Introduction To Parallel Algorithms And Architectures Arrays Trees And Hypercubes

Parallel Computers 2

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.

Parallel Optimization

Algorithms and Parallel Computing

Introduction to Parallel Algorithms and Architectures: Arrays Trees Hypercubes provides an introduction to the expanding field of parallel algorithms and architectures. This book focuses on parallel computation involving the most popular network architectures, namely, arrays, trees, hypercubes, and some closely related networks. Organized into three chapters, this book begins with an overview of the simplest architectures of arrays and trees. This text then presents the structures and relationships between the dominant network architectures, as well as the most efficient parallel algorithms for a wide variety of problems. Other chapters focus on fundamental results and techniques and on rigorous analysis of algorithmic performance. This book discusses as well a hybrid of network architecture based on arrays and trees called the mesh of trees. The final chapter deals with the most important properties of hypercubes. This book is a valuable resource for readers with a general technical background.

Introduction to Parallel Computing

This concise text is designed to present the recent advances in parallel and distributed architectures and algorithms within an integrated framework. Beginning with an introduction to the basic concepts, the book goes on discussing the basic methods of parallelism exploitation in computation through vector processing, super scalar and VLIW processing, array processing, associative processing, systolic algorithms, and dataflow computation. After introducing interconnection networks, it discusses parallel algorithms for sorting, Fourier transform, matrix algebra, and graph theory. The second part focuses on basics and selected theoretical issues of distributed processing. Architectures and algorithms have been dealt in an integrated way throughout the book. The last chapter focuses on the different paradigms and issues of high performance computing making the reading more interesting. This book is meant for the senior level undergraduate and postgraduate students of computer science and engineering, and information technology. The book is also useful for the postgraduate students of computer science and computer application.

Parallel Algorithms

A complete source of information on almost all aspects of parallel computing from introduction, to
Parallel Processing and Parallel Algorithms

This contributed volume highlights two areas of fundamental interest in high-performance computing: core algorithms for important kernels and computationally demanding applications. The first few chapters explore algorithms, numerical techniques, and their parallel formulations for a variety of kernels that arise in applications. The rest of the volume focuses on state-of-the-art applications from diverse domains. By structuring the volume around these two areas, it presents a comprehensive view of the application landscape for high-performance computing, while also enabling readers to develop new applications using the kernels. Readers will learn how to choose the most suitable parallel algorithms for any given application, ensuring that theory and practicality are clearly connected. Applications using these techniques are illustrated in detail, including: Computational materials science and engineering, Computational cardiovascular analysis, Multiscale analysis of wind turbines and turbomachinery, Weather forecasting. Machine learning techniques, Parallel Algorithms in Computational Science and Engineering will be an ideal reference for applied mathematicians, engineers, computer scientists, and other researchers who utilize high-performance computing in their work.

Fundamentals of Sequential and Parallel Algorithms

Parallel Sorting Algorithms

Our aim in this book is to present and enlarge upon those aspects of parallel computing that are needed by practitioners of computational science. Today almost all classical sciences, such as mathematics, physics, chemistry and biology, employ numerical methods to help gain insight into nature. In addition to the traditional numerical methods, such as matrix inversions and the like, a whole new field of computational techniques has come to assume central importance, namely the numerical simulation methods. These methods are much less fully developed than those which are usually taught in a standard numerical mathematics course. However, they form a whole new set of tools for research in the physical sciences and are applicable to a very wide range of problems. At the same time there have been not only enormous strides forward in the speed and capability of computers but also dramatic new developments in computer architecture, and particularly in parallel computers. These improvements offer exciting prospects for computer studies of physical systems, and it is the new techniques and methods connected with such computer simulations that we seek to present in this book, particularly in the light of the possibilities opened up by parallel computers. It is clearly not possible at this early stage to write a definitive book on simulation methods and parallel computing.

Deep Learning and Parallel Computing Environment for Bioengineering Systems

Parallel Computing: Methods, Algorithms and Applications presents a collection of original papers presented at the international meeting on parallel processing, methods, algorithms, and applications at Verona, Italy in September 1989.

Introduction to Parallel Algorithms

Focusing on algorithms for distributed-memory parallel architectures, Parallel Algorithms presents a rigorous yet accessible treatment of theoretical models of parallel computation, parallel algorithm design for homogeneous and heterogeneous platforms, complexity and performance analysis, and essential notions of scheduling. The book extract:

Parallel Algorithms in Computational Science

A comprehensive guide for students and practitioners to parallel computing models, processes, metrics, and implementation in MPI and OpenMP.

