Programming Pearls 2nd Edition

Yeah, reviewing a ebook Programming Pearls 2nd Edition could mount up your close associates listings. This is just one of the solutions for you to be successful. As understood, completion does not recommend that you have astonishing points.

Comprehending as skillfully as harmony even more than other will come up with the money for each success. neighboring to, the revelation as without difficulty as perception of this Programming Pearls 2nd Edition can be taken as without difficulty as picked to act.

Introductory Programming with Simple Games Brian C. Ladd 2010-03-29 This is an excellent resource for programmers who need to learn Java but aren't interested in just reading about concepts. Introduction to Java Programming with Games follows a spiral approach to introduce concepts and enable them to write game programs as soon as they start. It includes code examples and problems that are easy to understand and motivates them to work through to find the solutions. This game-motivated presentation will help programmers quickly apply what they've learned in order to build their skills.
Inleiding informatica J. Glenn Brookshear 2005
Algorithm Engineering Stefan Näher 2007-06-03 This volume contains the papers accepted for the 4th Workshop on Algorithm Engineering (WAE 2000) held in Saarbruc ? ken, Germany, during 5–8 September 2000, together with the abstract of the invited lecture given by Karsten Weihe. The Workshop on Algorithm Engineering covers research on all aspects of the subject. The goal is to present recent research results and to identify and explore directions for future research. Previous meetings were held in Venice (1997), Saarbruc ? ken (1998), and London (1999). Papers were solicited describing original research in all aspects of algorithm engineering, including: – Development of software repositories and platforms which allow the use of and experimentation with e?cient discrete algorithms. – Novel uses of discrete algorithms in other disciplines and the evaluation of algorithms for realistic environments. – Methodological issues including standards in the context of empirical - search on algorithms and data structures. – Methodological issues regarding the process of converting user requirements into e?cient algorithmic solutions and implementations. The program committee accepted 16 from a total of 30 submissions. The program committee meeting was conducted electronically.

The criteria for sel- tion were originality, quality, and relevance to the subject area of the workshop. Considerable e?ort was devoted to the evaluation of the submissions and to p- viding the authors with feedback. Each submission was reviewed by at least four program committee members (assisted by subreferees). A special issue of the ACM Journal of Experimental Algorithmics will be devoted to selected papers from WAE 2000.

Data Analysis with Open Source Tools Philipp K. Janert 2010-11-11 Collecting data is relatively easy, but turning raw information into something useful requires that you know how to extract precisely what you need. With this insightful book, intermediate to experienced programmers interested in data analysis will learn techniques for working with data in a business environment. You'll learn how to look at data to discover what it contains, how to capture those ideas in conceptual models, and then feed your understanding back into the organization through business plans, metrics dashboards, and other applications. Along the way, you'll experiment with concepts through hands-on workshops at the end of each chapter. Above all, you'll learn how to think about the results you want to achieve -- rather than rely on tools to think for you. Use graphics to describe data with one, two, or dozens of variables Develop conceptual models using back-of-the-envelope calculations, as well asscaling and probability arguments Mine data with computationally intensive methods such as simulation and clustering Make your conclusions understandable through reports, dashboards, and other metrics programs Understand financial calculations, including the time-value of money Use dimensionality reduction techniques or predictive analytics to conquer challenging data analysis situations Become familiar with different open source programming environments for data analysis "Finally, a concise reference for understanding how to conquer piles of data."--Austin King, Senior Web Developer, Mozilla "An indispensable text for aspiring data scientists."--Michael E. Driscoll, CEO/Founder, Dataspora

Quill & Quire 2001

Algorithms -- ESA 2004 Susanne Albers 2004-09 This book constitutes the refereed proceedings of the 12th Annual European Symposium on Algorithms, ESA 2004, held in Bergen, Norway, in September 2004. The 70 revised full papers presented were carefully reviewed from 208 submissions. The scope of the papers spans the entire range of algorithmics from design and mathematical issues to real-world applications in various fields, and engineering and analysis of algorithms.

