Read Online Design And Analysis Of Algorithms Aho Ullman Free Download Pdf

An Introduction to the Analysis of Algorithms **An Introduction to the Analysis of Algorithms Analysis of Algorithms** *Beyond the Worst-Case Analysis of Algorithms* Design and Analysis of Algorithms **Design and Analysis of Algorithms Mathematics for the Analysis of Algorithms The Design and Analysis of Algorithms** Design and Analysis of Algorithms *An Introduction to the Analysis of Algorithms Practical Analysis of Algorithms Design and Analysis of Algorithms The Design and Analysis of Computer Algorithms* **Average Case Analysis of Algorithms on Sequences** Analytic Combinatorics *Introduction To Design And Analysis Of Algorithms, 2/E* **Design and Analysis of Algorithms** *Analysis and Algorithms for Service Parts Supply Chains* **The Algorithm Design Manual Introduction to the Design and Analysis of Algorithms** Analysis and Design of Algorithms **Analysis of Algorithms** Compared to What? **Data Structures and Network Algorithms** Introduction To The Analysis Of Algorithms, An (3rd Edition) Design and Analysis of Algorithms **Probabilistic Analysis of Algorithms** *Computer Algorithms: Design, Analysis and Applications* **Data Structures and Algorithm Analysis in C** DESIGN AND ANALYSIS OF ALGORITHMS **Concentration of Measure for the Analysis of Randomized Algorithms Gabor Analysis and Algorithms** Techniques for Designing and Analyzing Algorithms **Design and Analysis of Distributed Algorithms** Advances in Computational Algorithms and Data Analysis Numerical Analysis with Algorithms and Programming Algorithms and Models for Network Data and Link Analysis Data Mining and Analysis **Data Structures and Algorithm Analysis in C+** Algorithms

"Problem solving is an essential part of every scientific discipline. It has two components: (1) problem identification and formulation, and (2) the solution to the formulated problem. One can solve a problem on its own using ad hoc techniques or by following techniques that have produced efficient solutions to similar problems. This requires the understanding of various algorithm design techniques, how and when to use them to formulate solutions, and the context appropriate for each of them. Algorithms: Design Techniques and Analysis advocates the study of algorithm design by presenting the most useful techniques and illustrating them with numerous examples -- emphasizing on design techniques in problem solving rather than algorithms topics like searching and sorting. Algorithmic analysis in connection with example algorithms are explored in detail. Each technique or strategy is covered in its own chapter through numerous examples of problems and their algorithms. Readers will be equipped with problem solving tools needed in advanced courses or research in science and engineering."--Provided by publisher. Data Structures & Theory of Computation This book, on Design and Analysis of Algorithms, in its second edition, presents a detailed coverage of the time complexity of algorithms. In this edition, a number of chapters have been modified and updated with new material. It discusses the various design factors that make one algorithm more efficient than others, and explains how to devise the new algorithms or modify the existing ones. The book begins with an introduction to algorithm analysis and then presents different methods and techniques—divide and conquer methods, the greedy method, search and traversal techniques, backtracking methods, branch and bound methods—used in the design of algorithms. Each algorithm that is written in this book is followed first by a detailed explanation and then is supported by worked-out examples. The book contains a number of figures to illustrate the theoretical aspects and also provides chapter-end questions to enable students to gauge their understanding of the underlying concepts. What distinguishes the text is its compactness, which

has been achieved without sacrificing essential subject matter. This text is suitable for a course on “Design and Analysis of Algorithms”, which is offered to the students of B.Tech (Computer Science and Engineering) and undergraduate and postgraduate students of computer science and computer applications [BCA, MCA, B.Sc. (CS), M.Sc. (CS)] and other computer-related courses. New to this Edition : Explains in detail the time complexity of the algorithms for the problem of finding the GCD and matrix addition. Covers the analysis of Knapsack and Combinatorial Search and Optimization problems. Illustrates the “Branch-and-Bound” method with reference to the Knapsack problem. Presents the theory of NP-Completeness. A computer algorithm is a set of instructions for performing calculation, data processing or automated reasoning. An initial state and input is provided, after which the algorithm proceeds through a succession of finite states to produce a final state and output. Algorithms may be classified on the basis of their implementation into recursive algorithm, logical algorithm, deterministic or non-deterministic algorithm, etc. They may also be classified as divide and conquer algorithm, search algorithm, randomized algorithm, etc. depending on the design paradigm or methodology. The study and analysis of algorithms is an important area of computer science. Algorithmic analysis is required