Handbook of Parallel Computing

This textbook is a concise introduction to the basic toolbox of structures that allow efficient organization and retrieval of data, key algorithms for problems on graphs, and generic techniques for modeling, understanding, and solving algorithmic problems. The authors aim for a balance between simplicity and efficiency, between theory and practice, and between classical results and the forefront of research. Individual chapters cover arrays and linked lists, hash tables and associative arrays, sorting and selection, priority queues, sorted sequences, graph representation, graph traversal, shortest paths, minimum spanning trees, optimization, collective communication and computation, and load balancing. The authors also discuss important issues such as algorithm engineering, memory hierarchies, algorithm libraries, and certifying algorithms. Moving beyond the sequential algorithms and data structures of the earlier related title, this book takes into account the paradigm shift towards the parallel processing required to solve modern performance-critical applications and how this impacts on the teaching of algorithms. The book is suitable for undergraduate and graduate students and professionals familiar with programming.
and basic mathematical language. Most chapters have the same basic structure: the authors discuss a problem as it occurs in a real-life situation, they illustrate the most important applications, and then they introduce simple solutions as informally as possible and as formally as necessary so the reader really understands the issues at hand. As they move to more advanced and optional issues, their approach gradually leads to a more mathematical treatment, including theorems and proofs. The book includes many examples, pictures, informal explanations, and exercises, and the implementation notes introduce clean, efficient implementations in languages such as C++ and Java.

An Introduction to Parallel Algorithms Following an introduction to the basis of the fast Fourier transform (FFT), this book focuses on the implementation details on FFT for parallel computers. FFT is an efficient implementation of the discrete Fourier transform (DFT), and is widely used for many applications in engineering, science, and mathematics. Presenting many algorithms in pseudo-code and a complexity analysis, this book offers a valuable reference guide for graduate students, engineers, and scientists in the field who wish to apply FFT to large-scale problems. Parallel computation is becoming indispensable in solving the large-scale problems increasingly arising in a wide range of applications. The performance of parallel supercomputers is steadily improving, and it is expected that a massively parallel system with hundreds of thousands of compute nodes equipped with multi-core processors and accelerators will be available in the near future. Accordingly, the book also provides up-to-date computational techniques relevant to the FFT in state-of-the-art parallel computers. Following the introductory chapter, Chapter 2 introduces readers to the DFT and the basic idea of the FFT. Chapter 3 explains mixed-radix FFT algorithms, while Chapter 4 describes split-radix FFT algorithms. Chapter 5 explains multi-dimensional FFT algorithms, Chapter 6 presents high-performance FFT algorithms, and Chapter 7 addresses parallel FFT algorithms for shared-memory parallel computers. In closing, Chapter 8 describes parallel FFT algorithms for distributed-memory parallel computers.

Parallel Algorithms and Cluster Computing Numerical algorithms, modern programming techniques, and parallel computing are often taught serially across different courses and different textbooks. The need to integrate concepts and tools usually comes only in employment or in research - after the courses are concluded - forcing the student to synthesise what is perceived to be three independent subfields into one. This book provides a seamless approach to stimulate the student simultaneously through the eyes of multiple disciplines, leading to enhanced understanding of scientific computing as a whole. The book includes both basic as well as advanced topics and places equal emphasis on the discretization of partial differential equations and on solvers. Some of the advanced topics include wavelets, high-order methods, non-symmetric systems, and parallelization of sparse systems. The material covered is suited to students from engineering, computer science, physics and mathematics.

Synthesis of Parallel Algorithms This book offers a unique pathway to methods of parallel optimization by introducing parallel computing ideas into both optimization theory and into some numerical algorithms for large-scale optimization problems. The three parts of the book bring together relevant theory, careful study of algorithms, and modeling of significant real-world problems such as image reconstruction, radiation therapy treatment planning, financial planning, transportation and multi-commodity network flow problems, planning under uncertainty, and matrix balancing problems.