Programming Pearls, 2nd Edition ?????(???) Jon Bentley 2019-10-30 "?Programming
Pearls???Jon??????????????????
Steve McConnell??Code Complete??? ?????????????????Jon
Bentley?????????????????????????(pearls)??????????????????????????????????????
Pearls??????????????
14????????Bentley??(1)?????????(2

Algorithm Engineering Gerd Stoelting Brodal 2003-06-30 This book constitutes the refereed proceedings of the 5th Workshop on Algorithm Engineering, WAE 2001, held in Aarhus, Denmark, in August 2001. The 15 revised full papers presented were carefully reviewed and selected from 25 submissions. Among the topics addressed are implementation, experimental testing, and fine-tuning of discrete algorithms; novel use of discrete algorithms in other disciplines; empirical research on algorithms and data structures; and methodological issues regarding the process of converting user requirements into efficient algorithmic solutions and implemenations.

Algorithm Engineering Denmark) Wae 200 (2001 Aarhus 2001-08-15 Compact DFA representation for fast regular expression search / Gonzalo Navarro / - The Max-Shift algorithm for approximate string matching / Costas S. Iliopoulos / - Fractal matrix multiplication : a case study on portability of cache performance / Gianfranco Bilardi / - Experiences with the design and implementation of space-efficient deques / Jyrki Katajainen / - Designing and implementing a general purpose halfedge data structure / Hervé Brönnimann / - Optimised predecessor data structures for internal memory / Naila Rahman / - An adaptable and expensible geometry kernel / Susan Hert / - Efficient resource allocation with noisy functions / Arne Andersson / - Improving the efficiency of branch and bound algorithms for the simple plant location problem / Boris Goldengorin / - Exploiting partial knowledge of satisfying assignments / Kazuo Iwama / - Using PRAM algorithms on a uniform-memory-access shared-memory architecture / David A. Bader / - An experimental study of basic communicat ...

Algorithms and Complexity Giancarlo Bongiovanni 1997-02-26 This book constitutes the refereed proceedings of the Third Italian Conference on Algorithms and Complexity, CIAC'97, held in Rome, Italy in March 1997. The 25 revised full papers included in the volume were carefully selected from a total of 74 submissions; also included is an invited paper and an invited abstract. All in all, the papers present an interesting snapshot of current research activities and recent results in theory and applications of sequential, distributed, and parallel algorithms, data structures, and computational complexity.

Algorithm Engineering 2001

Puzzles for Hackers Ivan Sklyarov 2005 These puzzles and mind-benders serve as a way to train logic and help developers, hackers, and system administrators discover unconventional solutions to common IT problems. Users will learn to find bugs in source code, write exploits, and solve nonstandard coding tasks and hacker puzzles. Cryptographic puzzles, puzzles for Linux and Windows hackers, coding puzzles, and puzzles for web designers are included.

Formules en Functies in Excel voor Dummies Ken Bluttman 2011

Programming Pearls Jon Louis Bentley 1986 Software -- Software Engineering.

