Algorithmic Number Theory - The Complexity Contribution

Leonard M. Adleman*

Department of Computer Science
University of Southern California
Los Angeles, California 90089, USA

Abstract

Though algorithmic number theory is one of man's oldest intellectual pursuits, its current vitality is perhaps unrivaled in history. This is due in part to the injection of new ideas from computational complexity. In this paper, a brief history of the symbiotic relationship between number theory and complexity theory will be presented. In addition, some of the technical aspects underlying 'modern' methods of primality testing and factoring will be described . Finally, an extensive lists of open problems in algorithmic number theory will be provided.

1 Introduction

History of the two central problems: primality and factoring.[1]

It all began a little over 2000 years ago. That's when Eratosthenes of Alexandria invented the sieve (and yes - this is the same Eratosthenes who measured the circumference of the earth). The sieve was apparently the first recorded method for testing primality (and factoring - at least square free factoring). Unfortunately, things did not progress rapidly from there. It was only around 1200 that Fibonacci (a.k.a. Leonardo Pisano) improved the method - noting that one could stop sieving when one reached the square root of the number of interest. Remarkably, this remained the fastest method known for both factoring and primality testing until 1974 when R. Sherman Lehman demonstrated that factoring could be done in time $O(n^{1/3})$. After Fibonacci, the problems began to attract the interest of mathematical heavy weights including Fermat, Euler, Legendre and most importantly Gauss. In 1801 Gauss published *Disquisitiones arithmeticae.* He was intensely interested in primality testing and factoring as the following (now famous) quote shows:

> *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. It has engaged the industry and wisdom of ancient and modern geometers to such an extent that it would be superfluous to discuss the problem at length.... Further, 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.*
> - Karl Freidrich Gauss.
> *Disquisitiones Arithmeticae* (1801)
> (translation: A. A. Clarke [Gau86])

Gauss was the first (that I know of) to realize that primality and factoring were 2 different problems and that one could hope to prove the compositness of a number without factoring it. Gauss gave two algorithms which he said would 'prove rewarding to lovers of arithmetic'[2] Gauss' algorithms are indeed rewarding if you happen to like incredibly brilliant ideas; however, you may feel less rewarded if you look at their complexity: strictly exponential. Unfortunately for algorithmic number theory, Gauss was too good and he created a schism which to this day is not fully repaired. In *Disquisitiones arithmeticae*, Gauss laid the foundations for the development of number theory as 'pure math'. He was the first to use algebraic extensions of the rationals. His proof of the quadratic reciprocity law spurred the search for generalizations, ultimately leading to the development of class field theory. He (and Euler, Legendre, Dirichlet and Riemann) set analytic number theory going. The problem was that everyone ran off to follow these various paths and computational number theory suffered. In addition, perhaps because Gauss had used his computational ability to build large tables of primes and this

*Research supported by the National Science Foundation under grant CCR-9214671. email: adleman@cs.usc.edu.

[1] This is not intended to be a scholarly historical account; rather, it records the impressions which have guided me. Extensive references can be found in the open problems section. For more on the history of these problems, see for example [Dic71],[WS91].

[2] These lines are followed by: 'It is in the nature of the problem that *any* method will become more prolix as the numbers get larger. Nevertheless, in the following methods the difficulty increases rather slowly,...". These and other comments from *Disquisitiones arithmeticae* lead me to suspect that if Gauss had had a free weekend sometime to think about all this, complexity theory would have begun a century and a half earlier than it did.

led to his (and Legendre's) 'prime number conjecture' (later Vallée Poussin/Hadamard theorem), it led researchers who followed to view computational number theory as an instrument for experimentation -a microscope into the world of numbers which allowed for the development and testing of conjectures. What was lost for over one hundred years was what Gauss apparently understood - that primality testing and factoring were intrinsically beautiful problems and that that alone was reason to study them.

In the 1930's the second major thread in the story of algorithmic number theory began to develop - in the form of mathematical logic and recursive function theory. In the hands of Gödel, Turing, Kleene, and Church, a new aesthetic began to emerge: algorithmic aspects of problems were intrinsically interesting - whether or not you ever did any computation (one can't program up the undecidability of arithmetic). This point of view can be seen in the well known letter which Gödel sent to Von Neumann. In addition, to foreshadowing the NP=P question, Gödel also wrote:

> *It would be interesting to know, for example, what the situation is with the determination if a number is a prime, and in general how much we can reduce the number of steps from the method of simply trying(,) for finite combinatorial problems*
> - Kurt Gödel
> Letter to Von Neumann, 1956
> (translation: J. Hartmanis [Har])

This kind of thinking, led to the development of our field. In the late 1960's and early 1970's, the very gifted first generation of complexity theorists (naming names here would be too risky) put the central notions of polynomial time, NP and the rest in place.

It is in this setting that what I consider to be a renaissance in algorithmic number theory began. I should reiterate that as of 1973 the fastest factoring algorithm known and the fastest primality algorithm know were still due to Fibonacci [3]. The rapid changes that followed, particularly in primality testing, are not attributable (at least not directly) to the advent of the electronic computer nor to any major breakthroughs in number theory but rather to the acceptance of the new aesthetic that algorithmic problems are interesting in their own right and that the correct way to proceed was by 'counting steps'.

This new point of view was powerful. It said that you could tell whether an algorithm was good or bad - not by trying a benchmark factorization or primality test - but by analyzing the algorithm's complexity. To some extent this was a license to steal. Young researchers in the field like Gary Miller, Vaughan Pratt, and myself - could go to the number theorists, look at their very clever algorithms, then 'cut and paste' their ideas together into new algorithms which to them often seemed useless because they did not factor numbers or test primality very well, but to us with our new aesthetic seemed important.

The first example of this was Pratt's 1974[4] proof (based on work of Lucas nearly one hundred years earlier) that the primes were in NP. This is a perfect example of a non-result - from the number theorist's perspective. They viewed it as entirely obvious [5] and of course no one could claim it was useful. However, in the new complexity theoretic aesthetic it was an important result. Pratt had culled an idea from the many current in the literature and 'said' that this one idea must be important since it allowed for a better complexity result than was known before. We will see shortly, how correct he was. But it is clear that without complexity theory there was no result at all.

Pratt's result was followed by a string of others, all made possible by the complexity theoretic point of view (and the hundreds of years of deep ideas by the number theorists). In 1975, Complexity theory received its first dividend. Solovay and Strassen (and later Miller and Rabin) showed that the composites were in R (random polynomial time). While Gill had begun a theoretical study of the use of randomness in computation a few year prior, it was this result which made randomness a central feature in complexity theory. So the ancient problem still had something new to give.

This result was then followed by what I think of as one of the most beautiful in the field - Miller's 1975 proof that on extended Riemann hypothesis primality is in P. This landmark result also marked the beginning of a new trend - the use of deep, modern mathematics in algorithmic number theory. This trend has continued in the almost 20 years since.

