web scripting, Apple System Profiler, the ColorSync program, and the numerous powerful language extensions called "osax" or scripting additions.Most other AppleScript books are hopelessly out of date. AppleScript in a

Nutshell covers the latest updates and improvements with practical, easy to understand tips, including: Using AppleScript as a tool for distributed computing, an exciting development that Apple Computer calls "program linking over IP." Programmers can now do distributed computing with Macs over TCP/IP networks, including controlling remote applications with AppleScript and calling AppleScript methods on code libraries that are located on other machines. Using the Sherlock find application to automate web and network searching. Insights on scripting new Apple technologies such as Apple Data Detectors, Folder Actions, Keychain Access, and Apple Verifier.AppleScript in a Nutshell is a high-end handbook at a low-end price--an essential desktop reference that puts the full power of this user-friendly programming language into every AppleScript user's hands.

Implementing Service Level Objectives

Sparse models are particularly useful in scientific applications, such as biomarker discovery in genetic or neuroimaging data, where the interpretability of a predictive model is essential. Sparsity can also dramatically improve the cost efficiency of signal processing. Sparse Modeling: Theory, Algorithms, and Applications provides an introduction to the growing field of sparse modeling, including application examples, problem formulations that yield sparse solutions, algorithms for finding such solutions, and recent theoretical results on sparse recovery. The book gets you up to speed on the latest sparsity-related developments and will motivate you to continue learning about the field. The authors first present motivating examples and a high-level survey of key recent developments in sparse modeling. The book then describes optimization problems involving commonly used sparsity-enforcing tools, presents essential theoretical results, and discusses several state-of-the-art algorithms for finding sparse solutions. The authors go on to address a variety of sparse recovery problems that extend the basic formulation to more sophisticated forms of structured sparsity and to different loss functions. They also examine a particular class of sparse graphical models and cover dictionary learning and sparse matrix factorizations.

ROMANCE ACTION & ADVENTURE MYSTERY & THRILLER BIOGRAPHIES & HISTORY CHILDREN'S YOUNG ADULT FANTASY HISTORICAL FICTION HORROR LITERARY FICTION NON-FICTION SCIENCE FICTION