Algorithms and Computation Rudolf Fleischer 2004-12-06 This volume contains

the proceedings of the 15th Annual International Sym- sium on Algorithms and Computation (ISAAC 2004), held in Hong Kong, 20–22 December, 2004. In the past, it has been held in Tokyo (1990), Taipei (1991), Nagoya (1992), Hong Kong (1993), Beijing (1994), Cairns (1995), Osaka (1996), Singapore (1997), Taejon (1998), Chennai (1999), Taipei (2000), Christchurch (2001), Vancouver (2002), and Kyoto (2003). ISAAC is an annual international symposium that covers a wide range of topics,namelyalgorithmsandcomputation.Themainpurposeofthesymposium is to provide a forum for researchers working in the active research community of algorithms and the theory of computation to present and exchange new ideas. In response to our call for papers we received 226 submissions. The task of selectingthepapersinthisvolumewasdonebyourprogramcommitteeandother referees. After a thorough review process the committee selected 76 papers, the decisions being based on originality and relevance to the ?eld of algorithms and computation. We hope all accepted papers will eventually appear in scienti?c journals in a more polished form. Two special issues, one of Algorithmica and one of the International Journal of Computational Geometry and Applications, with selected papers from ISAAC 2004 are in preparation. Thebeststudentpaperawardwillbegivenfor"Geometricoptimizationpr- lems over sliding windows" by Bashir S. Sadjad and Timothy M. Chan from the University of Waterloo. Two eminent invited speakers, Prof. Erik D. Demaine, MIT, and Prof. David M. Mount, University of Maryland, also contributed to this volume. Ben jij slim genoeg om voor Google te werken? William Poundstone 2012-06-06 o Je krimpt tot een grootte van een dubbeltje en wordt in een blender gegooid. De blender zal binnen een minuut worden aangezet. Wat doet je? o Je zet een glas met water op een grammofoon en geleidelijk begint die steeds sneller te draaien. Wat gebeurt er als eerst: het glas glijdt ervan af, het glas valt om of het water valt eruit? o Je krijgt een blok kaas en een mes. Hoeveel keer moet je de kaas snijden, zodat je 27 kleine, gelijke blokjes hebt? o Hoeveel hele getallen tussen de 1 en 1.000 bevatten een 3? o Op een verlaten snelweg is de kans dat er een auto langskomt binnen 30 minuten 95%. Hoe hoog is de kans dat er een auto langskomt binnen 10 minuten? o Leg aan je neefje van acht jaar uit wat een database is en gebruik hiervoor 3 zinnen. Dit zijn vragen die je gesteld kunnen worden als je bij Google - of een willekeurig ander bedrijf in de dotcom economy - solliciteert. Ben jij slim genoeg om bij Google te werken? loodst je langs verrassende antwoorden op tientallen van de meest uitdagende vragen die op je afgevuurd kunnen worden tijdens een sollicitatiegesprek. Effective Awk Programming Arnold Robbins 2015-03-03 When processing text files, the awk language is ideal for handling data extraction, reporting, and data-reformatting jobs. This practical guide serves as both a reference and tutorial for POSIX-standard awk and for the GNU implementation, called gawk. This book is useful for novices and awk experts alike. In this thoroughly revised edition, author and gawk lead developer Arnold Robbins describes the awk language

and gawk program in detail, shows you how to use awk and gawk for problem solving, and then dives into specific features of gawk. System administrators, programmers, webmasters, and other power users will find everything they need to know about awk and gawk. You will learn how to: Format text and use regular expressions in awk and gawk Process data using awk's operators and built-in functions Manage data relationships using associative arrays Define your own functions "Think in awk" with two full chapters of sample functions and programs Take advantage of gawk's many advanced features Debug awk programs with the gawk built-in debugger Extend gawk by writing new functions in C or C++ This book is published under the terms of the GNU Free Documentation License. You have the freedom to copy and modify this GNU manual. Royalties from the sales of this book go to the Free Software Foundation and to the author. De zon en de maan en de Rolling Stones Rich Cohen 2017-04-27 Rich Cohen ging in de jaren negentig als jonge verslaggever voor muziekblad Rolling Stone mee op tournee met de Rolling Stones. Hij viel als een blok voor hun humor en camaraderie, het onderlinge geplaag, het harde leven. En dat is zo gebleven. Voor Cohen zijn de Rolling Stones de grootste rock-'n-roll band aller tijden. Het verhaal begint bij het prille begin, als Mick en Keith elkaar in 1962 ontmoeten op een perron, en volgt de band langs de vele hoogte- en dieptepunten. Cohen staat stil bij de momenten die niet alleen aantonen dat de Stones getalenteerde muzikanten en de meest innovatieve songwriters van hun generatie zijn, maar ook de iconen van onze moderne cultuur. Want uiteindelijk, na de drugs, de vriendinnen, de ruzies en de eindeloze reünies, is er de muziek. Na De zon en de maan en de Rolling Stones wil je alle oude nummers opnieuw beluisteren, én de obscure pareltjes die je nog nooit hebt gehoord. De muziek zal, samen met Cohens frisse en energieke benadering van de band, voor eens en voor altijd duidelijk maken waarom de Stones er altijd toe blijven doen. Rich Cohen is journalist en schrijft onder meer voor Vanity Fair, The New Yorker en Rolling Stone. Daarnaast publiceerde hij tien boeken, waaronder Tough Jews en Sweet and Low. Ook schreef hij het script voor de HBO-serie Vinyl. Cohen ontving meerdere prijzen voor zijn werk.
Pearls of Functional Algorithm Design Richard Bird 2010-09-16 Richard Bird takes a radical approach to algorithm design, namely, design by calculation. These 30 short chapters each deal with a particular programming problem drawn from sources as diverse as games and puzzles, intriguing combinatorial tasks, and more familiar areas such as data compression and string matching. Each pearl starts with the statement of the problem expressed using the functional programming language Haskell, a powerful yet succinct language for capturing algorithmic ideas clearly and simply. The novel aspect of the book is that each solution is calculated from an initial formulation of the problem in Haskell by appealing to the laws of functional programming. Pearls of Functional Algorithm Design will appeal to the aspiring functional programmer, students and teachers interested in the principles of algorithm design, and anyone seeking to master