PARALLEL AND DISTRIBUTED COMPUTING: ARCHITECTURES AND ALGORITHMS

Introduction to Parallel Computing

Fast Parallel Algorithms for Graph Matching Problems Parallel Sorting Algorithms explains how to use parallel algorithms to sort a sequence of items on a variety of parallel computers. The book reviews the sorting problem, the parallel models of computation, parallel algorithms, and the lower bounds on the parallel sorting problems. The text also presents twenty different algorithms, such as linear arrays, mesh-connected computers, cube-connected computers. Another example where algorithm can be applied is on the shared-memory SIMD (single instruction stream multiple data stream) computers in which the whole sequence to be sorted can fit in the respective primary memories of the computers (random access memory), or in a single shared memory. SIMD processors communicate through an interconnection network or the processors communicate through a common and shared memory. The text also investigates the case of external sorting in which the sequence to be sorted is bigger than the available primary memory. In this case, the algorithms used in external sorting is very similar to those used to describe internal sorting, that is, when the sequence can fit in the primary memory. The book explains that an algorithm can reach its optimum possible operating time for sorting when it is running on a particular set of architecture, depending on a constant
Get Free Introduction To Parallel Algorithms And Architectures Arrays Trees And Hypercubes

multiplicative factor. The text is suitable for computer engineers and scientists interested in parallel algorithms.

Parallel Algorithms in Computational Science and Engineering Written by an authority in the field, this book provides an introduction to the design and analysis of parallel algorithms. The emphasis is on the application of the PRAM (parallel random access machine) model of parallel computation, with all its variants, to algorithm analysis. Special attention is given to the selection of relevant data structures and to algorithm design principles that have proved to be useful. Features Uses PRAM (parallel random access machine) as the model for parallel computation. Covers all essential classes of parallel algorithms. Rich exercise sets. Written by a highly respected author within the field. 0201548569B04062001

Introduction to Parallel Computing Take advantage of the power of parallel computers with this comprehensive introduction to methods for the design, implementation, and analysis of parallel algorithms. You'll examine many important core topics, including sorting and graph algorithms, discrete optimization techniques, and scientific computing applications, as you consider parallel algorithms for realistic machine models. Features: presents parallel algorithms as a small set of basic data communication operations in order to simplify their design and increase understanding; emphasizes practical issues of performance, efficiency, and scalability; provides self-contained discussion of the basic concepts of parallel computer architectures; covers algorithms for scientific computation, such as dense and sparse matrix computations, linear system solving, finite elements, and FFT; discusses algorithms for combinatorial optimization, including branch-and-bound, heuristic search, and dynamic programming; incorporates illustrative examples of parallel programs for commercially available computers; and contains extensive figures and examples that illustrate the workings of algorithms on different architectures.

Introduction to Parallel Algorithms and Architectures An Introduction to Parallel Programming, Second Edition presents a tried-and-true tutorial approach that shows students how to develop effective parallel programs with MPI, Pthreads and OpenMP. As the first undergraduate text to directly address compiling and running parallel programs on multi-core and cluster architecture, this second edition carries forward its clear explanations for designing, debugging and evaluating the performance of distributed and shared-memory programs. In edition, this new edition includes coverage of accelerators via new content on GPU programming and heterogeneous programming. New and improved user-friendly exercises teach student how to compile, run and modify example programs. Takes a tutorial approach, starting with small programming examples and building progressively to more challenging examples. Focuses on designing, debugging and evaluating the performance of distributed and shared-memory programs. Explains how to develop parallel programs using MPI, Pthreads and OpenMP programming models. Includes a robust package of online ancillaries for instructors and students. Provides lecture slides, a solutions manual, downloadable source code and an image bank.

Algorithms Sequential & Parallel: A Unified Approach This book is the result of several years of research trying to better characterize parallel genetic algorithms (pGAs) as a powerful tool for optimization, search, and learning. Readers can learn how to solve complex tasks by reducing their high computational times. Dealing with two scientific fields (parallelism and GAs) is always difficult, and the book seeks at gracefully introducing from basic concepts to advanced topics. The presentation is structured in three parts. The first one is targeted to the algorithms themselves, discussing their components, the physical parallelism, and best practices in using and evaluating them. A second part deals with the theory for pGAs, with an eye on theory-to-practice issues. A final third part offers a very wide study of pGAs as practical problem solvers, addressing domains such as natural language processing, circuits design, scheduling, and genomics. This volume will be helpful both for researchers and practitioners. The first part shows pGAs to either beginners and mature researchers looking for a unified view of the two fields: GAs and parallelism. The second part partially solves (and also opens) new investigation lines in theory of pGAs. The third part can be accessed independently for readers interested in applications. The result is an excellent source of information on the state of the art and future developments in parallel GAs.

