Selling Primes

PAULO RIBENBOIM*
Queen's University
Kingston, Ontario, Canada, K7L 3N6

I am a big shot in a factory that produces primes.

And I will tell you an interesting dialogue with a buyer, coming from an exotic country.

The Dialogue

—Buyer: I wish to buy some primes.

—I (generously): I can give to you, free of charge, many primes: 2, 3, 5, 7, 11, 13, 17, 19,....

—Buyer (interrupting my generous offer): Thank you, sir; but I want primes with 100 digits. Do you have these for sale?

—I: In this factory we can produce primes as large as you wish. There is in fact an old method of Euclid, which you may have heard about. If I have any number n of primes, say $p_1, p_2, \ldots, p_n$, we multiply them and add 1, to get the number $N = p_1 p_2 \cdots p_n + 1$. Either N is a prime or, if it is not a prime, we pick any prime dividing N. In this way, it is easy to see that we get a prime, which is different from the ones we mixed. Call it p_{n+1}. If we now mix $p_1, p_2, \ldots, p_n, p_{n+1}$ as I already said, we get still another prime p_{n+2}. Repeating this procedure we get as many primes as we wish and so, we are bound to get primes as large as we wish, for sure with at least 100 digits.

—Buyer: You are very nice to explain your procedure. Even in my distant country, I have heard about it. It gives primes that may be arbitrarily large. However, I want to buy primes that have exactly 100 digits, no more, no less. Do you have them?

—I: Yes. Long ago—at the beginning of last century—Bertrand observed that between any number $N > 1$ and its double $2N$, there exists at least one prime number. This experimental observation was confirmed by a rigorous proof by Chebyshev. So I can find the primes p_1, p_2, p_3 where

$$10^{99} < p_1 < 2 \times 10^{99}$$
$$2 \times 10^{99} < p_2 < 4 \times 10^{99}$$
$$4 \times 10^{99} < p_3 < 8 \times 10^{99}.$$

—Buyer: This means that you have guaranteed 3 primes with 100 digits, and perhaps a few more. But I want to buy many primes with 100 digits. How many can you produce?

—I: I have never counted how many primes of 100 digits could eventually be produced. I have been told that my colleagues in other factories have counted the total number of primes up to 10^{17}. We usually write $\pi(N)$ to denote the number of primes up to the number N. Thus, the count I mentioned has given:

$$\pi(10^8) = 5{,}761{,}455$$
$$\pi(10^9) = 50{,}847{,}534$$

*Lecture at the University of Augsburg, June 23, 1992.

$$\pi(10^{12}) = 37{,}607{,}912{,}018$$

$$\pi(10^{17}) = 2{,}625{,}557{,}157{,}654{,}233.$$

Even though all primes up to 10^{17} have not yet been produced by any factory, the count of $\pi(10^{17})$ is exact.

—Buyer (a bit astonished). If you cannot—as I understand—know how many primes of each large size there are in stock, how can you operate your factory and guarantee delivery of the merchandise?

—I: Your country sells oil, does it not? You can estimate the amount of oil at shallow depths quite accurately, but you cannot measure exactly the entire amount underground. It is just the same with us.

Gauss, one of the foremost scientists, discovered that

$$\pi(N) \sim \frac{N}{\log N}$$

for large values of N. This was confirmed, almost a century ago, by a proof given by Hadamard and de la Vallée Poussin.

—Buyer: Do you mean that $\pi(N)$ is approximately equal to $N/\log N$, with a small error?

—I: Yes. To be more precise, the relative error, namely the absolute value of the difference $|\pi(N) - N/\log N|$, divided by $\pi(N)$, tends to 0, as N increases indefinitely.

—Buyer: Then, because of the error, you cannot be very specific in your estimate. Unless you estimate the error.

