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