the techniques of reasoning about programs in an equational style.

C++ Coding Standards Herb Sutter 2004-10-25 Consistent, high-quality coding standards improve software quality, reduce time-to-market, promote teamwork, eliminate time wasted on inconsequential matters, and simplify maintenance. Now, two of the world's most respected C++ experts distill the rich collective experience of the global C++ community into a set of coding standards that every developer and development team can understand and use as a basis for their own coding standards. The authors cover virtually every facet of C++ programming: design and coding style, functions, operators, class design, inheritance, construction/destruction, copying, assignment, namespaces, modules, templates, genericity, exceptions, STL containers and algorithms, and more. Each standard is described concisely, with practical examples. From type definition to error handling, this book presents C++ best practices, including some that have only recently been identified and standardized-techniques you may not know even if you've used C++ for years. Along the way, you'll find answers to questions like What's worth standardizing--and what isn't? What are the best ways to code for scalability? What are the elements of a rational error handling policy? How (and why) do you avoid unnecessary initialization, cyclic, and definitional dependencies? When (and how) should you use static and dynamic polymorphism together? How do you practice "safe" overriding? When should you provide a no-fail swap? Why and how should you prevent exceptions from propagating across module boundaries? Why shouldn't you write namespace declarations or directives in a header file? Why should you use STL vector and string instead of arrays? How do you choose the right STL search or sort algorithm? What rules should you follow to ensure type-safe code? Whether you're working alone or with others, C++ Coding Standards will help you write cleaner code--and write it faster, with fewer hassles and less frustration.

Hacking Jon Mark Erickson 2004

Software Development, Design and Coding John F. Dooley 2017-11-25 Learn the principles of good software design, and how to turn those principles into great code. This book introduces you to software engineering — from the application of engineering principles to the development of software. You'll see how to run a software development project, examine the different phases of a project, and learn how to design and implement programs that solve specific problems. It's also about code construction — how to write great programs and make them work. Whether you're new to programming or have written hundreds of applications, in this book you'll re-examine what you already do, and you'll investigate ways to improve. Using the Java language, you'll look deeply into coding standards, debugging, unit testing, modularity, and other characteristics of good programs. With Software Development, Design and Coding, author and professor John Dooley distills his years of teaching and development experience to demonstrate practical techniques for great coding. What You'll Learn Review modern agile methodologies including Scrum and Lean programming Leverage

the capabilities of modern computer systems with parallel programming Work with design patterns to exploit application development best practices Use modern tools for development, collaboration, and source code controls Who This Book Is For Early career software developers, or upper-level students in software engineering courses

Algorithms on Trees and Graphs Gabriel Valiente 2013-04-17 Graph algorithms is a well-established subject in mathematics and computer science. Beyond classical application fields, such as approximation, combinatorial optimization, graphics, and operations research, graph algorithms have recently attracted increased attention from computational molecular biology and computational chemistry. Centered around the fundamental issue of graph isomorphism, this text goes beyond classical graph problems of shortest paths, spanning trees, flows in networks, and matchings in bipartite graphs. Advanced algorithmic results and techniques of practical relevance are presented in a coherent and consolidated way. This book introduces graph algorithms on an intuitive basis followed by a detailed exposition in a literate programming style, with correctness proofs as well as worst-case analyses. Furthermore, full C++ implementations of all algorithms presented are given using the LEDA library of efficient data structures and algorithms.

