# Preface

In this volume we have endeavored to provide a middle ground—hopefully even a bridge—between "theory" and "experiment" in the matter of prime numbers. Of course, we speak of number theory and computer experiment. There are great books on the abstract properties of prime numbers. Each of us working in the field enjoys his or her favorite classics. But the experimental side is relatively new. Even though it can be forcefully put that computer science is by no means young, as there have arguably been four or five computer "revolutions" by now, it is the case that the theoretical underpinnings of prime numbers go back centuries, even millennia. So, we believe that there is room for treatises based on the celebrated classical ideas, yet authored from a modern computational perspective.

### Design and scope of this book

The book combines the essentially complementary areas of expertise of the two authors. (One author (RC) is more the computationalist, the other (CP) more the theorist.) The opening chapters are in a theoretical vein, even though some explicit algorithms are laid out therein, while heavier algorithmic concentration is evident as the reader moves well into the book. Whether in theoretical or computational writing mode, we have tried to provide the most up-to-date aspects of prime-number study. What we do not do is sound the very bottom of every aspect. Not only would that take orders of magnitude more writing, but, as we point out in the opening of the first chapter, it can be said that no mind harbors anything like a complete picture of prime numbers. We could perhaps also say that neither does any team of *two* investigators enjoy such omniscience. And this is definitely the case for the present team! What we have done is attempt to provide references to many further details about primes, which details we cannot hope to cover exhaustively. Then, too, it will undoubtedly be evident, by the time the book is available to the public, that various prime-number records we cite herein have been broken already. In fact, such are being broken as we write this very preface. During the final stages of this book we were in some respects living in what electronics engineers call a "race condition," in that results on primes—via the Internet and personal word of mouth—were coming in as fast or faster than editing passes were carried out. So we had to decide on a cutoff point. (In compensation, we often give pointers to websites that do indeed provide up-to-the-minute results.) The race condition has become a natural part of the game, especially now that computers are on the team.

### Exercises and research problems

The exercises occur in roughly thematic order at the end of every chapter, and range from very easy to extremely difficult. Because it is one way of conveying the nature of the cutting edge in prime-number studies, we have endeavored to supply many exercises having a research flavor. These are set off after each chapter's "Exercises" section under the heading "Research problems." (But we still call both normal exercises and research problems "exercises" during in-text reference.) We are not saying that all the normal exercises are easy, rather we flag a problem as a research problem if it can be imagined as part of a long-term, hopefully relevant investigation.

### Algorithms and pseudocode

We put considerable effort—working at times on the threshold of frustration—into the manner of algorithm coding one sees presented herein. From one point of view, the modern art of proper "pseudocode" (meaning not machine-executable, but let us say human-readable code) is in a profound state of disrepair. In almost any book of today containing pseudocode, an incompatibility reigns between readability and symbolic economy. It is as if one cannot have both.

In seeking a balance we chose the C language style as a basis for our book pseudocode. The appendix describes explicit examples of how to interpret various kinds of statements in our book algorithms. We feel that we shall have succeeded in our pseudocode design if two things occur:

(1) The programmer can readily create programs from our algorithms;

(2) All readers find the algorithm expositions clear.

We went as far as to ask some talented programmers to put our book algorithms into actual code, in this way verifying to some extent our goal (1). (Implementation code is available, in *Mathematica* form, at website http://www.perfsci.com.) Yet, as can be inferred from our remarks above, a completely satisfactory symbiosis of mathematics and pseudocode probably has to wait until an era when machines are more "human."

### Notes for this 2nd edition

Material and motive for this 2nd edition stem from several sources, as follows. First, our astute readers—to whom we are deeply indebted—caught various errors or asked for clarification, even at times suggesting new lines of thought. Second, the omnipresent edge of advance in computational number theory moves us to include new results. Third, both authors do teach and have had to enhance 1st edition material during course and lecture development. Beyond repairs of errors, reader-friendly clarifications, and the updating (through early 2005) of computational records, this 2nd edition has additional algorithms, each expressed in our established pseudocode style. Some of the added algorithms are new and exciting discoveries.

Examples of computationally motivated additions to this 2nd edition are as follows:

- The largest known explicit prime (as of Apr 2005) is presented (see Table 1.2), along with Mersenne search-status data.
- Other prime-number records such as twin-prime records, long arithmetic progressions of primes, primality-proving successes, and so on are reported (see for example Chapter 1 and its exercises).
- Recent factoring successes (most—but not all—involving subexponential methods) are given (see Section 1.1.2).
- Recent discrete- and elliptic-discrete-logarithm (DL and EDL, respectively) records are given (see Section 5.2.3 for the DL and Section 8.1.3 for the EDL cases).
- New verification limits for the Riemann hypothesis (RH) are given (Section 1.4.2).