to determine how much of a particular resource is required for a given algorithm. It is usually practiced without the implementation of a specific programming language. Most algorithms are applied on hardware/software platforms in which their algorithmic efficiency is evaluated using real code. For fast, interactive and commercial or scientific usage, algorithm efficiency is vital. The topics included in this book on computer algorithms are of utmost significance and bound to provide incredible insights to readers. Also included herein is a detailed explanation of the various aspects of the design, analysis and applications of algorithms. This book, with its detailed analyses and data, will prove immensely beneficial to professionals and students involved in this area at various levels. Probabilistic Analysis of Algorithms begins with a presentation of the "tools of the trade" currently used in probabilistic analyses, and continues with an applications section in which these tools are used in the analysis ofr selected algorithms. The tools section of the book provides the reader with an arsenal of analytic and numeric computing methods which are then applied to several groups of algorithms to analyze their running time or storage requirements characteristics. Topics covered in the applications section include sorting, communications network protocols and bin packing. While the discussion of the various algorithms is sufficient to motivate their structure, the emphasis throughout is on the probabilistic estimation of their operation under distributional assumptions on their input. Probabilistic Analysis of Algorithms assumes a working knowledge of engineering mathematics, drawing on real and complex analysis, combinatorics and probability theory. While the book is intended primarily as a text for the upper undergraduate and graduate student levels, it contains a wealth of material and should also prove an important reference for researchers. As such it is addressed to computer scientists, mathematicians, operations researchers, and electrical and industrial engineers who are interested in evaluating the probable operation of algorithms, rather than their worst-case behavior. This text is based on a simple and fully reactive computational model that allows for intuitive comprehension and logical designs. The principles and techniques presented can be applied to any distributed computing environment (e.g., distributed systems, communication networks, data networks, grid networks, internet, etc.). The text provides a wealth of unique material for learning how to design algorithms and protocols perform tasks efficiently in a distributed computing environment. This book introduces the essential concepts of algorithm analysis required by core undergraduate and graduate computer science courses, in addition to providing a review of the fundamental mathematical notions necessary to understand these concepts. Features: includes numerous fully-worked examples and step-by-step proofs, assuming no strong mathematical background; describes the foundation of the analysis of algorithms theory in terms of the big-Oh, Omega, and Theta notations; examines recurrence relations; discusses the concepts of

basic operation, traditional loop counting, and best case and worst case complexities; reviews various algorithms of a probabilistic nature, and uses elements of probability theory to compute the average complexity of algorithms such as Quicksort; introduces a variety of classical finite graph algorithms, together with an analysis of their complexity; provides an appendix on probability theory, reviewing the major definitions and theorems used in the book. Analysis of Algorithms: Computational Methods & Mathematical Tools presents the methods and tools needed to determine the effectiveness of algorithms. It begins with basic computational tools such as generating functions, combinatorial calculus, and asymptomatic methods, and continues through applications such as searching and sorting, communications protocols, and bin packing heuristics. The techniques needed for an effective use of each concept are shown in examples, then in exercises for which detailed solutions are provided. Proofs are given to illustrate the focal topic of the chapter. While the book can be used as a reference tool for algorithm designers and scientists specializing in their analyses, the exercises also make this a useful guide for graduate courses and seminars. Much of the material is culled from recent journal articles, and is presented here for the first time in book form. A successor to the first edition, this updated and revised book is a great companion guide for students and engineers alike, specifically software engineers who design reliable code. While succinct, this edition is mathematically rigorous, covering the foundations of both computer scientists and mathematicians with interest in algorithms. Besides covering the traditional algorithms of Computer Science such as Greedy, Dynamic Programming and Divide & Conquer, this edition goes further by exploring two classes of algorithms that are often overlooked: Randomised and Online algorithms — with emphasis placed on the algorithm itself. The coverage of both fields are timely as the ubiquity of Randomised algorithms are expressed through the emergence of