Het huisje dat verhuisde Virginia Lee Burton 2002 Een huisje op het platteland komt langzamerhand in het centrum van een grote stad vol stof en rook te liggen. Gelukkig kan het huisje naar buiten verhuizen waar het de wisseling der seizoenen weer kan beleven. Prentenboek met illustraties in kleur. Vanaf ca. 4 jaar.

EBOOK: OBJECT-ORIENTED SOFTWAR LETHBRIDGE 2004-12-16 EBOOK: OBJECT-ORIENTED SOFTWAR

Algoritmen en datastructuren Niklaus Wirth 1989 Inleiding in het programmeren, bestemd voor programmeurs.

Algorithmic Puzzles Anany Levitin 2011-10-12 While many think of algorithms as specific to computer science, at its core algorithmic thinking is defined by the use of analytical logic to solve problems. This logic extends far beyond the realm of computer science and into the wide and entertaining world of puzzles. In Algorithmic Puzzles, Anany and Maria Levitin use many classic brainteasers as well as newer examples from job interviews with major corporations to show readers how to apply analytical thinking to solve puzzles requiring well-defined procedures. The book's unique collection of puzzles is supplemented with carefully developed tutorials on algorithm design strategies and analysis techniques intended to walk the reader step-by-step through the various approaches to algorithmic problem solving. Mastery of these strategies-- exhaustive search, backtracking, and divide-and-conquer, among others--will aid the reader in solving not only the puzzles contained in this book, but also others encountered in interviews, puzzle collections, and throughout everyday life. Each of the 150 puzzles contains hints and solutions, along with commentary on the

puzzle's origins and solution methods. The only book of its kind, Algorithmic Puzzles houses puzzles for all skill levels. Readers with only middle school mathematics will develop their algorithmic problem-solving skills through puzzles at the elementary level, while seasoned puzzle solvers will enjoy the challenge of thinking through more difficult puzzles.

The Art of Readable Code Dustin Boswell 2011-11-03 Chapter 5. Knowing What to Comment; What NOT to Comment; Don't Comment Just for the Sake of Commenting; Don't Comment Bad Names--Fix the Names Instead; Recording Your Thoughts; Include "Director Commentary"; Comment the Flaws in Your Code; Comment on Your Constants; Put Yourself in the Reader's Shoes; Anticipating Likely Questions; Advertising Likely Pitfalls; "Big Picture" Comments; Summary Comments; Final Thoughts--Getting Over Writer's Block; Summary; Chapter 6. Making Comments Precise and Compact; Keep Comments Compact; Avoid Ambiguous Pronouns; Polish Sloppy Sentences.

Programming Pearls Jon Louis Bentley 2000 A guide to practical programming techniques and design principles, with information on such topics as testing, debugging and timing, set representations, and string problems.

Hello, Startup Yevgeniy Brikman 2015-10-21 This book is the "Hello, World" tutorial for building products, technologies, and teams in a startup environment. It's based on the experiences of the author, Yevgeniy (Jim) Brikman, as well as interviews with programmers from some of the most successful startups of the last decade, including Google, Facebook, LinkedIn, Twitter, GitHub, Stripe, Instagram, AdMob, Pinterest, and many others. Hello, Startup is a practical, how-to guide that consists of three parts: Products, Technologies, and Teams. Although at its core, this is a book for programmers, by programmers, only Part II (Technologies) is significantly technical, while the rest should be accessible to technical and non-technical audiences alike. If you're at all interested in startups—whether you're a programmer at the beginning of your career, a seasoned developer bored with large company politics, or a manager looking to motivate your engineers—this book is for you.

How to Write Good Programs Perdita Stevens 2020-07-23 Learning to program isn't just learning the details of a programming language: to become a good programmer you have to become expert at debugging, testing, writing clear code and generally unsticking yourself when you get stuck, while to do well in a programming course you have to learn to score highly in coursework and exams. Featuring tips, stories and explanations of key terms, this book teaches these skills explicitly. Examples in Python, Java and Haskell are included, helping you to gain transferable programming skills whichever language you are learning. Intended for students in Higher or Further Education studying early programming courses, it will help you succeed in, and get the most out of, your course, and support you in developing the software engineering habits that lead to good programs.

