Preface

> The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic. ... Nevertheless we must confess that all methods that have been proposed thus far are either restricted to very special cases or are so laborious and prolix that even for numbers that do not exceed the limits of tables constructed by estimable men, i.e. for numbers that do not yield to artificial methods, they try the patience of even the practiced calculator. ... The dignity of the science itself seems to require that every possible means be explored for the solution of a problem so elegant and so celebrated.
>
> *C. F. Gauss* [**Gau01**, Art. 329]

Factoring integers is important. Gauss said so in 1801.

The problem of distinguishing prime numbers from composite numbers has been solved completely, both theoretically and practically. We have made some progress on factoring composite integers, but it remains a difficult problem.

Some mathematicians play a version of the television game show *Jeopardy!* with multiplication tables. Host Alex Trebek reads the

answer "thirty-five." A contestant rings in and gives the question, "What is five times seven?" Mathematicians currently play this game with 200-digit numbers rather than 2-digit numbers.

This book is intended for readers who have taken an introductory number theory course and want to learn more about factoring. This work offers many reasons why factoring is important. Chapter 2 reviews the elementary number theory material the reader is assumed to know. To fully understand this book, the reader will also need calculus and linear algebra. As factoring integers usually involves computers, the reader is assumed to be computer-literate and to understand simple pseudocode and protocols. In a few places we assume the reader is familiar with the notions of polynomial time (easy problem) and the nondeterministic polynomial-time class $\mathcal{NP}$ (hard problem).

This book explains and motivates the Cunningham Project, the largest factoring enterprise in the world today. The official tables of the Cunningham Project are published as [**BLS**$^+$**02**]; the first part of that book includes some material from this work in condensed form.

For readers not interested in the Cunningham Project, this book offers numerous other applications of factoring, especially to cryptography, and gives important results in these areas.

There is tremendous pleasure in devising a new factoring method, programming it, and using it to factor a number no one else could split. I hope that some readers will participate in this endeavor and experience the joy. The end of the last chapter suggests where the reader might begin.

In the chapters to follow we will give many reasons why the factorizations of certain numbers are important and useful. We will describe some of the major algorithms and devices for factoring and tell a little about the people who invented them. Finally, we will tell how you can help with some of the factoring projects currently in progress.

Chapters 1 and 4 tell some reasons why people factor integers. The discussion in Chapter 1 requires no more than high school mathematics. Chapter 4 gives additional reasons understood better with the (college-level) number theory of Chapters 2 and 3.

For the past thirty-five years, a very important reason for factoring has been the public-key cipher of Rivest, Shamir, and Adleman (RSA), whose security requires that the problem of factoring integers is hard. Chapter 1 describes the development of the RSA cipher. The mathematical details of it are presented in Chapter 4. Chapter 1 also discusses three older reasons for interest in factoring, repunits, decimal fractions, and perfect numbers. Then it describes the Cunningham Project, more than a century old and the greatest integer factoring collaboration in history.

Chapter 2 reviews some elementary number theory found in a first course in the subject. It considers divisibility, prime numbers, congruences, Euler's theorem, arithmetic functions, and Quadratic Reciprocity. Few proofs are given here. It is assumed that the reader has learned this material elsewhere. A few algorithms, such as the Euclidean Algorithm for the greatest common divisor, are stated.

Chapter 3 deals with more advanced number theory, probably not taught in a first course but needed to understand factoring algorithms and applications of factoring. It discusses the frequency of occurrence of integers whose greatest prime factor is small, how to compute modular square roots, cyclotomic polynomials, primality testing, and divisibility sequences such as the Fibonacci numbers and the numbers $b^m - 1$, $m = 1, 2, 3, \ldots$, for fixed b. Of course, the recognition of primes is essential to telling when a factorization is complete, and this topic is treated extensively. Many algorithms are stated in Chapter 3.

