Previous Chapter Return to Table of Contents Next Chapter

PART VII: Selected Topics

Introduction

This part contains a selection of algorithmic topics that extend and complement earlier material in this book. Some chapters introduce new models of computation such as combinational circuits or parallel computers. Others cover specialized domains such as computational geometry or number theory. The last two chapters discuss some of the known limitations to the design of efficient algorithms and introduce techniques for coping with those limitations.

Chapter 28 presents our first parallel model of computation: comparison networks. Roughly speaking, a comparison network is an algorithm that allows many comparisons to be made simultaneously. This chapter shows how to build a comparison network that can sort n numbers in $O(\lg^2 n)$ time.

Chapter 29 introduces another parallel model of computation: combinational circuits. This chapter shows that two n-bit numbers can be added in $O(\lg n)$ time using a combinational circuit called a carry-lookahead adder. It also shows how to multiply two n-bit numbers in $O(\lg n)$ time.

Chapter 30 introduces a general model of parallel computation called the PRAM. The chapter presents basic parallel techniques, including pointer jumping, prefix computations, and the Euler-tour technique. Most of the techniques are illustrated on simple data structures, including lists and trees. The chapter also discusses general issues in parallel computation, including work efficiency and concurrent access to shared memory. It proves Brent's theorem, which shows how a parallel computer can efficiently simulate a combinational circuit. The chapter concludes with a work-efficient, randomized algorithm for list ranking and a remarkably efficient deterministic algorithm for symmetry breaking in a list.

Chapter 31 studies efficient algorithms for operating on matrices. It begins with Strassen's algorithm, which can multiply two n x n matrices in $O(n^{2.81})$ time. It then presents two general methods--LU decomposition and LUP decomposition--for solving linear equations by Gaussian elimination in $O(n^3)$ time. It also shows that Strassen's algorithm can be used to solve linear systems faster and that, asymptotically, matrix inversion and matrix multiplication can be performed equally fast. The chapter concludes by showing how a least-squares approximate solution can be computed when a set of linear equations has no exact solution.

Chapter 32 studies operations on polynomials and shows that a well-known signal-processing technique--the Fast Fourier Transform (FFT)--can be used to multiply two

degree-n polynomials in $O(n \lg n)$ time. It also investigates efficient implementations of the FFT, including a parallel circuit.

Chapter 33 presents number-theoretic algorithms. After a review of elementary number theory, it presents Euclid's algorithm for computing greatest common divisors. Algorithms for solving modular linear equations and for raising one number to a power modulo another number are presented next. An interesting application of number-theoretic algorithms is presented next: the RSA public-key cryptosystem. This cryptosystem not only can be used to encrypt messages so that an adversary cannot read them, it also can be used to provide digital signatures. The chapter then presents the Miller-Rabin randomized primality test, which can be used to find large primes efficiently--an essential requirement for the RSA system. Finally, the chapter covers Pollard's "rho" heuristic for factoring integers and discusses the state of the art of integer factorization.

Chapter 34 studies the problem of finding all occurrences of a given pattern string in a given text string, a problem that arises frequently in text-editing programs. An elegant approach due to Rabin and Karp is considered first. Then, after examining an efficient solution based on finite automata, the chapter presents the Knuth-Morris-Pratt algorithm, which achieves efficiency by cleverly preprocessing the pattern. The chapter closes with a presentation of a string-matching heuristic due to Boyer and Moore.

Computational geometry is the topic of Chapter 35. After discussing basic primitives of computational geometry, the chapter shows how a "sweeping" method can efficiently determine whether or not a set of line segments contains any intersections. Two clever algorithms for finding the convex hull of a set of points--Graham's scan and Jarvis's march--also illustrate the power of sweeping methods. The chapter closes with an efficient algorithm for finding the closest pair from among a given set of points in the plane.

Chapter 36 concerns NP-complete problems. Many interesting computational problems are NP-complete, but no polynomial-time algorithm is known for solving any of them. This chapter presents techniques for determining when a problem is NP-complete. Several classic problems are proved to be NP-complete: determining if a graph has a hamiltonian cycle, determining if a boolean formula is satisfiable, and determining if a given set of numbers has a subset that adds up to a given target value. The chapter also proves that the famous traveling-salesman problem is NP-complete.

Chapter 37 shows how approximation algorithms can be used to find approximate solutions to NP-complete problems efficiently. For some NP-complete problems, approximate solutions that are near optimal are quite easy to produce, but for others even the best approximation algorithms known work progressively more poorly as the problem size increases. Then, there are some problems for which one can invest increasing amounts of computation time in return for increasingly better approximate solutions. This chapter illustrates these possibilities with the vertex-cover problem, the traveling-salesman problem, the set-covering problem, and the subset-sum problem.

Go to Chapter 28 Back to Table of Contents