—I: Correct (the buyer is not stupid...). Chebyshev showed, even before the prime number theorem was proved, that if N is large, then

$$0.9\frac{N}{\log N} < \pi(N) < 1.1\frac{N}{\log N}.$$

To count primes with 100 digits:

$$0.9\frac{10^{99}}{99\log 10} < \pi(10^{99}) < 1.1\frac{10^{99}}{99\log 10}$$

$$0.9\frac{10^{100}}{100\log 10} < \pi(10^{100}) < 1.1\frac{10^{100}}{100\log 10}.$$

It is easy to estimate the difference $\pi(10^{100}) - \pi(10^{99})$, which gives the number of primes with exactly 100 digits:

$$3.42 \times 10^{97} < \pi(10^{100}) - \pi(10^{99}) < 4.38 \times 10^{97}.$$

—Buyer: You are rich! I think you have more primes than we have oil. But I wonder how your factory produces the primes with 100 digits. I have an idea but I'm not sure how efficient my method would be.

1°) Write all the numbers with 100 digits.

2°) Cross out, in succession, all the multiples of 2, of 3, of 5, ..., of each prime p less than 10^{99}. For this purpose, spot the first multiple of p, then cross out every pth number.

What remains are the primes between 10^{99} and 10^{100}, that is, the primes with exactly 100 digits.

—I: This procedure is correct and was already discovered by Eratosthenes (in the 3rd century B.C.). In fact, you may stop when you have crossed out the multiples of all the primes less than 10^{50}.

However, this method of production is too slow. This explains why the archeologists never found a factory of primes amongst the Greek ruins, but just temples to Apollo, statues of Aphrodite (known as Venus, since the time of Romans), and other ugly remains, which bear witness to a high degree of decadence.

Even with computers this process is too slow to be practical. Think of a computer that writes 10^6 digits per second.

- There are $10^{100} - 10^{99} = 10^{99} \times 9$ numbers with 100 digits.
- These numbers have a total of $10^{101} \times 9$ digits.
- One needs $10^{95} \times 9$ seconds to write these numbers, that is about 1.5×10^{94} minutes, that is about 25×10^{92} hours, so more than 10^{91} days, that is of the order of 3×10^{88} years, that is 3×10^{86} centuries!

And after writing the numbers (if there is still an After...) there is much more to be done!

Before the buyer complained, I added:

—I: There are shortcuts, but even then the method would still be too slow. So, instead of trying to list the primes with 100 digits, our factory uses fast algorithms to produce enough primes to cover our orders.

—Buyer: I am amazed. I never thought how important it is to have a fast method. Can you tell me the procedure used in your factory? I am really curious. [Yes, this buyer was being too nosy. Now I became convinced that he was a spy.]

—I: When you buy a Mercedes, you don't ask how it was built. You choose your favorite color, pink, purple, or green with orange dots, you drive it and you are happy, because everyone else is envious of you.

Our factory will deliver the primes you ordered and we do better than Mercedes. We support our product with a lifetime guarantee. Goodbye, sir.

[He may have understood: Good buy, sir...]

After the Dialogue

I hope that after the dialogue with the spy-buyer, you became curious to know about our fast procedure to produce large primes. I shall tell you some of our most cherished secrets. In our factory there are two main divisions.

1) Production of primes.
2) Quality control.

Production of Primes

One of the bases of our production methods was discovered long ago by Pocklington [4]. I will state and prove his theorem, in the particular situation adapted to our production requirements. Then, I shall discuss how it may be used to obtain, in a surprisingly short time, primes with the required number of digits.

