The book is not meant to be a state-of-the-art monograph and has been designed to be read by both undergraduates and graduates. There are some theorems and proofs, many examples, and an extensive set of problems. A novel feature of the book is the inclusion of full solutions of all problems which should make the book particularly useful for self study.

JOSEPH D. WARD

14[65D17]—*The mathematics of surfaces*, IV, Glen Mullineux (Editor), Oxford University Press, New York, NY, 1996, xiv+569 pp., 24 cm, cloth, $145.00

These are the proceedings from a conference at Brunel University in 1994. While otherwise a typical "Proceedings", it is distinguished by the two articles of R. E. Barnhill and N. Dyn on the work of the late John Gregory ("From computable error bounds through Gregory's square to convex combinations", and "Rational spline interpolation, subdivision algorithms and C^2 polygonal patches", respectively).

LARS B. WAHLBIN

15[11A05, 11A51, 11A55, 11T06, 11Y11, 11Y16, 68Q25]—*Algorithmic number theory, Volume* I: *Efficient algorithms*, by Eric Bach and Jeffrey Shallit, The MIT Press, Cambridge, MA, 1996, xvii+512 pp., $23\frac{1}{2}$ cm, hardcover, $55.00

This book treats the design and analysis of algorithms for solving problems in elementary number theory for which more or less efficient algorithms are known. For example, good algorithms are known for testing large integers for primality, but none are known for factoring large composite integers. Primality testing appears in Chapter 9 of this book, while factoring is reserved for a projected second volume.

Algorithmic number theory is one of the principal sources of examples of problems in complexity classes studied in theoretical computer science. This is especially true for the randomized or probabilistic complexity classes. For example, let $\mathcal{RP}$ denote the class of languages (sets) L for which there is a randomized algorithm (one which can choose random numbers) whose running time is bounded by a polynomial in the size of the input, which accepts inputs in L (says that the input is an element of L if it really is in L) with probability ≥ 0.5, and which rejects every input not in L (says that the input is not in L whenever it really is not in L). The algorithm is allowed to assert that an input is not in L when it really is in L, provided that this happens for no more than half of the choices of random numbers. An algorithm in class $\mathcal{RP}$ is called a Monte Carlo algorithm.

Let `COMPOSITE` be the language of composite numbers, that is, the set of binary representations of all composite positive integers $\{4, 6, 8, 9, 10, \ldots\}$. Here is a Monte Carlo algorithm which shows that `COMPOSITE` is in the complexity class $\mathcal{RP}$. Let

(b/n) denote the Jacobi symbol.

Input a binary number n.

If $n < 4$, print $n \notin$ COMPOSITE and stop.

Choose a random integer b in $1 < b < n$ with uniform distribution.

If $\gcd(2b, n) \neq 1$, print $n \in$ COMPOSITE and stop.

If $b^{(n-1)/2} \equiv (b/n) \pmod{n}$, print $n \notin$ COMPOSITE and stop.

Print $n \in$ COMPOSITE and stop.

If the input number n is a prime > 4, then the algorithm will say that n is not composite because $b^{(n-1)/2} \equiv (b/n) \pmod{n}$ by Euler's Criterion. Solovay and Strassen proved that if n is odd and composite, then $b^{(n-1)/2} \not\equiv (b/n) \pmod{n}$ for at least one-half of all b in $1 < b < n$ which are relatively prime to n. Since one can compute $\gcd(b, n)$, $b^{(n-1)/2} \bmod n$, and (b/n) in $O(\log^3 n)$ bit operations when $1 < b < n$, COMPOSITE is in complexity class $\mathcal{RP}$.

Chapter 1 is a general introduction which precisely delineates the material covered and distinguishes it from computational number theory. To the authors, computational number theory consists of constructing tables, gathering evidence for conjectures, searching for counterexamples, and proofs by enumeration of cases, all in number theory. This contrasts with algorithmic number theory, which studies number-theoretic algorithms and may be defined as finding solutions to equations, or proving their nonexistence, while making efficient use of time and space. For example, from this viewpoint, proving that n is prime means efficiently showing that the equation $n = xy$ has no solution in integers $x, y > 1$. This chapter also outlines the basic facts and history of number theory and computational complexity.

Chapter 2 presents the basic results of elementary and analytic number theory which are needed later. These include Euler's Theorem, the Möbius inversion formula, Euler's summation formula, and formulas for estimating sums taken over the primes.