Examples of algorithmic additions to this 2nd edition are as follows:

- We provide theory and algorithms for the new "AKS" method and its even newer variants for polynomial-time primality proving (see Section 4.5).
- We present a new fast method of Bernstein for detecting those numbers in a large set that have only small prime factors, even when the large set has no regular structure that might allow for sieving (see Section 3.3).
- We present the very new and efficient Stehlé–Zimmermann fast-gcd method (see Algorithm 9.4.7).
- We give references to new results on "industrial algorithms," such as elliptic-curve point-counting (see Section 7.5.2), elliptic algebra relevant to smart-cards (see for example Exercise 8.6), and "gigaelement" FFTs—namely FFTs accepting a billion complex input elements (end of Section 9.5.2).
- Because of its growing importance in computational number theory, a nonuniform FFT is laid out as Algorithm 9.5.8 (and see Exercise 1.62).

Examples of new theoretical developments surveyed in this 2nd edition are as follows:

- We discuss the sensational new theorem of Green and Tao that there are arbitrarily long arithmetic progressions consisting entirely of primes (see end of Section 1.1.5).
- We discuss the latest updates on the Fermat–Catalan conjecture that there are at most finitely many coprime positive integer powers $x^p, y^q, z^r$ with $x^p + y^q = z^r$ and with $1/p + 1/q + 1/r \leq 1$. The special case that one of these powers is the number 1 is also discussed: There is just the one solution $8 + 1 = 9$, a wonderful recent result of Mihăilescu (see Section 8.4), thus settling the original Catalan conjecture.

Exercises have changed in various ways. Additional exercises are presented, often because of new book algorithms. Some exercises have been improved. For example, where our 1st book edition said essentially, in some exercise, "Find a method for doing X," this 2nd edition might now say "Develop this outline on how to do X. Extend this method to do the (harder problem) Y."

**Acknowledgments**

The authors express their profound thanks to a diverse population of colleagues, friends, supporters—including astute readers of our 1st edition—whom we name as follows: S. Arch, E. Bach, D. Bailey, A. Balog, M. Barnick, P. Bateman, D. Bernstein, F. Beukers, O. Bonfim, D. Bleichenbacher, J. Borwein, D. Bradley, N. and P. Bragdon, R. Brent, D. Bressoud, D. Broadhurst, N. Bruin, Y. Bugeaud, L. Buhler, G. Campbell, M. Campbell, D. Cao, P. Carmody, E. Catmull, H. Cohen, D. Copeland, D. Coppersmith, J. Cosgrave, H. Darmon, T. Day, K. Dilcher, J. Doenias, G. Effinger, N. Elkies, T. Engelsma, J. Essick, J. Fessler, J. Fix, W. Galway, B. Garst, M. Gesley, G. Gong, A. Granville, D. Griffiths, R. Harley, E. Hasibar, D. Hayes, D. Hill, U. Hofmann, N. Howgrave-Graham, J. Huang, S. Jobs, A. Jones, B. Kaliski, W. Keller, M. Kida, K. Kim, J. Klivington, K. and S. Koblik, D. Kohel, D. Kramer, A. Kruppa, S. Kurowski, S. Landau, A. Lenstra, H. Lenstra, M. Levich, D. Lichtblau, D. Lieman, I. Lindemann, D. Loebenberger, M. Martin, E. Mayer, F. McGuckin, M. Mignotte, P. Mihăilescu, V. Miller, D. Mitchell, V. Mitchell, T. Mitra, P. Montgomery, W. Moore, V. Müller, G. Nebe, A. Odlyzko, H. Oki, F. Orem, J. Papadopoulos, N. Patson, A. Perez, J. Pollard, A. Powell, J. Powell, L. Powell, J. Renze, P. Ribenboim, B. Salzberg, A. Schinzel, T. Schulmeiss, J. Seamons, J. Shallit, M. Shokrollahi, J. Solinas, L. Somer, D. Stehlé, D. Symes, D. Terr, E. Teske, A. Tevanian, R. Thompson, M. Trott, S. Wagon, S. Wagstaff Jr., M. Watkins, P. Wellin, N. Wheeler, M. Wiener, T. Wieting, J. Williams, P. Winkler, S. Wolfram, G. Woltman, A. Wylde, A. Yerkes, A. Zaccagnini, Z. Zhang, and P. Zimmermann. These people contributed combinations of technical, theoretical, literary, moral, computational, debugging, and manuscript support. We would like to express a special gratitude to our long-time friend and colleague Joe Buhler, who quite unselfishly, and in his inimitable, encyclopedic style, aided us at many theoretical and computational junctures during the book project. Because of all of these spectacular colleagues, this book is immeasurably better than it would otherwise have been.

Portland, Oregon, USA — Richard Crandall
Hanover, New Hampshire, USA — Carl Pomerance

December 2000
December 2001 (Second printing, with corrections)
April 2005 (Second edition)

http://www.springer.com/978-0-387-25282-7

Prime Numbers
A Computational Perspective
Crandall, R.; Pomerance, C.
2005, XV, 597 p., Hardcover
ISBN: 978-0-387-25282-7