Next came the second dividend for complexity theory - the RSA public key cryptosystem in 1977. After 2000 years we could finally see that primality was 'easy' and factoring was 'hard' - and likely to stay that way for a long time. This led to the development of one of the primary subdisciplines of theoretical computer science. It also took algorithmic number theory out of its 2000 plus years of isolation and made it into the stuff of popular culture:

> *"I'm really not into computers, Jay. I don't know much. I do know the key to the code*

[3] It is important to make clear that despite this fact, in the years between Fibonacci's era and 1973, number theorists had produced many important ideas. For example, in primality testing, Lucas and Lehmer had developed important ideas for testing numbers of special form like Mersenne numbers. In factoring Gauss, Legendre, Seelhoff, Kraïchik, Lehmer, and Powers were coming to understand the value of 'smoothness' in factoring.

[4] The years given for this and other discoveries are not necessarily those of the publication of the relevant papers, but my own best recollections of when the discoveries actually occurred

[5] This is not speculation, I have discussed this and related issues with number theorists on numerous occasions.

was the product of two long prime numbers, each about a hundred digits, right?"

"Yes, that's correct. It's called the RSA cryptosystem."

"Right, for Rivest, Shamir and Adleman from MIT. That much I know. I also understand that even using a sophisticated computer to decipher the code it would take forever," she recalled. "Something like three point eight billion years for a two-hundred-digit key, right?"

"That's exactly correct. All of the stolen information was apparently tapped from the phone lines running from the company offices to your house. Supposedly no one except Mike had the decoding key, and no one could figure it out unless he passed it along, but there has to be a bug in that logic somewhere" he said, loosening his dark green silk tie. "Vee, it's much warmer than I thought. Would you mind if I removed my jacket?"

"Of course not. You're so formal," she remarked ....
- Harlequin American Romance
Sunward Journey by Katherine Coffaro.

Three years later in 1980 came the result of Adleman, Pomerance and Rumely: a 'near polynomial time' deterministic primality test. There exists an $c \in N_{>0}$ such that the actual running time is:

$$O((\log n)^{c \log \log \log n})$$

This result is also of interest sociologically since the authors collectively represented the experimental, the complexity theoretical and the 'pure' mathematical traditions - which in my mind bodes well for the eventual healing of the schisms which have existed. It is also worth remarking that when implemented (for example by Cohen and Lenstra) this algorithm actually has the pleasant added feature of being the fastest primality testing algorithm in practice.

Next came the lovely 1986 result of Goldwasser and Kilian (only a little over a decade after Pratt). They resurrected the idea of Pratt and mixed it with Hendrik Lenstra and René Schoof's new idea of using elliptic curves. Goldwasser and Kilian proved that assuming an old conjecture due to Cramér, primality was in R. This was followed by the result of Adleman and Huang who replaced the elliptic curves with the Jacobian's of hyperelliptic curves and proved that, without hypothesis, primality was in R. Some technical aspects of this development are informally described in the next section. To the extent one accepts randomness in computation, this completes the long quest for an efficient primality test. To the extend that one insists on determinism, the search goes on.

I have said much less about developments in factoring during this period. That's because the recent progress here has not been as pronounced. Perhaps this is because the problem is intractable and not much progress can be expected. Most of the progress that has occurred has been through consolidation of ideas going back to Gauss and Legendre. Primarily through the efforts of Seelhoff, Kraïchik, Lehmer, Powers and finally Morrison and Brillhart in 1975, it became clear that the single really important idea in factoring was 'smoothness'. Recall that a number is 'smooth' iff all of its prime factors are 'small'. This is the fundamental insight that 2000 years of effort brought forth. It led to the first subexponential factoring algorithm and recently has been exploited by Pollard and a host of other researchers to produce the so called 'number field sieve' - the fastest method currently known. I will describe some of the technical aspects of these developments in the following section.

In the past year a very interesting development has occurred which suggests that the symbiosis between number theory and complexity theory is not finished. And that Physics may also have a part to play. Peter Shor has produced a striking result: factoring can be done in polynomial time on a quantum Turing machine. This result in some ways resembles that of Solovay and Strassen almost 20 years before: an interesting theoretical notion made tangible by application to a major problem in number theory. The result is new and needs further analysis, but assuming that everything (including the physics) holds together, it suggests that primality and factoring may play an even greater role in the future than they have in the past.

In closing, let me say that I have concentrated here on primality and factoring, but the field is much larger than that. The last section of this paper contains an extensive 'open problems' list.

2 Primality Testing and Factoring [6]

Below are two algorithms which will prove rewarding to lovers of arithmetic.

2.1 How to Test Primality

In this subsection I want give an informal description of the ideas that led from Pratt to Goldwasser and Kilian to Adleman and Huang and resulted in the proof that Primality is in R (random polynomial time).

Consider the following proof that 11 is prime:

[6] The open problems section contains a detailed account of the state of these problems along with many references.

(1) $(4-1, 11) = 1$.
(4 does not have order 1)

(2) $4^5 = 1024 \equiv 1 \bmod 11$.
(the order of 4 divides 5)

(3) 5 is prime.
(the order of 4 is exactly 5)

Assume 11 is not prime. Then there exists a prime $p \leq \sqrt{11} < 4$ which divides 11. (1) implies that the order of 4 in Z/pZ^* is not 1. (2) implies that the order of 4 in Z/pZ^* divides 5. (3) implies that the order of 4 in Z/pZ^* is exactly 5 which is impossible since $p < 4$.

If we had omitted (3) above, we would still have produced a proof that *if* 5 *is prime then* 11 *is prime.* Such a proof can be though of as a *reduction* of the proof of primality of 11 to the proof of primality of 5. This idea of reducing the proof of primality of one number to that of another number is central to the algorithms discussed here. The seeds of this idea can be found in Pratt.

For all sufficiently large primes p, is there always a reduction of the proof of primality of p to the proof of primality of $\frac{p-1}{2}$? The answer is yes. Can a computer generate such a reduction quickly (i.e. in random polynomial time)? Again the answer is yes. One (not particularly efficient) way for the computer to proceed is as follows:

(1) Pick $a \in Z_{>0}^{<p}$ at random until an a is found such that $(a-1, p) = 1$ and:

(2) $a^{\frac{p-1}{2}} \equiv 1 \bmod p$.

If such an a is found, then, as above, a reduction of the proof of primality of p to the proof of primality of $\frac{p-1}{2}$ is easily generated. Since GCD and modular exponentiation can be done in deterministic polynomial time and since for sufficiently large p, satisfactory a's are 'abundant' (i.e. for all primes $p > 2$, $\frac{p-3}{2}$ of the positive integers $a < p$ are such that $(a-1, p) = 1$ and $a^{\frac{p-1}{2}} \equiv 1 \bmod p$), we can expect that when p is sufficiently large (in this case $p > 3$ will do) the computer will quickly choose an appropriate a and hence generate the desired reduction.