cryptography while Online algorithms are essential in numerous fields as diverse as operating systems and stock market predictions. While being relatively short to ensure the essentiality of content, a strong focus has been placed on self-containment, introducing the idea of pre/post-conditions and loop invariants to readers of all backgrounds. Containing programming exercises in Python, solutions will also be placed on the book's website. Contents:PreliminariesGreedy AlgorithmsDivide and ConquerDynamic ProgrammingOnline AlgorithmsRandomized AlgorithmsAppendix A: Number Theory and Group TheoryAppendix B: RelationsAppendix C: Logic Readership: Students of undergraduate courses in algorithms and programming. Keywords:Algorithms;Greedy;Dynamic Programming;Online;Randomized;Loop InvariantKey Features:The book is concise, and of a portable size that can be conveniently carried around by studentsIt emphasizes correctness of algorithms: how to prove them correct, which is of great importance to software engineersIt contains a chapter on randomized algorithms and applications to cryptography, as well as a chapter on online algorithms and applications to caching/paging, both of which are relevant and current topicsReviews: “Summing up, the book contains very nice introductory material for beginners in the area of correct algorithm's design.” Zentralblatt MATH 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 There has been an explosive growth in the field of combinatorial algorithms. These algorithms depend not only on results in combinatorics and especially in graph theory, but also on the development of new data structures and new techniques for analyzing algorithms. Four classical problems in network optimization are covered in detail, including a development of the data structures they use and an analysis of their running time. Data Structures and Network Algorithms attempts to provide the reader with both a practical understanding of the algorithms, described to facilitate their easy implementation, and an appreciation of the depth and beauty of the field of graph algorithms. Advances in Computational Algorithms and Data Analysis offers state of the art tremendous advances in computational algorithms and data analysis. The selected articles are representative in these subjects sitting on the top-end-high technologies. The volume serves as an excellent reference work for researchers and graduate students working on computational algorithms and data analysis. A timely book on a topic that has witnessed a surge of interestover the last decade, owing in part to several novel applications,most notably in data compression and computational molecularbiology. It describes methods employed in average case analysis ofalgorithms, combining both analytical and probabilistic tools in asingle volume. * Tools are illustrated through problems on words with applicationsto molecular biology, data compression, security, and patternmatching. * Includes chapters on algorithms and data structures on words,probabilistic and analytical models, inclusion-exclusionprinciples, first and second moment methods, subadditive ergodictheorem and large deviations, elements of information theory,generating functions, complex asymptotic methods, Mellin transformand its applications, and analytic poissonization anddepoissonization. * Written by an established researcher with a strong internationalreputation in the field. A comprehensive overview of data mining from an algorithmic perspective, integrating related concepts from machine learning and statistics. Network data are produced automatically by everyday interactions - social networks, power grids, and links between data sets are a few examples. Such data capture social and economic behavior in a form that can be analyzed using powerful computational tools. This book is a guide to both basic and advanced techniques and algorithms for extracting useful information from network data. The content is organized around 'tasks', grouping the algorithms needed to gather specific types of information and thus answer specific types of questions. Examples include similarity between nodes in a network, prestige or centrality of individual nodes, and dense regions or communities in a network. Algorithms are derived in detail and summarized in pseudo-code. The book is intended primarily for computer scientists, engineers, statisticians and physicists, but it is also accessible to network scientists based in the social sciences. MATLAB®/Octave code illustrating some of the algorithms will be available at: http://www.cambridge.org/9781107125773. An Algorithm is a sequence of steps to solve a problem. The Design and Analysis of Algorithm is very important for designing algorithms to solve different types of problems in the branch of computer science and information technology. This book introduces the fundamental concepts of Designing Strategies, Complexity analysis of Algorithms, followed by problems on Graph Theory, and Sorting methods. Techniques for Designing and Analyzing Algorithms Design and analysis of algorithms can be a difficult subject for students due to its sometimes-abstract nature and its use of a wide variety of mathematical tools. Here the author, an experienced and successful textbook writer, makes the subject as straightforward as possible in an up-to-date textbook incorporating various new developments appropriate for an introductory course. This text presents the main techniques of algorithm design, namely, divide-and-conquer algorithms, greedy algorithms,