Parallel Computing Over the last fifteen years GIS has become a fully-fledged technology, deployed across a range of application areas. However, although computer advances in performance appear to continue unhindered, data volumes and the growing sophistication of analysis procedures mean that performance will increasingly become a serious concern in GIS. Parallel computing offers a potential solution. However, traditional algorithms may not run effectively in a parallel environment, so utilization of parallel technology is not entirely straightforward. This groundbreaking book examines some of the current strategies facing scientists and engineers at this crucial interface of...
parallel computing and GIS; The book begins with an introduction to the concepts, terminology and techniques of parallel processing, with particular reference to GIS. High level programming paradigms and software engineering issues underlying parallel software developments are considered and emphasis is given to designing modular reusable software libraries. The book continues with problems in designing parallel software for GIS applications, potential vector and raster data structures and details the algorithmic design for some major GIS operations. An implementation case study is included, based around a raster generalization problem, which illustrates some of the principles involved. Subsequent chapters review progress in parallel database technology in a GIS environment and the use of parallel techniques in various application areas, dealing with both algorithmic and implementation issues; "Parallel Processing Algorithms for GIS" should be a useful text for a new generation of GIS professionals whose principal concern is the challenge of embracing major computer performance enhancements via parallel computing. Similarly, it should be an important volume for parallel computing professionals who are increasingly aware that GIS offers a major application domain for their technology.

Parallel Algorithms for Regular Architectures There is a software gap between the hardware potential and the performance that can be attained using today's software parallel program development tools. The tools need manual intervention by the programmer to parallelize the code. Programming a parallel computer requires closely studying the target algorithm or application, more so than in the traditional sequential programming we have all learned. The programmer must be aware of the communication and data dependencies of the algorithm or application. This book provides the techniques to explore the possible ways to program a parallel computer for a given application.

Introduction to Parallel Algorithms The matching problem is one of the central problems in graph theory as well as in the theory of algorithms and their applications. This book will provide the reader with a comprehensive and straightforward introduction to the basic methods of designing efficient parallel algorithms for graph matching problems. The text is written for students at the beginning graduate level. The exposition is mostly self-contained and example-driven. Prerequisites have been kept to a minimum by including relevant background material. The book contains full details of several new techniques and should also be of interest to research workers in computer science, operations research, discrete mathematics, and electrical engineering. The main theoretical tools are combined into three independent chapters, devoted to combinatorial tools, probabilistic tools, and algebraic tools. One of the main goals of the book is to bring together these three approaches and highlight how their combination works in the development of efficient parallel algorithms. The reader will be provided with a simple and transparent presentation of a variety of interesting algorithms, including many examples and illustrations. The combination of different approaches makes the matching problem and its applications an attractive and fascinating subject. It is hoped that the book represents a meeting point of interesting algorithmic techniques and opens up new algebraic and geometric areas.

Marek Karpinski is Chair Professor of Computer Science at the University of Bonn. Wojciech Rytter is Professor of Computer Science at the University of Warsaw and at the University of Liverpool.

Marek Karpinski is Chair Professor of Computer Science at the University of Bonn. Wojciech Rytter is Professor of Computer Science at the University of Warsaw and at the University of Liverpool.

Marek Karpinski is Chair Professor of Computer Science at the University of Bonn. Wojciech Rytter is Professor of Computer Science at the University of Warsaw and at the University of Liverpool.
Get Free Introduction To Parallel Algorithms And Architectures Arrays Trees And Hypercubes

An Introduction to Parallel Computing: Design and Analysis of Algorithms, 2/e Parallel Algorithms for Regular Architectures is the first book to concentrate exclusively on algorithms and paradigms for programming parallel computers such as the hypercube, mesh, pyramid, and mesh-of-trees. Algorithms are given to solve fundamental tasks such as sorting and matrix operations, as well as problems in the field of image processing, graph theory, and computational geometry. The first chapter defines the computer models, problems to be solved, and notation that will be used throughout the book. It also describes fundamental abstract data movement operations that serve as the foundation to many of the algorithms presented in the book. The remaining chapters describe efficient implementations of these operations for specific models of computation and present algorithms (with asymptotic analyses) that are often based on these operations. The algorithms presented are the most efficient known, including a number of new algorithms for the hypercube and mesh-of-trees that are better than those that have previously appeared in the literature. The chapters may be read independently, allowing anyone interested in a specific model to read the introduction and then move directly to the chapter(s) devoted to the particular model of interest. Russ Miller is Assistant Professor in the Department of Computer Science, State University of New York at Buffalo. Quentin F. Stout is Associate Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. Parallel Algorithms for Regular Architectures is included in the Scientific Computation series, edited by Dennis Gannon.