More applications of factoring are given in Chapter 4. It begins with a set of algebraic factorizations of some numbers in a divisibility sequence discovered by Aurifeuille and others. A more complete discussion of perfect numbers follows, with a sample of theorems in this area. Next come harmonic numbers, prime proving aided by factoring, and linear feedback shift registers, which are hardware devices used to generate cryptographic keys and random numbers. Testing conjectures is an important and common application of factoring. We give three examples. Bernoulli numbers are connected to the structure of cyclotomic fields and to Fermat's Last Theorem. While most work in this area is beyond the scope of this book, we do give a taste

of the possible results. The chapter ends with a deeper discussion of public-key cryptography, more applications of factoring to cryptography, and other assorted uses of factoring. These include accelerating RSA signature generation, zero-knowledge proofs, and sums of two or four squares.

The remaining chapters discuss methods of factoring integers. Each chapter presents algorithms with related ideas, roughly in historical order. Most algorithms are described both in words and in pseudocode. Simple examples are given for each method. Chapter 5 gives the oldest, simplest, and slowest algorithms, from Trial Division and Fermat's Method to techniques Pollard developed in the 1970s.

Chapter 6 treats simple continued fractions and several factoring algorithms that use them. While most of these have been superseded by faster algorithms, at least one of them (SQUFOF) is often used as a procedure in the powerful sieve algorithms of Chapter 8. Chapter 6 also proves a simple but important theorem (Theorem 6.18) that tells how most of the factoring algorithms in Chapter 6 and Chapter 8 finish, that is, how the factors are produced at the end of the algorithm.

In Chapter 7, we examine the basic properties of elliptic curves and tell how they lead to good algorithms for factoring and primality proving. Elliptic curves have many uses in cryptography and data security. In some of these applications, factoring integers is an important tool for constructing elliptic curves with desirable properties to make computing with them efficient while maintaining security. Some of these techniques are mentioned in the final section of Chapter 7. This chapter uses the newest mathematics in the book.

Chapter 8 deals with the notion of sieve, from the Sieve of Eratosthenes more than 2,000 years old to the Number Field Sieve factoring algorithm about 25 years old. On the way we describe the Quadratic Sieve factoring algorithm and a couple of other sieves for factoring. The Number Field Sieve works especially well for numbers in the Cunningham Project.

In Chapter 9 we describe special hardware rather than software for factoring. Some of these are mechanical or electronic devices for

performing the sieve process. Others are computers with special architecture to facilitate factoring. We also discuss factoring with quantum objects and with DNA molecules. The author is neither a physicist nor a biologist, so he can give only a taste of these new factoring methods. The reader who really wants to learn about these topics should consult the references.

Chapter 10 discusses practical aspects of factoring and also some purely theoretical results about the difficulty of factoring. We tell how computers calculate with very large integers. Another section reveals special methods for factoring integers quickly when they have special form or when partial information is known about their factors. These tricks include applications to breaking the RSA cipher. We describe some of the ongoing factoring projects and how the reader can help with them. The final section tosses out some new ideas for possible future factoring methods.

Most of the algorithms in this work are written in pseudocode and described in words. We have tried to make the pseudocode clear enough so that programmers with limited knowledge of number theory can write correct programs. We have also tried to make the verbal descriptions of algorithms understandable to number theorists unfamiliar with computer programming.

This book does not discuss factoring integers mentally, although one reviewer suggested it as a topic. The only items at all related to mental arithmetic are Example 2.29 and Exercises 0.1 and 2.13.

If you have taken a course in elementary number theory, are computer-literate, don't care about applications, and wish to learn about factoring algorithms immediately, then you could begin reading with Chapter 5. But then you would wonder why we keep factoring the number 13290059 over and over again. This choice is explained in Section 4.6.3.

The author thanks CERIAS, the Center for Education and Research in Information Security and Assurance at Purdue University, for its support.

The author is grateful to Richard Brent, Greg Childers, Graeme Cohen, Carl Pomerance, and Richard Weaver, who answered questions about the material of this book. He is indebted to Robert

Baillie, Arjen Lenstra, Richard Schroeppel, Hugh Williams, and at least one anonymous reviewer for helpful comments. Hugh Williams generously allowed the use of the sieve photos in Chapter 9. The author thanks Junyu Chen for checking and programming many of the algorithms in this work. Any remaining errors are the responsibility of the author.

Keep the factors coming!

Sam Wagstaff

Exercise

0.1. No computers are allowed in multiplication table *Jeopardy!*. Alex Trebek reads the clue "299." You ring in and say what?