Returning to the example above, we have a reduction of the proof of primality of 11 to the proof of primality of 5. We could now recurse and obtain a reduction of the proof of primality of 5 to the proof of primality of 2. Assuming that we have a direct proof that 2 is prime, we then have a proof that 11 (and also 5) is prime. Can recursing with this type of reduction yield a random polynomial time algorithm for proving primality? The answer is no, because for most primes p, the order of Z/pZ^* is not 2 times a prime. Reducing the proof of primality of 13 to the proof of primality of 6 works, but is not very useful.

Goldwasser and Kilian realized that there was a way to circumvent this problem by using groups other than Z/pZ^*. Their idea had been foreshadowed in the work of Lenstra on integer factoring and their method makes use of elliptic curves. A computer can generate a Goldwasser-Kilian type reduction (essentially) as follows:

(0) Pick $a, b \in Z/pZ$ at random until a pair is found such that $f(x) = x^3 - ax + b$ has no multiple roots and such that the number n of rational points on the elliptic curve associated with $y^2 - f(x)$ is even (such can be done in random polynomial time, including (thanks to Schoof) the calculation of n).

(1) Compute a pair $s, t \in Z/pZ$ such that $v = < s, t >$ is on the curve.

(v does not have order 1)

(This is because an affine point cannot be the identity in the group of rational points on the elliptic curve.)

(2) $n/2 * v$ is the identity in the group associated with the curve.

(the order of v is divisible by $n/2$)

Modifying the arguments above, it is possible to show that for all sufficiently large primes p, if such a point v is found, then a reduction of the proof of primality of p to the proof of primality of $m = n/2$ has been generated. Further, from the Riemann hypothesis for finite fields (as proved by Weil) m will always be approximately half as big as p. However, like the previous reduction, a Goldwasser-Kilian type reduction is useful only if m is prime.

Can recursing with Goldwasser-Kilian type reductions yield a random polynomial time algorithm for primality testing? For this to be the case, it would be necessary that for all sufficiently large primes p, the number of rational points on a randomly chosen elliptic curve defined over Z/pZ would be 2 times a prime with high probability. By the Riemann hypothesis for finite fields, the number n of rational points on an elliptic curve defined over Z/pZ must always be between $p + 1 - 2\sqrt{p}$ and $p + 1 + 2\sqrt{p}$. Unfortunately, this interval is too small. Current knowledge does not allow us to rule out the possibility that there are infinitely many primes p for which there are <u>no</u> n's in the interval of the correct form. For such primes, there would be no 'useful' Goldwasser-Kilian type reductions at all. For this reason whether the Goldwasser-Kilian method yields a random polynomial time test for primality remains an open question.

Goldwasser and Kilian (using estimates of Heath-Brown) did succeed in demonstrating that their method 'works' for 'most' primes. That is, except for a small (but perhaps infinite) set of 'bad' primes, successive Goldwasser-Kilian type reductions will in polynomial time with high probability produce a proof of primality.

The Adleman Huang method begins by replacing the elliptic curves in the Goldwasser-Kilian approach by the Jacobians of curves of genus 2. A computer can in random polynomial time generate a reduction of the new type (essentially) as follows:

(0) Pick $f \in Z/pZ[x]$ of degree 6 at random until one is found such that f has no multiple roots. Calculate the number n of rational points on the Jacobian of the curve associated with $y^2 - f(x)$.

(1) Compute a pair $s, t \in Z/pZ$ with $t \neq 0$ such that $< s, t >$ is on the curve and let $v = \phi(< s, t >) - \phi(< s, -t >)$ (where ϕ is an embedding of the curve into the Jacobian)

(v does not have order 1)

(By construction, v cannot be the identity in the group of rational points on the Jacobian of the curve.)

(1) $n * v$ is the identity in the group of rational points on the Jacobian of the curve.)

(the order of v is divisible by n)

($n/2$ is no longer important in this variation.)

Again by a modification of earlier arguments, it is possible to show that for all sufficiently large primes p, if such a point v is found then a reduction of the proof of primality of p to the proof of primality of n is easily generated. However, as before, the reduction is useful only if n is prime.

The new method carries with it some 'good news' and some 'bad news'.

The good news: with 'high probability' n will be prime. That is: there is a constant $c \in Z_{>0}$ such that for sufficiently large primes p, at least $1/\log^c p$ of the f's chosen as above give rise to a Jacobian with a prime number of points.

It follows from this good news that there exists a random polynomial time algorithm for reducing the primality of p to a new prime n.

The bad news: n will always be <u>much larger</u> than p. Thus the new 'reductions' actually make the problem harder! Again by Weil's Riemann hypothesis for finite fields n must always be between $p^2 - 4p^{1.5} + 6p - 4p^{.5} + 1$ and $p^2 + 4p^{1.5} + 6p + 4p^{.5} + 1$. Hence n will be approximately p^2.

Fortunately there is a way to overcome the bad news. If we are trying to prove the primality of p and we have successfully produced a sequence of reductions of the new type (a sequence of 3 reductions is enough) then we have reduced the proof of primality of p to the proof of primality of a new (much larger) prime q. Since the new method uses randomness in selecting curves, there are in fact many different primes q which might arise. But the (generalized) Goldwasser-Kilian method produces primality proofs for all primes with a small set of exceptions. It turns out that with 'high probability', the q's which the new method produces can be proven to be prime using the (generalized) Goldwasser-Kilian method. Thus by combining the two methods a proof that p is prime will be obtained.

Put still less formally, the problem with the (generalized or ungeneralized) Goldwasser-Kilian method was that if p was one of the rare 'bad' primes, then there is no way to produce a proof of primality. The new method lets one 'randomize'. That is, one can use the new method to reduce the proof of primality for p to the proof of primality of a new prime q which is, roughly speaking, chosen at random. Consequently q is unlikely to be 'bad' and so the (generalized) Goldwasser-Kilian method can be used to prove the primality of q thus completing the proof of primality of p.

2.2 How to Factor an Integer

In this subsection I will give an informal description of how an old idea - smoothness - yields the fastest current factoring algorithm: the 'number field sieve'.

To begin with a number is smooth iff all of its prime factors are 'small'. $6960096 = 2^5 * 3^2 * 11 * 13^2$ is smooth. $69600970 = 2 * 5 * 6047 * 1151$ is not (6047 and 1151 are prime). The idea of using smoothness in factoring appears to be due to Gauss and Legendre. In *Disquisitiones arithmeticae* Gauss has the following example of applying his method to factoring 997331. Basically, he squares numbers (near $\sqrt{997331}$) and takes their residues mod 997331. He keeps only those with smooth residues:

$999^2 \equiv 2 * 5 * 67$
$994^2 \equiv -5 * 11 * 13^2$
$706^2 \equiv -2^{-1} * 3^3 * 17$
$575^2 \equiv -3^{-1} * 4^2 * 11 * 31$
$577^2 \equiv 3^{-1} * 4^2 * 7 * 13$
$578^2 \equiv 3^{-1} * 7 * 19 * 37$
$299^2 \equiv -2 * 3 * 4^2 * 5 * 11^{-1} * 29$
$301^2 \equiv -2^4 * 3^2 * 5 * 11^{-1}$

He then combines these congruences (implicitly using **Gaussian** elimination mod 2 on the exponents) to find small quadratic residues mod 997331:

For example $994^2 * 299^2 \equiv 2*3*4^2*5^2*13^2*29$ so that $174 = 2*3*29$ is a quadratic residue mod 997331.