Parallel Genetic Algorithms The ability of parallel computing to process large data sets and handle time-consuming operations has resulted in unprecedented advances in biological and scientific computing, modeling, and simulations. Exploring these recent developments, the Handbook of Parallel Computing: Models, Algorithms, and Applications provides comprehensive coverage on a

Design and Analysis of Algorithms

Parallel Processing Algorithms For GIS Advancements in microprocessor architecture, interconnection technology, and software development have fueled rapid growth in parallel and distributed computing. However, this development is only of practical benefit if it is accompanied by progress in the design, analysis and programming of parallel algorithms. This concise textbook provides, in one place, three mainstream parallelization approaches, Open MPP, MPI and OpenCL, for multicore computers, interconnected computers and graphical processing units. An overview of practical parallel computing and principles will enable the reader to design efficient parallel programs for solving various computational problems on state-of-the-art personal computers and computing clusters. Topics covered range from parallel algorithms, programming tools, OpenMP, MPI and OpenCL, followed by experimental measurements of parallel programs’ run-times, and by engineering analysis of obtained results for improved parallel execution performances. Many examples and exercises support the exposition.


An Introduction to Parallel Algorithms Mathematics of Computing -- Parallelism.

Efficient Parallel Algorithms Deep Learning and Parallel Computing Environment for Bioengineering Systems delivers a significant forum for the technical advancement of deep learning in parallel computing environment across bio-engineering diversified domains and its applications. Pursuing an interdisciplinary approach, it focuses on methods used to identify and acquire valid, potentially useful knowledge sources. Managing the gathered knowledge and applying it to multiple domains including health care, social networks, mining, recommendation systems, image processing, pattern recognition and predictions using deep learning paradigms is the major strength of this book. This book integrates the core ideas of deep learning and its applications in bio-engineering application domains, to be accessible to all scholars and academicians. The proposed techniques and concepts in this book can be extended in future to accommodate changing business organizations’ needs as well as practitioners’ innovative ideas. Presents novel, in-depth research contributions from a methodological/application perspective in understanding the fusion of deep machine learning paradigms and their capabilities in solving a diverse range of problems. Illustrates the state-of-the-art and recent developments in the new theories and applications of deep learning approaches applied to parallel computing environment in bioengineering systems. Provides concepts and technologies that are successfully used in the implementation of today’s intelligent data-centric critical systems and multi-media Cloud-Big data.

An Introduction to Parallel Algorithms This book presents advances in high performance computing as well as
advances accomplished using high performance computing. It contains a collection of papers presenting results achieved in the collaboration of scientists from computer science, mathematics, physics, and mechanical engineering. From science problems to mathematical algorithms and on to the effective implementation of these algorithms on massively parallel and cluster computers, the book presents state-of-the-art methods and technology, and exemplary results in these fields.

Introducing Parallel Computing Motivation It is now possible to build powerful single-processor and multiprocessor systems and use them efficiently for data processing, which has seen an explosive expansion in many areas of computer science and engineering. One approach to meeting the performance requirements of the applications has been to utilize the most powerful single-processor system that is available. When such a system does not provide the performance requirements, pipelined and parallel processing structures can be employed. The concept of parallel processing is a departure from sequential processing. In sequential computation one processor is involved and performs one operation at a time. On the other hand, in parallel computation several processors cooperate to solve a problem, which reduces computing time because several operations can be carried out simultaneously. Using several processors that work together on a given computation illustrates a new paradigm in computer problem solving which is completely different from sequential processing. From the practical point of view, this provides sufficient justification to investigate the concept of parallel processing and related issues, such as parallel algorithms. Parallel processing involves utilizing several factors, such as parallel architecture, parallel algorithms, parallel programming languages and performance analysis, which are strongly interrelated. In general, four steps are involved in performing a computational problem in parallel. The first step is to understand the nature of computations in the specific application domain.

An Introduction to Parallel Programming THE CONTEXT OF PARALLEL PROCESSING The field of digital computer architecture has grown explosively in the past two decades. Through a steady stream of experimental research, tool-building efforts, and theoretical studies, the design of an instruction-set architecture, once considered an art, has been transformed into one of the most quantitative branches of computer technology. At the same time, better understanding of various forms of concurrency, from standard pipelining to massive parallelism, and invention of architectural structures to support a reasonably efficient and user-friendly programming model for such systems, has allowed hardware performance to continue its exponential growth. This trend is expected to continue in the near future. This explosive growth, linked with the expectation that performance will continue its exponential rise with each new generation of hardware and that (in stark contrast to software) computer hardware will function correctly as soon as it comes off the assembly line, has its down side. It has led to unprecedented hardware complexity and almost intolerable development costs. The challenge facing current and future computer designers is to institute simplicity where we now have complexity; to use fundamental theories being developed in this area to gain performance and ease-of-use benefits from simpler circuits; to understand the interplay between technological capabilities and limitations, on the one hand, and design decisions based on user and application requirements on the other.