CRITERION OF POCKLINGTON. *Let p be an odd prime, let k be a natural number such that p does not divide k and $1 \leq k < 2(p+1)$; and let $N = 2kp + 1$. Then the following conditions are equivalent:*

1) *N is a prime.*

2) *There exists a natural number* a, $2 \le a < N$, *such that*

$$a^{kp} \equiv -1 \pmod{N}$$

and

$$\gcd(a^k + 1, N) = 1.$$

Proof. $1 \Rightarrow 2$. Assume that N is a prime. As it is known, there is some integer a, $1 < a < N$, such that $a^{N-1} \equiv 1 \pmod{N}$, but $a^m \not\equiv 1 \pmod{N}$ if $1 < m < N - 1$; such a number a is called a *primitive root modulo N*. Thus $a^{2kp} \equiv 1 \pmod{N}$, but $a^{kp} \not\equiv 1 \pmod{N}$; then $a^{kp} \equiv -1 \pmod{N}$. Also $a^k \not\equiv -1 \pmod{N}$ otherwise $a^{2k} \equiv 1 \pmod{N}$, which is not true; so $\gcd(a^k + 1, N) = 1$.

$2 \Rightarrow 1$. In order to show that N is a prime, we shall prove: If q is any prime dividing N, then $\sqrt{N} < q$. It follows that N cannot have two (equal or distinct) prime factors, so N is a prime.

So, let q be any prime factor of N. Then $a^{kp} \equiv -1 \pmod{q}$ and $a^{2k} \equiv 1 \pmod{q}$. Hence $\gcd(a, q) = 1$. Let e be the order of a modulo q, hence e divides $q - 1$, by Fermat's little theorem. Similarly, e divides $2kp = N - 1$, because $a^{2kp} \equiv 1 \pmod{q}$. Note that $a^k \not\equiv 1 \pmod{q}$, otherwise $a^{kp} \equiv 1 \pmod{q}$; from $a^{kp} \equiv -1 \pmod{q}$, it follows that $q = 2$ and N would be even, which is false.

From $\gcd(a^k + 1, N) = 1$, it follows that $a^k \not\equiv -1 \pmod{q}$. Hence $a^{2k} \not\equiv 1 \pmod{q}$, thus $e \nmid 2k = (N - 1)/p$. But $e \mid N - 1$, so $(N - 1)/e$ is an integer, hence $p \nmid (N - 1)/e$. Since $N - 1 = e\,(N - 1/e)$ and $p \mid N - 1$, then $p \mid e$, thus $p \mid q - 1$. Also $2 \mid q - 1$, hence $2p \mid q - 1$, so $2p \le q - 1$ and $2p + 1 \le q$. It follows that $N = 2kp + 1 < 2 \times 2(p + 1)p + 1 = 4p^2 + 4p + 1 = (2p + 1)^2 \le q^2$, therefore $\sqrt{N} < q$. This concludes the proof.

The criterion of Pocklington is applied as follows to obtain primes of a required size, say with 100 digits.

First step: Choose, for example, a prime p_1 with $d_1 = 5$ digits. Find $k_1 < 2(p_1 + 1)$ such that $p_2 = 2k_1p_1 + 1$ has $d_2 = 2d_1 = 10$ digits or $d_2 = 2d_1 - 1 = 9$ digits and there exists $a_1 < p_2$ satisfying the conditions $a_1^{k_1p_1} \equiv -1 \pmod{p_2}$ and $\gcd(a_1^{k_1} + 1, p_2) = 1$. By Pocklington's criterion, p_2 is a prime.

Subsequent steps: Repeat the same procedure starting with the prime p_2 to obtain the prime p_3, etc... In order to produce a prime with 100 digits, the process must be iterated five times. In the last step, k_5 should be chosen so that $2k_5p_5 + 1$ has 100 digits.

Feasibility of the Algorithm

Given p and k, with $1 \le k < 2(p + 1)$, k not a multiple of p, if $N = 2kp + 1$ is a prime, then it has a primitive root. It would be much too technical to explain in detail the following results, some known to experts, others still unpublished. It follows from a generalized form of the Riemann hypothesis, that if x is a large positive real number and the positive integer a is not a square, then the ratio

$$\frac{\#\{\text{primes } q \le x \text{ such that } a \text{ is a primitive root modulo } q\}}{\#\{\text{primes } q \le x\}}$$

converges; if a is a prime, the limit is at least equal to Artin's constant

$$\prod_{q \text{ prime}} \left(1 - \frac{1}{q(q-1)}\right) \approx 0.37.$$

Better, given positive integers, a, b, which are not squares and a large prime q, the probability that a or b is a primitive root modulo q, is much larger. Taking $a = 2$, $b = 3$, it is at least 58%. The corresponding probability increases substantially when taking three positive integers a, b, c that are not squares.