Chapter 3 is a survey of computational complexity theory, including language classes, reductions, $\mathcal{NP}$-completeness, and computational models.

After these preliminaries, Chapter 4 begins the real subject matter of the book by analyzing the Euclidean and binary algorithms for the greatest common divisor. The Euclidean algorithm computes a GCD by repeatedly dividing the larger number by the smaller one and replacing the larger number by the remainder in this division. When the smaller number is 0, the large one is the answer. The binary GCD algorithm removes all factors of 2 from the input numbers and then repeatedly subtracts the smaller number from the larger one. The larger number is replaced by the difference after all factors of 2 have been removed. The power of 2 in the GCD is computed separately. The Euclidean algorithm was the first nontrivial algorithm whose worst-case running time was analyzed. This was done more than 150 years ago by Lamé.

Chapter 5 continues with basic algorithms for computing in the ring of integers modulo n: the power algorithm for $a^e \bmod n$, the Chinese Remainder Theorem, and algorithms for Legendre and Jacobi symbols.

There are many analogies between the ring $\mathbf{Z}$ of integers and the polynomial ring $k[X]$, where k is a finite field. Chapter 6 discusses finite fields and some of these analogies. It describes the structure of the field $k[X]/(f)$, where f is an irreducible

polynomial in $k[X]$. Also discussed are the Euclidean algorithm in $k[X]$, continued fractions in the field $k((\frac{1}{X}))$ of expressions $f = \sum_{i \le d} c_i X^i$ with $c_i \in k$, and the generalized Jacobi symbol (g/f) for $f, g \in k[X]$.

Chapter 7 studies algorithms for solving certain equations over finite fields. It begins with the algorithms of Tonelli and Cipolla for finding square roots in $\mathbf{F}_q$ and continues with efficient algorithms for computing dth roots in $\mathbf{F}_q$. It describes Hensel's lemma, randomized algorithms for factoring polynomials, and what is known about deterministic algorithms for factoring polynomials.

Chapter 8 reviews important results from analytic number theory needed to analyze algorithms for prime numbers. It discusses the prime number theorem, the Riemann Hypothesis, the Extended Riemann Hypothesis, primitive roots, Linnik's theorem on primes in arithmetic progression, the difference between consecutive primes, and extensions of this theory to prime ideals in algebraic number fields. The final section lists many explicit estimates for functions related to prime numbers.

Chapter 9 discusses algorithms for testing primality, for generating "random" prime numbers, for finding the nth prime number, for computing the number $\pi(x)$ of primes $\le x$, and for creating a table of primes between 1 and n. Primality tests described here include converses to Fermat's theorem, special tests for Fermat and Mersenne numbers, probabilistic prime tests, fast tests which are valid if the Extended Riemann Hypothesis is true, and correct tests whose running time is small provided the Generalized Riemann Hypothesis holds. The authors mention Carmichael numbers, Euler pseudoprimes and strong pseudoprimes. These are the numbers that cause certain probabilistic prime tests to produce incorrect results. Euler pseudoprimes are the composite numbers which the Monte Carlo algorithm above says are prime. Prime tests using elliptic curves are saved for the second volume, as is the result of Aldeman and Huang that the set of primes is in complexity class $\mathcal{ZPP}$. Sieves are recommended for constructing tables of primes and also for tabulating the number of prime divisors function, $d(n)$. The authors give a simplified version of the algorithm of Lagarias, Miller and Odlyzko for computing $\pi(x)$ in $O(x^{2/3+\varepsilon})$ time and $O(x^{2/3+\varepsilon})$ space.

There are dozens of interesting exercises at the end of each chapter, as well as extensive notes which embellish the text, tell the history of the subject matter, and give references to the literature.

Appendix A gives solutions, or at least hints of or references to solutions for every exercise except the few that are open problems. The bibliography lists more than 1750 references. There is a two page index to notation and 24 pages of ordinary index.

The only typo known to me is one reported to me by an author: The bound on q in Theorem 8.7.13 on page 232 should be $O(p^{2e}(\log n + e \log p)^2)$.

This beautifully written volume is an excellent survey of the subject. I look forward to seeing the second volume.

S. S. Wagstaff, Jr.
Department of Computer Sciences
Purdue University
West Lafayette, Indiana 47907-1398