He then uses these small quadratic residues to restrict potential divisors and it is here that he has apparently made his first and only wrong move. What he apparently should have done, as about one and three-quarters centuries of research by Seelhoff, Kraïchik, Lehmer, Powers, Morrison ,Brillhart and others would finally reveal, was to use Gaussian elimination to obtain a congruence of squares. For example:

$994^2 * 301^2 \equiv 2^4 * 3^2 * 5^2 * 13^2$

or in other words:

$(994 * 301)^2 \equiv (4*3*5*13)^2$

Now obtaining such congruences is tantamount to factoring (as Gauss and Legendre knew, by the way). One now takes $\gcd(997331, 994*301 - 4*3*5*13) = 7853$. This is a non trivial prime factor of 997331 (the other one is 127). So the algorithm became:

1. Input n.
2. Choose random $r \in Z_{>1}^{<n}$.
3. Calculate $r^2 \bmod n = a$, if a is smooth - i.e. all prime factors are less than an appropriate bound B keep it.
4. Once $B+1$ such r have been kept, use Gaussian elimination mod 2 on the exponents to build a congruence of squares $x^2 \equiv y^2 \bmod n$
5. Calculate $\mathrm{GCD}(x-y, n)$, it will be a nontrivial factor of n about half the time (or more).

One easily argues heuristically (a rigorous proof requires more work) that there exists a $\delta \in \Re_{>0}$ such the algorithm has expected running time:

$$L_n[1/2, \delta + o(1)]$$

Where for all $n, \gamma, \delta \in \Re_{>0}$, with $n > e$

$$L_n[\gamma, \delta] = exp(\delta(\log n)^{\gamma}(\log\log n)^{1-\gamma})$$

Making this argument uses the following central fact about smooth numbers:

Theorem [Canfield, Erdös and Pomerance] For all $n, \alpha, \beta, \gamma, \delta \in \Re_{>0}$ with $\alpha > \gamma$ and $n > e$, the number of $L[\gamma, \delta + o(1)]$ smooth numbers less than $L[\alpha, \beta + o(1)]$ is

$$n/L[\alpha - \gamma, (\beta(\alpha - \gamma)/\delta) + o(1)]$$

For example, the above theorem implies that when $B = L_n[1/2, 1/\sqrt(2) + o(1)]$, the probability that a numbers less than n is B-smooth is essentially $1/B$.

So the effort since that time has been to use smoothness more efficiently. One can get a better factoring algorithm by improving one of the following:

- The function implicit in the $o(1)$ in the exponent - this is not interesting theoretically but can make actual implementations faster.
- The constant δ which is of limited theoretical interest but has a significant impact on actual implementations.
- The constant γ which is of interest theoretically and practically.
- The big prize - the shape of the function itself - e.g. to find a polynomial time algorithm.

Four has not been done. If a polynomial time algorithm exists, I suspect it will need a better idea than smoothness. However, recently Pollard has come up with an idea for improving γ. Many researchers have contributed and the resulting algorithm, 'the number field sieve', has (heuristic) expected running time:

$$L_n[1/3, \delta + o(1)]$$

Where the best currently known δ is approximately 1.90 due to Coppersmith.

Here is how the number field sieve works. A positive integer n is given. One finds an irreducible polynomial $f \in Z[y]$ of degree d and an $m \in Z$ such that $f(m) \equiv 0 \bmod n$. Basically , this is done by choosing an m of appropriate size at random and then defining the coefficients of f to be the digits of n when written base m. Then working simultaneously in Q and the number field $L = Q[y]/(f)$, one seeks 'double smooth pairs' $< r, s > \in Z \times Z$ such that r and s are relatively prime and both $rm + s$ and $N_Q^L(ry + s) = r^d f(-s/r)$ are 'smooth' in Z (i.e. have no prime factors greater than some predetermined 'smoothness bound' B). In the simplest case where O_L, the ring of integers of L, is a PID then from such a double smooth pair we have:

$$rm + s = \prod_{i=1}^{z} p_i^{e_i}$$

where $p_1, p_2, ..., p_z$ are the primes less than B. And:

$$ry + s = \prod_{i=1}^{v} \epsilon_i^{f_i} \prod_{i=1}^{w} \wp_j^{g_j}$$

where $\epsilon_1, \epsilon_2, ...\epsilon_v$ is a basis for the units of O_L and $\wp_1, \wp_2, ..., \wp_w \in O_L$ are generators of the (residue class degree one) prime ideals of norm less than B.

Since there is a homomorphism ϕ from O_L to $Z/(n)$ induced by sending $y \mapsto m$, the above equations give rise to the congruence:

$$\prod_{i=1}^{z} p_i^{e_i} \equiv \prod_{i=1}^{v} \phi(\epsilon_i)^{f_i} \prod_{i=1}^{w} \phi(\wp_j)^{g_j}$$

One collects sufficiently many such double smooth pairs and puts the corresponding congruences together using linear algebra on the exponents to create a congruence of squares of the form:

$$(\prod_{i=1}^{z} p_i^{e_i'})^2 \equiv (\prod_{i=1}^{v} \phi(\epsilon_i)^{f_i'} \prod_{i=1}^{w} \phi(\wp_j)^{g_j'})^2$$

One now proceeds as before.

Of course when O_L is not a PID complications arise. But these can be overcome rather efficiently and the resulting algorithm is the most efficient (both in theory and in practice) currently known.

3 Open Problems in Algorithmic Complexity

For the last decade or so, Kevin McCurley and I have periodically maintained, revised and published [AM86, AMar] a collection of open problems in algorithmic number theory. McCurley deserves equal credit for what follows. I thank him for generously allowing me to publish this version under my own authorship.

The problems presented here arose from many different places and times. To those whose research has generated these problems or has contributed to our present understanding of them but to whom inadequate credit has been given, we apologize.

We expect that none of the problems presented here are easy - we are convinced that many are very hard. It is likely that some of the problems will remain open for the foreseeable future. However, it may be possible in some cases to make progress by solving subproblems, or by establishing reductions between problems, or by settling problems under the assumption of one or more well known hypotheses (e.g. the various extended Riemann hypotheses, $\mathcal{NP} \neq \mathcal{P}, \mathcal{NP} \neq \text{co}\mathcal{NP}$).