This suggests that we proceed as follows. Given the prime p, choose k, not a multiple of p, $1 \le k < 2(p+1)$. If $N = 2kp + 1$ is a prime, then very likely 2, 3, or 5 is a primitive root modulo N. If this is not the case, it is more practical to choose another integer k', like k, and investigate whether $N' = 2k'p + 1$ is a prime.

The question arises: What are the chances of finding k such that N is a prime? I now discuss this point.

1°) According to a special case of Dirichlet's famous theorem (see [5], [6]), given p, there exist infinitely many integers $k \ge 1$ such that $2kp + 1$ is a prime. This may be proved in elementary way.

2°) How small may k be, so that $2kp + 1$ is a prime? A special case of a deep theorem of Linnik asserts:
For every sufficiently large p, in the arithmetic progression with first term 1 and difference $2p$, there exists a prime $p_1 = 2kp + 1$ satisfying $p_1 \le (2p)^L$; here L is a positive constant, (that is, L is independent of p) (see [5]).

3°) Recently, Heath-Brown has shown that $L \le 5.5$.

4°) In Pocklington's criterion, it is required to find $k < 2(p+1)$ such that $p_1 = 2kp + 1$ is a prime. This implies that $p_1 < (2p+1)^2$. No known theorem guarantees that such small values of k lead to a prime.

5°) Recent work of Bombieri, Friedlander and Iwaniec deals with primes p for which there are small primes $p_1 = 2kp + 1$. Their results, which concern averages, point to the existence of a sizable proportion of primes p with small prime $p_1 = 2kp + 1$.

The problems considered above are of great difficulty. In practice, we may ignore these considerations and find, with a few trials, the appropriate value of k.

Estimated Time to Produce Primes with 100 Digits

The time required to perform an algorithm depends on the speed of the computer and on the number of bit operations (i.e., operations with digits) that are necessary.

As a basis for this discussion, we may assume that the computer performs 10^6 bit operations per second. If we estimate an upper bound for the number of bit operations, dividing by 10^6 gives an upper bound for the number of seconds required.

A closer look at the procedure shows that it consists of a succession of the following operations on natural numbers: multiplication ab modulo n, power a^b modulo n, calculation of greatest common divisor.

It is well known (see [1], [2]) and not difficult to show that for each of the above operations there exist $C > 0$ and an integer $e \ge 1$ such that the number of bit operations required to perform the calculation is at most Cd^e, where d is the maximum of the number of digits of the numbers involved. Combining these estimates gives an upper bound of the same form Cd^e *for the method* ($C > 0$, $e \ge 1$

and d is the maximum of the number of digits of all integers involved in the calculation).

It is not my purpose to give explicit values for C and e when p, k, a are given. Let me just say that C, e are rather small, so the algorithm runs very fast. I stress that in this estimate the time required in the search for k, a is not taken into account.

The above discussion makes clear that much more remains to be understood in the production of primes and the feasibility of the algorithm. This task is delegated to our company's division of research and development, and I admire our colleagues in the research subdivision who face the deep mysteries of prime numbers.

Before I rapidly tour our division of quality control, I would like to make a few brief comments about our preceding considerations. They concern the complexity of an algorithm.

An algorithm A, performed on natural numbers, is said to run in *polynomial time* if there exist positive integers C, e (depending on the algorithm) such that the number of bit operations (or equivalently, the time) required to perform the algorithm on natural numbers with at most d digits is at most Cd^e.

An algorithm that does not run in polynomial time is definitely too costly to implement and is rejected by our factory. It is one of the main subjects of research to design algorithms that run in polynomial time. The algorithm to produce primes of a given size, for all practical purposes, runs in polynomial time, even though this has not yet been supported by a proof.