Fast Fourier Transform Algorithms for Parallel Computers Parallel algorithms Made Easy The complexity of today’s applications coupled with the widespread use of parallel computing has made the design and analysis of parallel algorithms topics of growing interest. This volume fills a need in the field for an introductory treatment of parallel algorithms appropriate even at the undergraduate level, where no other textbooks on the subject exist. It features a systematic approach to the latest design techniques, providing analysis and implementation details for each parallel algorithm described in the book. Introduction to Parallel Algorithms covers foundations of parallel computing: parallel algorithms for trees and graphs; parallel algorithms for sorting, searching, and merging; and numerical algorithms. This remarkable book:* Presents basic concepts in clear and simple terms.* Incorporates numerous examples to enhance students’ understanding.* Shows how to develop parallel algorithms for all classical problems in computer science, mathematics, and engineering.* Employs extensive illustrations of new design techniques.* Discusses parallel algorithms in the context of PRAM model.* Includes end-of-chapter exercises and detailed references on parallel computing. This book enables universities to offer parallel algorithm courses at the senior undergraduate level in computer science and engineering. It is also an invaluable text/reference for graduate students, scientists, and engineers in computer science, mathematics, and engineering.

An Introduction to Data Structures and Algorithms Equip yourself for success with a state-of-the-art approach to algorithms available only in Miller/Boxer’s ALGORITHMS SEQUENTIAL AND PARALLEL: A UNIFIED APPROACH, 3E. This unique and functional text gives you an introduction to algorithms and paradigms for modern
computing systems, integrating the study of parallel and sequential algorithms within a focused presentation. With a wide range of practical exercises and engaging examples drawn from fundamental application domains, this book prepares you to design, analyze, and implement algorithms for modern computing systems. Important Notice: Media content referenced within the product description or the product text may not be available in the ebook version.

Introduction to Parallel Processing Introduction from ancient to modern times; Elementary data structures; Design analysis of sequential algorithms; Sequential sorting algorithms and their analysis; Introduction to parallel algorithms and architectures; Parallel sorting; Expanding the design and analysis of the algorithms toolkit; Introduction, correctness proofs, and recurrence relations; Graphs, digraphs, and sets; Probability and average complexity of algorithms; Introduction to Lower bound theory; Parallel prefix, matrix multiplication, and pointer jumping; Major design strategies; The Greedy method; Divide conquer; Dynamic programming; Backtracking and branch-and-bound; Special topics; Heuristic search: A* search, game trees; The dictionary problem: hashing and balanced trees; Probabilistic algorithms; Graph algorithms; NP-complete problems and the class NC; The classes NC and P-complete; Closing remarks.

Sequential and Parallel Algorithms and Data Structures Since the publication of the first edition, parallel computing technology has gained considerable momentum. A large proportion of this has come from the improvement in VLSI techniques, offering one to two orders of magnitude more devices than previously possible. A second contributing factor in the fast development of the subject is commercialization. The supercomputer is no longer restricted to a few well-established research institutions and large companies. A new computer breed combining the architectural advantages of the supercomputer with the advance of VLSI technology is now available at very attractive prices. A pioneering device in this development is the transputer, a VLSI processor specifically designed to operate in large concurrent systems. Parallel Computers 2: Architecture, Programming and Algorithms reflects the shift in emphasis of parallel computing and tracks the development of supercomputers in the years since the first edition was published. It looks at large-scale parallelism as found in transputer ensembles. This extensively rewritten second edition includes major new sections on the transputer and the OCCAM language. The book contains specific information on the various types of machines available, details of computer architecture and technologies, and descriptions of programming languages and algorithms. Aimed at an advanced undergraduate and postgraduate level, this handbook is also useful for research workers, machine designers, and programmers concerned with parallel computers. In addition, it will serve as a guide for potential parallel computer users, especially in disciplines where large amounts of computer time are regularly used.

Copyright code: d3927bbccad4d7a7bbcfb72766dad37e8