Review of the book

"Computational Number Theory"

by Abhijit Das

CRC Press, Taylor & Francis Group, 2013

ISBN: 978-1-4398-6615-3

Jorge Nakahara Jr

1 Summary of the review

In this report I present my review of the book "Computational Number Theory" by Abhijit Das.
It is a textbook that grew out of lecture notes the author developed for teaching computational number theory to MTech and final-year BTech students from the Dept. of Computer Science and Engineering, the Dept. of Mathematics, and other engineering departments at the Indian Institute of Technology (IIT) Kharagpur.
The author wrote these notes since he could not find a single book with all the theory and practical aspects (e.g. software implementation) he was looking for for his lectures.
This book was designed to fit the needs and context of engineering students in advanced undergraduate and graduate levels. Therefore, the emphasis of the book is on (software) implementation aspects of computational algorithms rather than on theory.
Consequently, the audience for this book includes students and practitioners rather than theoreticians. Some proofs are provided but most are omitted. The focus is on practical aspects of computational algorithms and their time and storage complexities, such as how to represent arbitrarily large values in a finite machine word, how to deal with carry and borrow bits, how to deal with sign and magnitude values, how to store and access multi-precision integer values, etc.
Plenty of exercises and examples make the presentation didactic and suited for lectures and for self study.
The material in this books is enough for a one-semester course on the subject, and depending on the content, also for two semesters. Answers to selected exercises are listed at the end of the book.
To ease the calculations, the readers do not need to implement the algorithms themselves in any particular programming language. Rather, the au-

thor suggests and exemplifies the theory using the freely-available software GP-PARI, a number-theory calculator (that is available in several different operating systems). Examples and instructions on how to program in GP-PARI are provided along the chapters, as well as with exercises.

2 Summary of the book

The content of this book is divided into nine chapters. Chap. 1 deals with basic arithmetic operations, fast arithmetic (e.g. extended GCD algorithm), congruences, Chinese Remainder theorem, Hensel lifting, primitive roots, continued fractions and the Riemann hypothesis.
Chap. 2 deals with finite fields arithmetic, representation and properties, polynomial GCD and inverses, Fermat's little theorem, normal elements, normal and optimal bases.
Chap. 3 deals with arithmetic of polynomials in finite fields, irreducible polynomials, roots of polynomials, polynomial factorization (Berlekamp's algorithm, LLL-factoring) and properties of polynomials with integer coefficients.
Chap. 4 discusses the arithmetic of elliptic curves, basic concepts, elliptic curves over finite fields, affine and projective curves, pairings on elliptic curves, Weil and Tate pairings, pairing-friendly curves, efficient implementations and point counting (Schoof's algorithm).
Chap. 5 is concerned with primality testing. Topics in this chapter include Pratt certificates, the sieve of Eratostenes, probabilistic primality testing, tests such as Fermat, Solovay-Strassen, Miller-Rabin, Fibonacci and Lucas. Also, deterministic primality tests are presented, including the celebrated AKS test. Finally, tests for primes of special form are described such as Pepin's test for Fermat numbers and Lucas-Lehmer test for Mersenne numbers.
Chap. 6 discusses integer factorization, Pollard's rho method, Floyd's and Brent's variants, Pollard's p-1 method, Dixon's and Cfrac methods, the quadratic sieve method, the cubic, the elliptic curve and the number field-sieve methods.
Chap. 7 discusses the discrete logarithm problem, listing classical attacks such as Shank's baby-step-giant-step, Pollard's rho, Pollard's lambda and Pohlig-Hellman methods. Algorithms over finite fields include the linear sieve, the residue-list sieve, the Gaussian integer method, the cubic sieve and the number-field sieve method.
Algorithms for general fields include the index calculus and the function-field sieve methods.
Chap. 8 gives methods to solve large sparse systems of linear equations. There is the usual Gaussian elimination technique, but also the Lanczos and Wiedemann methods, as well as block variants of both Lanczos and

Wiedemann methods.
Chap. 9 gives applications to public-key cryptography using the techniques explained in previous chapters. Some classical public-key cryptosystems are detailed such as RSA and ElGamal. Further, key agreement and digital signatures schemes are explained, like RSA, ElGamal, DSA and ECDSA signatures. Another section describes pairing-based cryptography, including identity-based encryption (IDE), the Boneh-Franklin IDE, key-agreement based on pairings, and identity-based signatures, such as Shamir's and Paterson's schemes.

3 What is the book like (style)?

The style of this book is didactic with plenty of examples and step-by-step calculations. Also, lots of exercises are presented at the end of the chapters to further test the students on the theory.
Limited background is required from the reader. A previous course on abstract algebra would be helpful. These facts make the book appropriate also for self study.
Readers interested in the proofs (which are omitted) are provided with appropriate bibliographic references.

4 Would you recommend this book?

This is a textbook that grew out of lecture notes by the author developed over a period of five years.
The implementation aspects of computational number theory algorithms are certainly crucial to any student as well to all practitioners. There are several algorithmic details in practice that are not treated (not taken care of) in a theoretical setting. One only finds this out when one needs to actually implement them (in a particular computing and software/hardware environment).
The main audience includes not only programmers, engineers, students in general, but also lecturers looking for a hands-on bibliographic reference covering this important practical side of computational number theory algorithms.