dynamic programming algorithms, and backtracking. Graph algorithms are studied in detail, and a careful treatment of the theory of NP-completeness is presented. In addition, the text includes useful introductory material on mathematical background including order notation, algorithm analysis and reductions, and basic data structures. This will serve as a useful review and reference for students who have covered this material in a previous course. Features The first three chapters provide a mathematical review, basic algorithm analysis, and data structures Detailed pseudocode descriptions of the algorithms along with illustrative algorithms are included Proofs of correctness of algorithms are included when appropriate The book presents a suitable amount of mathematical rigor After reading and understanding the material in this book, students will be able to apply the basic design principles to various real-world problems that they may encounter in their future professional careers. Software -- Programming Techniques. Based on a new classification of algorithm design techniques and a clear delineation of analysis methods, Introduction to the Design and Analysis of Algorithms presents the subject in a coherent and innovative manner. Written in a student-friendly style, the book emphasises the understanding of ideas over excessively formal treatment while thoroughly covering the material required in an introductory algorithms course. Popular puzzles are used to motivate students' interest and strengthen their skills in algorithmic problem solving. Other learning-enhancement features include chapter summaries, hints to the exercises, and a detailed solution manual. The full text downloaded to your computer With eBooks you can: search for key concepts, words and phrases make highlights and notes as you study share your notes with friends eBooks are downloaded to your computer and accessible either offline through the Bookshelf (available as a free download), available online and also via the iPad and Android apps. Upon purchase, you'll gain instant access to this eBook. Time limit The eBooks products do not have an expiry date. You will continue to access your digital ebook products whilst you have your Bookshelf installed. A coherent and unified account of techniques for analyzing the performance of randomized algorithms. This book analyzes the basic techniques involved in providing efficient numerical solutions to problems in science and engineering and focuses on the development of numerical algorithms and detailed analysis of different numerical methods. While developing the basic understanding for the construction of numerical algorithms, it also emphasizes the applicability of these methods and explains the guaranteed accuracy that various methods provide for the efficiency and scalability of large-scale systems. In addition, it contains exercises, examples, and algorithms used for preparing computer programs in Mathematica®. This book is designed for the way we learn and intended for one-semester course in Design and Analysis of Algorithms . This is a very useful guide for graduate and undergraduate students and teachers of computer science. This book provides a coherent and pedagogically sound framework for learning and teaching. Its breadth of coverage insures that algorithms are carefully and comprehensively discussed with figures and tracing of algorithms. Carefully developing topics with sufficient detail, this text enables students to learn about concepts on their own, offering instructors flexibility and allowing them to use the text as lecture reinforcement.Key Features:" Focuses on simple explanations of techniques that can be applied to real-world problems." Presents algorithms with self-explanatory pseudocode." Covers a broad range of algorithms in depth, yet makes their design and analysis accessible to all levels of readers." Includes chapter summary, self-test quiz and exercises at the end of each chapter. Key to quizzes and solutions to exercises are given in appendices. * Provides a broad overview of modeling approaches and solution methodologies for addressing inventory problems, particularly the management of high cost, low demand rate service parts found in multi-echelon settings * The text may be used in a variety of courses for first-year graduate students or senior undergraduates, or as a reference for researchers and practitioners * A background in stochastic processes and optimization is assumed "All aspects pertaining to algorithm design and algorithm analysis have been discussed over the chapters in this book--