For the sake of clarity we have often chosen to state a specific version of a problem rather than a general one. For example, questions about the integers modulo a prime often have natural generalizations to arbitrary finite fields, to arbitrary cyclic groups, or to problems with a composite modulus. Questions about the integers often have natural generalizations to the ring of integers in an algebraic number field, and questions about elliptic curves often generalize to arbitrary curves or Abelian varieties.

The earliest version of these open problems appeared in 1986 [AM86]. At that time, we feared that the problems presented were so difficult that young researchers reading the list might be discouraged rather than inspired. Happily, despite the difficulties, eight years has brought considerable progress on a number of these problems. Even for the two most central problems in the field, primality testing and factoring, there has been impressive progress: the primes are now known to be decidable in random polynomial time and the 'number field sieve' has given us the most powerful factoring algorithm yet. To emphasize the progress that has been made, the statement of each problem is followed by the original 1986 remarks and then the remarks which now seem appropriate.

Your comments would be appreciated, particularly with regard to further progress on these problems.

Definitions, notation, and conventions.

In this paper:

- $\mathbb{R}$ denotes the set of real numbers,
- $\mathbb{Z}$ denotes the set of integers,
- $\mathbb{N}$ denotes the set of positive integers,
- *Primes* denotes the set of primes in $\mathbb{N}$,
- *Squarefrees* denotes the set of squarefree numbers in $\mathbb{N}$,
- $\mathbb{Q}$ denotes the set of rationals.
- ERH refers to the extended Riemann hypothesis.

For $a, b \in \mathbb{Z}$,

- we write $a \mid b$ if there exists $k \in \mathbb{Z}$ with $b = ka$,
- we write $a \nmid b$ if there does not exist $k \in \mathbb{Z}$ with $b = ka$,
- $\gcd(a, b)$ denotes the greatest common divisor of a and b,
- $(\frac{a}{b})$ denotes the Jacobi symbol if b is odd and $\gcd(a, b) = 1$,
- $\langle a, b \rangle$ denotes the ordered pair.

For $n \in \mathbb{N}$,

- $\mathbb{Z}/n\mathbb{Z}$ denotes the ring of integers modulo n,
- $(\mathbb{Z}/n\mathbb{Z})^*$ denotes the corresponding multiplicative group,
- $\phi(n)$ denotes the number of elements in $(\mathbb{Z}/n\mathbb{Z})^*$,
- $L(n)$ represents any function of the form

$$\exp((1+o(1))(\log n \log\log n)^{1/2}) \ .$$

- For $n \in \mathbb{N}$, $\alpha, \beta \in \mathbb{R}$, $\alpha, \beta > 0$, $L_n[\alpha, \beta]$ represents any function of the form

$$\exp((\beta + o(1))((\log n)^{\alpha}(\log\log n)^{1-\alpha})) \ .$$

If R is a ring, then we write $R[x]$ for the ring of polynomials with coefficients in R. The set of finite strings composed of the letters a and b is denoted $\{a,b\}^*$. For n, a, $b \in \mathbb{N}$ with $\gcd(n, 4a^3 + 27b^2) = 1$, let

$$S_{n,a,b} =$$
$$\{\langle x,y\rangle \mid x,y \in \mathbb{Z}/n\mathbb{Z} \ \&\ y^2 \equiv x^3 + ax + b \pmod{n}\} \cup \{0\}$$

When $p \in \mathit{Primes}$, $S_{p,a,b}$ is well known to be endowed with a group structure. We denote this group by $E_{p,a,b}$ and use $\#E_{p,a,b}$ for the number of elements of this group. More generally, if S is a set, we write $\#S$ for the cardinality of S.

In stating open problems we have decided to continue the ad hoc notation from [AM86]. For example, we label the first computational problem as **C1**, the corresponding open problem as **O1** (or **O1a** and **O1b** if there are two), and the original 1986 remarks concerning **C1** and **O1** we label as **Rem1**$_{86}$. Any new remarks we label as **Rem1**$_{94}$. Any additional references are given in **Ref1**. Computational problems **C2** and **C6** are stated in terms of a parameter S which is an arbitrary subset of $\mathbb{N}$. Computational problem **C30** is stated in terms of a parameter $c \in \mathbb{N}$.

While it seems inappropriate to spend a great deal of time giving rigorous definitions of the complexity-theoretic notions used in this paper, it seems worthwhile to provide some guidance in anticipation of this paper falling into the hands of researchers in areas other than theoretical computer science. More information on these notions may be found in [Gil77], [AHU74], [AM77], and [GJ79]. We assume the concept of a polynomial time computable function is understood. A computational problem **C** is thought of as a set of pairs $\langle x, S_x\rangle$, where x is an input for which an output is desired and S_x is the set of possible 'correct' outputs on input x. For example

$$\mathbf{C1} =$$
$$\{\langle n, S_n\rangle \mid n \in \mathit{Primes} \Rightarrow S_n = \{1\} \ \&\ n \notin \mathit{Primes} \Rightarrow S_n = \{0\}\}$$

$$\mathbf{C17} =$$
$$\{\langle\langle p,d\rangle, S_{\langle p,d\rangle}\rangle \mid d \in \mathbb{N} \ \&\ p \in \mathit{Primes} \ \& S_{\langle p,d\rangle} =$$
$$\{f \mid f \in (\mathbb{Z}/p\mathbb{Z})[x] \mid \deg(f) = d \ \&\ f \text{ irreducible}\}\}$$

$$\mathbf{C19} =$$
$$\{\langle p, S_p\rangle \mid p \in \mathit{Primes} \ \&\ S_p =$$
$$\{g \mid g \in \mathbb{N}, 1 \le g \le p-1 \ \& g \text{ generates } (\mathbb{Z}/p\mathbb{Z})^*\}\}$$

$$\mathbf{C28} =$$
$$\{\langle\langle a,b,p,P,Q\rangle, S_{\langle a,b,p,P,Q\rangle}\rangle \mid a,b \in \mathbb{N}, p \in \mathit{Primes}, P,Q \in E_{p,a,b}$$
$$\& (\exists n \in \mathbb{N})[nP = Q] \ \&\ S_{\langle a,b,p,P,Q\rangle} = \{n \mid n \in \mathbb{N} \ \&\ nP = Q\}\}$$

Definition 1 *If* $\mathbf{C} = \{\langle x, S_x\rangle\}$ *is a computational problem then we let* $\pi(\mathbf{C}) = \{x \mid \langle x, S_x\rangle \in \mathbf{C}\}$.

We use $|x|$ to denote the length of an object x, where we hope that the meaning of 'length' will be clear from the context.

Definition 2 **C** *is in* $\mathcal{P}$ *iff there exists a polynomial time computable function* f *such that* $(\forall x \in \pi(\mathbf{C}))[f(x) \in S_x]$.

Thus for example, in **O18** below we ask if **C18** is in $\mathcal{P}$. Any deterministic algorithm which runs in polynomial time with input-output behavior consistent with that described in **C18** would provide an affirmative answer to **O18**. In particular how that algorithm behaves on an input $p \notin \mathit{Primes}$ is irrelevant.