Quality Control

The division of quality control in our factory watches that the primes we sell are indeed primes. When Pocklington's method is used we only need to worry if no silly calculation error was made, because it leads automatically to prime numbers. If other methods are used, as I shall soon invoke, there must be a control. The division of quality control also engages in consulting work. A large number N is presented, with the question: Is N a prime number?

Thus, our division of quality control also deals with tests of primality. Since this is a cash rewarding activity, there are now many available tests of primality. I may briefly classify them from the following three points of view:

1. Tests for generic numbers.
   Tests for numbers of special forms, like $F_n = 2^{2^n} + 1$ (Fermat numbers), $M_p = 2^p - 1$, (p prime, Mersenne numbers), etc....
2. Tests fully justified by theorems.
   Tests based on justification that depends on forms of Riemann's hypothesis of the zeros of the zeta function, or on heuristic arguments.
3. Deterministic tests.
4. Probabilistic or Monte Carlo tests.

A deterministic test applied to a number N will certify that N is a prime or that N is a composite numbers. A Monte Carlo test applied to N will certify either that N is composite, or that, with a large probability, N is a prime.

Before I proceed, let me state that the main problem tempting the researchers is the following: Will it be possible to find a fully justified and deterministic test of primality for generic numbers, which runs in polynomial time? Or will it be proven that there cannot exist a deterministic, fully justified test of primality that runs in polynomial time, when applied to any natural number?

This is a tantalizing and deep problem.

It would be long-winded and complex even to try to describe all the methods and algorithms used in primality testing. So, I shall concentrate only on the strong pseudoprime test, which is of Monte Carlo type.

Pseudoprimes Let N be a prime, let a be such that $1 < a < N$. By Fermat's little theorem, $a^{N-1} \equiv 1 (\text{mod } N)$.

However, the converse is not true. The smallest example is $N = 341 = 11 \times 31$, with $a = 2$, $2^{340} \equiv 1 (\text{mod } 341)$.

The number N is called a *pseudoprime in base* a, where $\gcd(a, N) = 1$, if N is composite and $a^{N-1} \equiv 1 (\text{mod } N)$. For each $a \geq 2$, there are infinitely many pseudoprimes in base a. Now observe that every odd prime N satisfies the following property:

> For any a, $2 \leq a < N$, with $gcd(a, N) = 1$, writing $N - 1$ in the form $N - 1 = 2^s d$ (where $1 \leq s$, d is odd), either $a^d \equiv 1 (\text{mod } N)$ or there exists r, $0 \leq r < s$, such that $a^{2^r d} \equiv -1 (\text{mod } N)$. (*)

Again, the converse is not true, as illustrated by $N = 2047 = 23 \times 89$, with $a = 2$.

The number N is called a *strong pseudoprime in base* a, where $\gcd(a, N) = 1$, if N is composite and the condition (*) is satisfied.

It has been shown by Pomerance, Selfridge, and Wagstaff that for every $a \geq 2$ there exist infinitely many strong pseudoprimes in base a.

The strong pseudoprime test The main steps in the strong pseudoprime test for a number N are the following:

1°) Choose $k > 1$ numbers a, $2 \leq a < N$, such that $\gcd(a, N) = 1$. This is easily done by trial division and does not require knowledge of the prime factors of N. If $\gcd(a, N) > 1$ for some a, $1 < a < N$, then N is composite.

2°) For each chosen base a, check if the condition (*) is satisfied.

If there is a such that (*) is not satisfied, then N is composite. Thus, if N is a prime, then (*) is satisfied for each base a. The events that condition (*) is satisfied for different bases may be legitimately considered as independent if the bases are randomly chosen.