Design and Analysis of Algorithms"--Resource description page. 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 Each chapter focuses on a basic programming problem and works through a variety of options for its solution, thus covering the essentials, incorporating pedagogical material, and giving students the experience of analysis. Math concepts are explained in the appendices. Annotation copyright by Book News, Inc., Portland, OR The text covers important algorithm design techniques, such as greedy algorithms, dynamic programming, and divide-and-conquer, and gives applications to contemporary problems. Techniques including Fast Fourier transform, KMP algorithm for string matching, CYK algorithm for context free parsing and gradient descent for convex function minimization are discussed in detail. The book's emphasis is on computational models and their effect on algorithm design. It gives insights into algorithm design techniques in parallel, streaming and memory hierarchy computational models. The book also emphasizes the role of randomization in algorithm design, and gives numerous applications ranging from data-structures such as skip-lists to dimensionality reduction methods. This monograph collects some fundamental mathematical techniques that are required for the analysis of algorithms. It builds on the fundamentals of combinatorial analysis and complex variable theory to present many of the major paradigms used in the precise analysis of algorithms, emphasizing the more difficult notions. The authors cover recurrence relations, operator methods, and asymptotic analysis in a format that is concise enough for easy reference yet detailed enough for those with little background with the material. Analysis and Design of Algorithms provides a structured view of algorithm design techniques in a concise, easy-to-read manner. The book was written with an express purpose of being easy -- to understand, read, and carry. It presents a pioneering approach in the teaching of algorithms, based on learning algorithm design techniques, and not merely solving a collection of problems. This allows students to master one design technique at a time and apply it to a rich variety of problems. Analysis and Design of Algorithms covers the algorithmic design techniques of divide and conquer, greedy, dynamic programming, branch and bound, and graph traversal. For each of these techniques, there are templates and guidelines on when to use and not to use each technique. Many sections contain innovative mnemonics to aid the readers in remembering the templates and key takeaways. Additionally, the book covers NP-completeness and the inherent hardness of problems. The third edition includes a new

section on polynomial multiplication, as well as additional exercise problems, and an updated appendix. Written with input from students and professionals, Analysis and Design of Algorithms is well suited for introductory algorithm courses at the undergraduate and graduate levels. The structured organization of the text makes it especially appropriate for online and distance learning. A successor to the first and second editions, this updated and revised book is a leading companion guide for students and engineers alike, specifically software engineers who design algorithms. While succinct, this edition is mathematically rigorous, covering the foundations for both computer scientists and mathematicians with interest in the algorithmic foundations of Computer Science. Besides expositions on traditional algorithms such as Greedy, Dynamic Programming and Divide & Conquer, the book explores two classes of algorithms that are often overlooked in introductory textbooks: Randomised and Online algorithms — with emphasis placed on the algorithm itself. The book also covers algorithms in Linear Algebra, and the foundations of Computation. The coverage of Randomized and Online algorithms is timely: the former have become ubiquitous due to the emergence of cryptography, while the latter are essential in numerous fields as diverse as operating systems and stock market predictions. While being relatively short to ensure the essentiality of content, a strong focus has been placed on self-containment, introducing the idea of pre/post-conditions and loop invariants to readers of all backgrounds, as well as all the necessary mathematical foundations. The programming exercises in Python will be available on the web (see http: //www.msoltys.com/book for the companion web site). Contents: Preliminaries Greedy Algorithms Divide and Conquer Dynamic Programming Online Algorithms Randomized Algorithms Algorithms in Linear Algebra Computational Foundations Mathematical Foundations Readership: Students of undergraduate courses in algorithms and programming and associated professionals. Keywords: Algorithms;Greedy;Dynamic Programming;Online;Randomized;Loop InvariantReview:0 Introduces exciting new methods for assessing algorithms for problems ranging from clustering to linear programming to neural networks. In his paper Theory of Communication [Gab46], D. Gabor proposed the use of a family of functions obtained from one Gaussian by time-and frequency shifts. Each of these is well concentrated in time and frequency; together they are meant to constitute a complete collection of building blocks into which more complicated time-depending functions can be decomposed. The application to communication proposed by Gabor was to send the coeffi cients of the decomposition into this family of a signal, rather than the signal itself. This remained a proposal-as far as I know there were no seri ous attempts to implement it for communication purposes in practice, and in fact, at the critical time-frequency density proposed originally, there is a mathematical obstruction; as was understood later, the family of shifted and modulated Gaussians spans the space of square integrable