Definition 3 **C** *is in* $\mathcal{R}$ *iff there exists a* c *in* $\mathbb{N}$ *and a polynomial time computable function* f *such that*

i. $(\forall x \in \pi(\mathbf{C}))(\forall |r| \le |x|^c)$

$$[f(x,r) \in S_x \text{ or } f(x,r) = \text{"?"}]$$

ii. $(\forall x \in \pi(\mathbf{C}))$

$$\left[\frac{\#\{r \mid |r| \le |x|^c \ \&\ f(x,r) \in S_x\}}{\#\{r \mid |r| \le |x|^c\}} \ge \frac{1}{2}\right]$$

Definition 4 **C** *is in* $\mathcal{NP}$ *iff there exists a* c *in* $\mathbb{N}$ *and a polynomial time computable function* f *such that*

i. $(\forall x \in \pi(\mathbf{C}))(\forall |r| \le |x|^c)$

$$[f(x,r) \in S_x \text{ or } f(x,r) = \text{"?"}].$$

ii. $(\forall x \in \pi(\mathbf{C}))(\exists y \in S_x)(\exists |r| \le |x|^c)$

$$[f(x,r) = y].$$

Definition 5 **C** *is recognized in* $\mathcal{R}$ *iff*

i. $(\forall x \in \pi(\mathbf{C}))[S_x = \{1\} \Rightarrow$

$$(\forall |r| \le |x|^c)[f(x,r) = \{1\} \text{ or } f(x,r) = \text{"?"}]]$$

ii. $(\forall x \in \pi(\mathbf{C}))[S_x = \{1\} \Rightarrow$

$$\frac{\#\{r \mid |r| \le |x|^c \ \& \ f(x,r) = 1\}}{\#\{r \mid |r| \le |x|^c\}} \ge \frac{1}{2}]$$

iii. $(\forall x \in \pi(\mathbf{C}))[S_x \neq \{1\} \Rightarrow$

$$(\forall |r| \le |x|^c)[f(x,r) = "?"]].$$

Definition 6 **C** *is recognized in* $\mathcal{NP}$ *iff there exists a* c *in* **N** *and a polynomial time computable function* f *such that*

i. $(\forall x \in \pi(\mathbf{C}))[S_x = \{1\} \Rightarrow$

$$(\forall |r| \le |x|^c)[f(x,r) = \{1\} \text{ or } f(x,r) = "?"]]$$

ii. $(\forall x \in \pi(\mathbf{C}))[S_x = \{1\} \Rightarrow$

$$(\exists |r| \le |x|^c)[f(x,r) = 1]]$$

iii. $(\forall x \in \pi(\mathbf{C}))[S_x \neq \{1\} \Rightarrow$

$$(\forall |r| \le |x|^c)[f(x,r) = "?"]].$$

For notions involving the reduction of one problem to another we will be even less formal.

Definition 7 f *is a deterministic solution to* **C** *iff* $(\forall x \in \pi(\mathbf{C}))[f(x) \in S_x]$.

Let $D(\mathbf{C}) = \{f \mid f \text{ is a deterministic solution to } \mathbf{C}\}$. For all deterministic algorithms $\mathcal{A}$ and functions f and g, we say that $\mathcal{A}$ translates f into g iff when given a subroutine for f, $\mathcal{A}$ computes g in polynomial time (where the time used in the subroutine for f is not counted). We remark that calls to the subroutine may be 'dovetailed' but the algorithm A cannot know if the absence of a response on a particular call means that no response is forthcoming or that a response has just not arrived yet. See **C18** for an example.

Definition 8 $\mathbf{C1} \le_{\mathcal{P}} \mathbf{C2}$ *iff there exists a deterministic algorithm* $\mathcal{A}$ *such that for all* $f \in D(\mathbf{C2})$, *there exists a* $g \in D(\mathbf{C1})$ *such that* $\mathcal{A}$ *translates* f *into* g *in polynomial time.*

Definition 9 **C** *is* $\mathcal{NP}$-hard *with respect to* $\mathcal{P}$ *iff for all* $\mathbf{C}'$, $(\mathbf{C}' \text{ is in } \mathcal{NP}) \Rightarrow (\mathbf{C}' \le_{\mathcal{P}} \mathbf{C})$.

We will follow the convention of using $\mathcal{NP}$-hard to denote $\mathcal{NP}$-hard with respect to $\mathcal{P}$.

Definition 10 f *is a random solution to* **C** *iff there exists a* c *in* **N** *such that*

i. $(\forall x \in \pi(\mathbf{C}))(\forall |r| \le |x|^c)$

$$[f(x,r) \in S_x \text{ or } f(x,r) = "?"]$$

ii. $(\forall x \in \pi(\mathbf{C}))$

$$\left[\frac{\#\{r \mid |r| \le |x|^c \ \& \ f(x,r) \in S_x\}}{\#\{r \mid |r| \le |x|^c\}} \ge \frac{1}{2}\right]$$

Let $R(\mathbf{C}) = \{f \mid f \text{ is a random solution to } \mathbf{C}\}$.

Definition 11 $\mathbf{C1} \le_{\mathcal{R}} \mathbf{C2}$ *iff there exists a deterministic algorithm* $\mathcal{A}$ *such that for all* $f \in D(\mathbf{C2})$, *there exists a* $g \in R(\mathbf{C1})$ *such that* $\mathcal{A}$ *translates* f *into* g *in polynomial time.*

Definition 12 **C** *is* $\mathcal{NP}$-hard *with respect to* $\mathcal{R}$ *iff for all* $\mathbf{C}'$,

$$(\mathbf{C}' \text{ is in } \mathcal{NP}) \Rightarrow \mathbf{C}' \le_{\mathcal{R}} \mathbf{C} \ .$$

1 Primality testing

C1	Input	$n \in \mathbf{N}$
	Output	1 if $n \in$ *Primes*, 0 otherwise.
O1a	Is **C1** in $\mathcal{P}$?	
O1b	Is **C1** recognized in $\mathcal{R}$?	

Rem1$_{86}$ A classical problem. The following quote appears in art. 329 of Gauss' *Disquisitiones Arithmeticæ*:(translation from [Knu81, page 398])

> The problem of distinguishing prime numbers from composites, and of resolving composite numbers into their prime factors, is one of the most important and useful in all of arithmetic. ...The dignity of science seems to demand that every aid to the solution of such an elegant and celebrated problem be zealously cultivated.