Now, Rabin proved (see [5]): Let N be composite. Then the number of bases a for which N is a strong pseudoprime in base a is less than $\frac{1}{4}(N - 1)$. Thus, if N is composite, the probability that (*) is satisfied for k bases is at most $1/4^k$. Hence, certification that N is a prime when (*) is satisfied for k distinct bases is incorrect in only one out of 4^k numbers; for example, if $k = 30$, the certification is incorrect only once in every 10^{18} numbers.

The strong pseudoprime test runs in polynomial time and it is applicable to any number.

If a generalized form of Riemann's hypothesis is assumed to be true, Miller showed (see [5]): If N is composite, there exists a base a, with $\gcd(a, N) = 1$, such that $a < (\log N)^{2+\varepsilon}$, for which (*) is not satisfied.

A new production method We may use Rabin's test to produce numbers with 100 digits that may be certified to be prime numbers, with only very small probability of error.

1°) Pick a number N with 100 digits. Before doing any hard work, it is very easy, with trial division, to find out if this number does, or does not have, any prime

factor less than, say, 1000. In the latter case, keep this number; in the first case, discard N, pick another number N', and proceed similarly.

2°) Use $k = 30$ small numbers a, prime to N, as bases to verify if condition (*) is satisfied. Discard N if, for some base a, the condition (*) is not satisfied and repeat the process with some other number N'. If (*) is satisfied for all a then, according to Rabin's calculation, we may certify that N is prime; in doing so, we are incorrect only in at most one out of 10^{18} numbers. How unlucky can we be and choose, in succession, numbers which are composite? According to the inequalities of Chebyshev already indicated, the proportion of numbers with 100 digits that are prime is not less than $\frac{3.42}{9\times 10^2} \approx \frac{1}{260}$ and not more than $\frac{4.38}{9\times 10^9} \approx \frac{1}{205}$. Unintelligent employees who would pick even numbers, or numbers divisible by $3, 5, \ldots,$ or small primes (say up to 1000) are sure to be fired. Thus the luck of picking a prime increases and it becomes quite reasonable to use Rabin's method of production.

We may sell the number N as if it were a prime, even with a "money back guarantee" because the probability that we are selling a composite number is only 1 in every 1,000,000,000,000,000,000 sales! This is a better guarantee than anyone can get in any deal. We are sure that our company will not be bankrupt and will continue to support generously my trips to advertise our products—all complemented by lavish dinners and the finest wines, to help convince our customers that primes are the way of life.

Appendix

Here is an example of a prime with 100 digits that was calculated by L. Roberts, using the method of Pocklington.

$p_1 = 2333$ $\qquad k_1 = 2001$ $\qquad a_1 = 2$

$p_2 = 9336667$ $\qquad k_2 = 9336705$ $\qquad a_2 = 3$

$p_3 = 174347410924471$ $\qquad k_3 = 174347410924479$ $\qquad a_3 = 2$

$p_4 = 607940393921354891483080512 19$

$k_4 = 607940393921354891483080512 56$ $\qquad a_4 = 3$

$p_5 = 739183045122504318980574950295193587260702667372985056212 9$

$k_5 = 500000000000000000000000000000000000000137$ $\qquad a_5 = 2$

$p_6 = 739183045122504318980574950295193587260905203527348622396300677536$
$380883042909432530860197905402334 7$

Acknowledgement to Bart Braden for friendly style suggestions.

REFERENCES

1. Le Veque, W. J. (editor) *Studies in Number Theory*, MAA, Washington, 1975.
2. Mignotte, M. *Mathematics for Computer Algebra*, Springer-Verlag New York, 1991.
3. Plaisted, D. A. Fast verification, testing and generation of large primes, *Theor. Comp. Sci.* 9 (1979), 1–16.
4. Pocklington, H. C. The determination of the prime or composite nature of large numbers by Fermat's theorem, *Proc. Cambridge Phil. Soc.* 18 (1914/ 16), 29–30.
5. Ribenboim, P. *The Book of Prime Number Records*, 2nd edition, Springer-Verlag New York, 1989.
6. Ribenboim, P. *The Little Book of Big Primes*, Springer-Verlag New York, 1991.