The Art of UNIX Programming Eric S. Raymond 2003-09-23 The Art of UNIX

Programming poses the belief that understanding the unwritten UNIX engineering tradition and mastering its design patterns will help programmers of all stripes to become better programmers. This book attempts to capture the engineering wisdom and design philosophy of the UNIX, Linux, and Open Source software development community as it has evolved over the past three decades, and as it is applied today by the most experienced programmers. Eric Raymond offers the next generation of "hackers" the unique opportunity to learn the connection between UNIX philosophy and practice through careful case studies of the very best UNIX/Linux programs.

R Programming for Bioinformatics Robert Gentleman 2008-07-14 Due to its data handling and modeling capabilities as well as its flexibility, R is becoming the most widely used software in bioinformatics. R Programming for Bioinformatics explores the programming skills needed to use this software tool for the solution of bioinformatics and computational biology problems. Drawing on the author's first-hand experiences as an expert in R, the book begins with coverage on the general properties of the R language, several unique programming aspects of R, and object-oriented programming in R. It presents methods for data input and output as well as database interactions. The author also examines different facets of string handling and manipulations, discusses the interfacing of R with other languages, and describes how to write software packages. He concludes with a discussion on the debugging and profiling of R code. With numerous examples and exercises, this practical guide focuses on developing R programming skills in order to tackle problems encountered in bioinformatics and computational biology.

Algoritmes aan de macht Hannah Fry 2018-12-11 Stel, je staat terecht. Wie laat je liever beslissen over je lot: een foutgevoelige want menselijke rechter of een algoritme zonder enige empathie? Stel, je koopt een zelfrijdende auto. Wil je dat die zo veel mogelijk levens redt bij een botsing, of dat hij de eigen inzittenden bevoordeelt? Stel, een nieuwe machine heeft je medische gegevens nodig om kankerpatiënten te redden. Geef je je privacy op voor het algemeen belang? Algoritmes spelen een steeds grotere rol in ons leven. Op wat voor manier precies? En is het wel verstandig om belangrijke beslissingen zo klakkeloos aan ze uit te besteden? Wiskundige Hannah Fry gidst ons langs de dilemma's van ons nieuwe, geautomatiseerde bestaan.

A Concise Introduction to Data Structures using Java Mark J. Johnson 2013-11-18 A student-friendly text, A Concise Introduction to Data Structures Using Java takes a developmental approach, starting with simpler concepts first and then building toward greater complexity. Important topics, such as linked lists, are introduced gradually and revisited with increasing depth. More code and guidance are provided at the beginning, allowing students time to adapt to Java while also beginning to learn data structures. As students develop fluency in Java, less code is provided and more algorithms are outlined in pseudocode. The text is designed to support a second course in computer science with an

emphasis on elementary data structures. The clear, concise explanations encourage students to read and engage with the material, while partial implementations of most data structures give instructors the flexibility to develop some methods as examples and assign others as exercises. The book also supplies an introductory chapter on Java basics that allows students who are unfamiliar with Java to quickly get up to speed. The book helps students become familiar with how to use, design, implement, and analyze data structures, an important step on the path to becoming skilled software developers.

Oakeshott on Rome and America Gene Callahan 2012-07-04 The political systems of the Roman Republic were based almost entirely on tradition, "the way of the ancestors", rather than on a written constitution. While the founders of the American Republic looked to ancient Rome as a primary model for their enterprise, nevertheless, in line with the rationalist spirit of their age, the American founders attempted to create a rational set of rules that would guide the conduct of American politics, namely, the US Constitution. These two examples offer a striking case of the ideal types, famously delineated by Michael Oakeshott in "Rationalism in Politics" and elsewhere, between politics as a practice grounded in tradition and politics as a system based on principles flowing from abstract reasoning. This book explores how the histories of the two republics can help us to understand Oakeshott's claims about rational versus traditional politics. Through examining such issues we may come to understand better not only Oakeshott's critique of rationalism, but also modern constitutional theory, issues in the design of the European Union, and aspects of the revival of republicanism.

Real-World Algorithms Panos Louridas 2017-03-17 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.