functions [BBGK71, Per71] (it even has one function to spare [BGZ75] . . .) but it does not constitute what we now call a frame, leading to numerical insta bilities. The Balian-Low theorem (about which the reader can find more in some of the contributions in this book) and its extensions showed that a similar mishap occurs if the Gaussian is replaced by any other function that is "reasonably" smooth and localized. One is thus led naturally to considering a higher time-frequency density. Despite growing interest in the mathematical analysis of algorithms, basic information on methods and models has rarely been directly accessible to practitioners, researchers, or students. This book organizes and presents that knowledge, fully introducing today's primary techniques for mathematically analyzing algorithms. Robert Sedgewick and the late Philippe Flajolet have drawn from both classical mathematical and computer science material, integrating discrete mathematics, elementary real analysis, combinatorics, algorithms, and data structures. They focus on "average-case" or "probabilistic" analysis, while also covering tools for "worst case" or "complexity" analysis. Improvements in this edition include: Upgraded figures and code Newer style for presenting much of the text's math An all-new chapter on trees This book's thorough, self-contained coverage will help readers appreciate the field's

challenges, prepare them for advanced results covered in Donald Knuth's books, and provide the background they need to keep abreast of new research. Coverage includes: recurrences, generating functions, asymptotics, trees, strings, maps, sorting, tree search, string search, and hashing algorithms. Ideal for junior- or senior-level courses on mathematical analysis of algorithms, this book will also be useful in courses on discrete mathematics for computer scientists, and in introducing mathematics students to computer science principles related to algorithms and data structures. In The Second Edition Of This Best-Selling Book, The Author Continues To Refine And Enhance His Innovative Approach To Algorithms And Data Structures. Using A C Implementation, He Highlights Conceptual Topics, Focusing On Adts And The Analysis Of Algorithms For Efficiency As Well As Performance And Running Time. In this second edition of his successful book, experienced teacher and author Mark Allen Weiss continues to refine and enhance his innovative approach to algorithms and data structures. Written for the advanced data structures course, this text highlights theoretical topics such as abstract data types and the efficiency of algorithms, as well as performance and running time. Before covering algorithms and data structures, the author provides a brief introduction to C++ for programmers unfamiliar with the language. Dr Weiss's clear writing style, logical organization of topics, and extensive use of figures and examples to demonstrate the successive stages of an algorithm make this an accessible, valuable text. New to this Edition *An appendix on the Standard Template Library (STL) *C++ code, tested on multiple platforms, that conforms to the ANSI ISO final draft standard 0201361221B04062001 These are my lecture notes from CS681: Design and Analysis of Algo rithms, a one-semester graduate course I taught at Cornell for three consec utive fall semesters from '88 to '90. The course serves a dual purpose: to cover core material in algorithms for graduate students in computer science preparing for their PhD qualifying exams, and to introduce theory students to some advanced topics in the design and analysis of algorithms. The material is thus a mixture of core and advanced topics. At first I meant these notes to supplement and not supplant a textbook, but over the three years they gradually took on a life of their own. In addition to the notes, I depended heavily on the texts • A. V. Aho, J. E. Hopcroft, and J. D. Ullman, The Design and Analysis of Computer Algorithms. Addison-Wesley, 1975. • M. R. Garey and D. S. Johnson, Computers and Intractibility: A Guide to the Theory of NP-Completeness. w. H. Freeman, 1979. • R. E. Tarjan, Data Structures and Network Algorithms. SIAM Regional Conference Series in Applied Mathematics 44, 1983. and still recommend them as excellent references. Analytic combinatorics aims to enable precise quantitative predictions of the properties of large combinatorial structures. The theory has emerged over recent decades as essential both for the analysis of algorithms and for the study of scientific models in many disciplines, including probability theory, statistical physics, computational biology, and information theory. With a careful combination of symbolic enumeration methods and complex analysis, drawing heavily on generating functions, results of sweeping generality emerge that can be applied in particular to fundamental structures such as permutations, sequences, strings, walks, paths, trees, graphs and maps. This account is the definitive treatment of the topic. The authors give full coverage of the underlying mathematics and a thorough treatment of both classical and modern applications of the theory. The text is complemented with exercises, examples, appendices and notes to aid understanding. The book can be used for an advanced undergraduate or a graduate course, or for self-study.