It is known that the set of composites is recognized in $\mathcal{R}$ [SS77]. If the extended Riemann hypothesis for Dirichlet L-functions is true, then **C1** is in $\mathcal{P}$ [Mil76]. There exists a constant $c \in \mathbf{N}$ and a deterministic algorithm for **C1** with running time $O((\log n)^{c \log\log\log n})$ [APR83]. If Cramér's conjecture on the gaps between consecutive primes is true, then **C1** is recognized in $\mathcal{R}$ [GK86]. **C1** is recognized in $\mathcal{NP}$ [Pra75]. Fürer [Für85] has shown that the problem of distinguishing between products of two primes that are $\not\equiv 1 \pmod{24}$ and primes that are $\not\equiv 1 \pmod{24}$ is in $\mathcal{R}$.

Rem1$_{94}$ Problem **O1b** has been settled in the affirmative by Adleman and Huang [AH92]. As a result of the work of H. Maier on gaps between consecutive primes, the exact formulation of Cramér's conjecture has now been called into question, however the conjecture required for [GK86] is unaffected.

Ref1[Guy77], [Knu81], [Len81], [CL84], [Pom81], [Rab80a], [Rie85b], [Rie85a], [Wil78].

2 Testing an infinite set of primes

Let $S \subset \mathbf{N}$.

C2	Input	$n \in \mathbf{N}$.
	Output	1 if $n \in S$, 0 otherwise.
O2	Does there exist an infinite set $S \subset$ *Primes* such that **C2** is in $\mathcal{P}$?	

Rem2$_{86}$ In light of **Rem1**$_{86}$ it is remarkable that **O2** remains unsettled. The related problem of the existence of an infinite set $S \subset$ *Primes* such that **C2** is recognized in $\mathcal{R}$ is addressed in [GK86].

Rem2$_{94}$ Problem **O2** has been settled in the affirmative by Pintz, Steiger, and Szemerédi [PSS89]. One can now ask what the densest such set S is. In this direction, Konyagin and Pomerance [KP94] have proved that for every $\epsilon > 0$ there exists an algorithm that will prove primality in deterministic polynomial time for at least $x^{1-\epsilon}$ primes less than x.

Ref2[PSS88].

3 Prime greater than a given bound

C3	Input	$n \in \mathbf{N}$.
	Output	$p \in$ *Primes* with $p > n$.
O3	Is **C3** in $\mathcal{P}$?	

Rem3$_{86}$ If Cramér's conjecture (see [Cra36]) on the gaps between consecutive primes is true, then **C3** $\leq_{\mathcal{P}}$**C1**. Since the density of primes between n and $2n$ is approximately $1/\log n$, it follows that **C3** $\leq_{\mathcal{R}}$**C1**. This problem has cryptographic significance [DH76], [RSA78].

Rem3$_{94}$ As we mentioned in **Rem1**$_{94}$, the exact formulation of Cramér's conjecture has now been called into question. It is still probably true that for every constant $c > 2$, there is a constant $d > 0$ such that there is a prime between x and $x + d(\log x)^c$. This hypothesis still implies that **C3** $\leq_{\mathcal{P}}$**C1**.

Note, since **C1** is recognized in $\mathcal{R}$ (see **Rem1**$_{94}$), it follows that **C3** is in $\mathcal{R}$. If anything, the importance of this problem has grown since 1986, since there have been numerous cryptosystems proposed since then that require the ability to construct large primes, sometimes with special properties. See [Pom90].

Ref3[Bac88], [Pla79]. See also **Ref1**.

4 Prime in an arithmetic progression

C4	Input	$a, n \in \mathbf{N}$.
	Output	$p \in$ *Primes* with $p \equiv a \pmod{n}$ if $\gcd(a, n) = 1$.
O4	Is **C4** in $\mathcal{P}$?	

Rem4$_{86}$ It was conjectured by Heath-Brown [HB78] that if $\gcd(a, n) = 1$, then the least prime $p \equiv a \pmod{n}$ is $O(n \log^2 n)$, and this would imply that **C4** $\leq_{\mathcal{P}}$**C1**. If there are no Siegel zeroes, then the density of small primes in the arithmetic progression a modulo n is sufficient to conclude that **C4** $\leq_{\mathcal{R}}$**C1** [Bom74]. Without hypothesis, it is known [EH71] that Heath-Brown's conjecture is true for almost all pairs a, n with $\gcd(a, n) = 1$. Hence if **C1** is in $\mathcal{P}$, then one can solve **C4** in deterministic polynomial time for almost all inputs. See also **Rem20**$_{86}$.

Rem4$_{94}$ Since **C1** is now known to be in $\mathcal{R}$ (see **Rem1**$_{94}$), it follows that **C4** is also in $\mathcal{R}$. **C4** also has cryptographic applications [Sch91], [BM92], [oC91].

Ref4[AM77]

5 Integer factoring

C5	Input	$n \in \mathbf{N}$.
	Output	$p_1, p_2, \ldots, p_k \in$ *Primes* and $e_1, e_2, \ldots, e_k \in \mathbf{N}$ such that $$n = \prod_{i=1}^{k} p_i^{e_i} \text{ if } n > 1 \ .$$
O5a	Is **C5** in $\mathcal{P}$?	
O5b	Is **C5** in $\mathcal{R}$?	

Rem5$_{86}$ Another classical problem, mentioned by Gauss in his *Disquisitiones Arithmeticæ* (see **Rem1**$_{86}$). There are a large number of random algorithms for **C5** whose running time is believed to be $L(n)^c$ for varying constants $c \geq 1$ [Pom82], [Len87], [SL84]. The only random algorithm of this class whose running time has actually been proved to be $L(n)^c$ is due to Dixon [Dix81]. Dixon's algorithm is unfortunately not practical. A determination of the complexity of **C5** would have significance in cryptography [RSA78].

Rem5$_{94}$ A great deal of progress has been made in the area of factoring integers. Lenstra and Pomerance [LP92] proved the existence of a probabilistic algorithm for factoring integers with an expected running time of $L_n[1/2, 1]$, improving on Dixon's bound. Another interesting development was Pollard's discovery of the number field sieve. A heuristic analysis suggests that there exists a constant $c > 0$ such that the number field sieve factors an integer n in expected time $L_n[1/3, c]$. Contributions to the number field sieve were made by a number of researchers, including (but not limited to) Adleman, Buhler, Coppersmith, Couveignes, A.K. Lenstra, H.W. Lenstra, Manasse, Odlyzko, Pollard, Pomerance and Schroeppel. See [Adl91], [Cop90], [Cou93], [LL93], and the references cited therein.

In a very recent development Peter Shor [Shoar] has shown that factoring can be done in polynomial time on a "quantum computer". It is premature to judge the implications of this development.

Ref5[Dix81], [Guy77], [Knu81], [Len87], [MB75], [Pom82], [Rie85b], [Rie85a], [Sha71], [Sch82], [SL84], [Wil84].

6 Factoring a set of positive density

Let $S \subset \mathbf{N}$.