Software Development and Professional Practice John Dooley 2011-10-13 Software Development and Professional Practice reveals how to design and code great software. What factors do you take into account? What makes a good design? What methods and processes are out there for designing software? Is designing small programs different than designing large ones? How can you tell a good design from a bad one? You'll learn the principles of good software design, and how to turn those principles back into great code. Software Development and Professional Practice is also about code construction—how to write great programs and make them work. What, you say? You've already written eight gazillion programs! Of course I know how to write code! Well, in this book you'll re-examine what you already do, and you'll investigate ways to improve. Using the Java language, you'll look deeply into coding standards, debugging, unit testing, modularity, and other characteristics of good programs. You'll also talk about reading code. How do you read code? What makes a program readable? Can good, readable code replace documentation? How much documentation do you really need? This book introduces you to software engineering—the application of engineering principles to the development of software. What are these engineering principles? First, all engineering efforts follow a defined process. So, you'll be spending a bit of time talking about how you run a software development project and the different phases of a project. Secondly, all engineering work has a basis in the application of science and mathematics to real-world problems. And so does software development! You'll therefore take the time to examine how to design and implement programs that solve specific problems. Finally, this book is also about human-computer interaction and user interface design issues. A poor user interface can ruin any desire to actually use a program; in this book, you'll figure out why and how to avoid those errors. Software Development and Professional Practice covers many of the topics described for the ACM Computing Curricula 2001 course C292c Software Development and Professional Practice. It is designed to be both a textbook and a manual for the working professional.

Understanding the Discrete Element Method Hans-Georg Matuttis 2014-05-12 Gives readers a more thorough understanding of DEM and equips researchers for independent work and an ability to judge methods related to simulation of polygonal particles Introduces DEM from the fundamental concepts (theoretical mechanics and solidstate physics), with 2D and 3D simulation methods for polygonal particles Provides the fundamentals of coding discrete element method (DEM) requiring little advance knowledge of granular matter or

numerical simulation Highlights the numerical tricks and pitfalls that are usually only realized after years of experience, with relevant simple experiments as applications Presents a logical approach starting withthe mechanical and physical bases,followed by a description of the techniques and finally their applications Written by a key author presenting ideas on how to model the dynamics of angular particles using polygons and polyhedral Accompanying website includes MATLAB-Programs providing the simulation code for two-dimensional polygons Recommended for researchers and graduate students who deal with particle models in areas such as fluid dynamics, multi-body engineering, finite-element methods, the geosciences, and multi-scale physics. Functional Python Programming Steven F. Lott 2018-04-13 Create succinct and expressive implementations with functional programming in Python Key Features Learn how to choose between imperative and functional approaches based on expressiveness, clarity, and performance Get familiar with complex concepts such as monads, concurrency, and immutability Apply functional Python to common Exploratory Data Analysis (EDA) programming problems Book Description If you're a Python developer who wants to discover how to take the power of functional programming (FP) and bring it into your own programs, then this book is essential for you, even if you know next to nothing about the paradigm. Starting with a general overview of functional concepts, you'll explore common functional features such as first-class and higher-order functions, pure functions, and more. You'll see how these are accomplished in Python 3.6 to give you the core foundations you'll build upon. After that, you'll discover common functional optimizations for Python to help your apps reach even higher speeds. You'll learn FP concepts such as lazy evaluation using Python's generator functions and expressions. Moving forward, you'll learn to design and implement decorators to create composite functions. You'll also explore data preparation techniques and data exploration in depth, and see how the Python standard library fits the functional programming model. Finally, to top off your journey into the world of functional Python, you'll at look at the PyMonad project and some larger examples to put everything into perspective. What you will learn Use Python's generator functions and generator expressions to work with collections in a non-strict (or lazy) manner Utilize Python library modules including itertools, functools, multiprocessing, and concurrent features to ensure efficient functional programs Use Python strings with object-oriented suffix notation and prefix notation Avoid stateful classes with families of tuples Design and implement decorators to create composite functions Use functions such as max(), min(), map(), filter(), and sorted() Write higher-order functions Who this book is for This book is for Python developers who would like to perform Functional programming with Python. Python Programming knowledge is assumed.