C6 Input $n \in \mathbf{N}$.
Output $p_1, p_2, \ldots, p_k \in$ *Primes* and $e_1, e_2, \ldots, e_k \in \mathbf{N}$ such that

$$n = \prod_{i=1}^{k} p_i^{e_i} \text{ if } n > 1 \text{ and } n \in S \ .$$

O6 Does there exist a set S such that

$$\liminf_{x\to\infty} \frac{\#\{n \mid n \leq x \ \&\ n \in S\}}{x} > 0$$

and $\mathbf{C6}(S)$ is in $\mathcal{P}$?

Rem6$_{86}$ Assuming the necessary hypotheses for the running time analysis for Lenstra's elliptic curve factoring method (see [Len87]), it is probably possible to prove that a set S satisfying

$$\liminf_{x\to\infty} \frac{\#\{n \mid n \leq x \ \&\ n \in S\}}{\frac{x \log\log^2 x}{\log x \log\log\log x}} > 0 \qquad (1)$$

can be factored in random polynomial time. This set will still have density zero, however. A related question is whether factoring a set of positive density is random polynomial time equivalent to **C5**. The set *Squarefrees* has density $6/\pi^2$ however it is not even clear that $\mathbf{C5} \leq_{\mathcal{R}} \mathbf{C6}(\textit{Squarefrees})$.

Rem6$_{94}$ Let A denote a deterministic algorithm for factoring integers, and define $F(x, t, A)$ to be the number of integers n with $1 \leq n \leq x$ such that A will factor n in at most t bit operations. **O6** can then be stated as asking whether there exists an algorithm A and a constant $c > 0$ such that

$$\liminf_{x\to\infty} \frac{F(x, \log^c x, A)}{x} > 0 \ .$$

This problem remains open, but Hafner and McCurley [HM89a] and later Sorenson [Sor90] proved several results about the behaviour of F for various factoring algorithms (including a generalization to cover probabilistic algorithms). The estimate (1) has still not been proved, and the best result known [HM89a] in this direction is

$$F(x, \log^c x, A) >>_c \frac{x(\log\log x)^{\frac{6}{5}-\epsilon}}{\log x} \ ,$$

using a probabilistic algorithm. In this formulation, one may also ask for the slowest growing function $t(x)$ such that there exists an algorithm A with

$$\liminf_{x\to\infty} \frac{F(x, t(x), A)}{x} > 0 \ .$$

7 Squarefree part

C7 Input $n \in \mathbf{N}$.
Output $r, s \in \mathbf{N}$ with $n = r^2 s$ and $s \in$ *Squarefrees*.

O7a Is **C7** in $\mathcal{P}$?

O7b Is $\mathbf{C5} \leq_{\mathcal{R}} \mathbf{C7}$?

Rem7$_{86}$ See **Rem13**$_{86}$. Clearly $\mathbf{C7} \leq_{\mathcal{P}} \mathbf{C5}$. The analogous question for $f \in \mathbf{Q}[x]$ or $(\mathbf{Z}/p\mathbf{Z})[x]$ is solvable in polynomial time by performing calculations of the form $\gcd(f, f')$, where f' is the (formal) derivative of f. (see [Knu81, page 421]).

Rem7$_{94}$ Landau [Lan88] proved that $\mathbf{C7} \leq_{\mathcal{P}} \mathbf{C23}$. According to [Len92], Chistov [Chi89] has shown that **C7** is polynomial time equivalent to determining the ring of integers in a number field.

8 Squarefreeness

C8 Input $n \in \mathbf{N}$.
Output 1 if $n \in$ *Squarefrees*,
0 otherwise.

O8 Is **C8** in $\mathcal{P}$?

Rem8$_{86}$ A generalization of this is, given n and $k \in \mathbf{N}$, to determine if n is divisible by the kth power of a prime. Another generalization is to output $\mu = \mu(n)$, where

$$\mu(n) = \begin{cases} 1 & \text{if } n = 1, \\ 0 & \text{if exists a } p \in \textit{Primes} \text{ with } p^2 \mid n \\ (-1)^k & \text{if } n \text{ is a product of } k \text{ distinct primes.} \end{cases}$$

Shallit and Shamir have shown that this generalization is reducible to the problem of computing the function d mentioned in **Rem9**$_{86}$.

Rem8$_{94}$ We are unaware of any progress on this problem.

9 Number of distinct prime factors

C9	Input	$n \in \mathbb{N}$.
	Output	$\omega(n) = \#\{p \mid p \in \mathit{Primes} \;\&\; p \mid n\}$.
O9	Is **C9** in $\mathcal{P}$?	

Rem9$_{86}$ Clearly **C1** $\leq_{\mathcal{P}}$**C9**, since we can easily check to see if n is a perfect power. An interesting variant of **C9** is to output $\Omega(n) = e_1 + \ldots + e_k$, where $n = \prod_{i=1}^{k} p_i^{e_i}$ is the prime factorization of n. Another variant is to output $d(n) = \#\{k \mid k \in \mathbb{N} \;\&\; k \mid n\}$, and still another variant is to output the multiset $\{e_1, \ldots, e_k\}$. Shallit and Shamir [SS85] have proved that the last two variants are polynomial time equivalent to each other. As a consequence we have that **C9** is polynomial time reducible to the problem of computing the function $d(n)$ mentioned above.

Rem9$_{94}$ We are unaware of any progress on this problem. It is remarkable that one can decide if $\omega(n) = 1$ in random polynomial time [AH92], but there are no other partial results known on this problem.

10 Roots modulo a composite

C10	Input	$e, a, n \in \mathbb{N}$.
	Output	$x \in \mathbb{N}$ such that $x^e \equiv a \pmod{n}$, if $\gcd(e, \phi(n)) = 1$ and $\gcd(a, n) = 1$.
O10	Is **C5** $\leq_{\mathcal{R}}$**C10**?	

Rem10$_{86}$ When the restriction that $\gcd(e, \phi(n)) = 1$ is dropped, it is known that **C5** $\leq_{\mathcal{R}}$**C10** [Rab79]. A resolution of this problem would have important consequences in public-key cryptography [RSA78]. It is known that **C10** $\leq_{\mathcal{P}}$**C23**.

Rem10$_{94}$ We are unaware of any progress on this problem.

11 Quadratic residuosity modulo a composite

C11	Input	$a, n \in \mathbb{N}$.	
	Output	1	if there exists an $x \in \mathbb{N}$ such that $x^2 \equiv a \pmod{n}$ and $\gcd(a, n) = 1$,
		0	otherwise.
O11a	Is **C11** in $\mathcal{P}$?		
O11b	Is **C5** $\leq_{\mathcal{R}}$**C11**?		

Rem11$_{86}$ It is easy to show that **C11** $\leq_{\mathcal{P}}$**C5**. There is an obvious generalization where the exponent 2 is replaced by another exponent k that is either fixed for the problem or supplied as an input. The presumed difficulty of **C11** has been used as a basis for cryptographic systems [GM82], [GM84], [Yao82], [BBS86]. **C11** is related to **C9** since the proportion of residues modulo n that are quadratic residues is $2^{-\omega(n)}$, where $\omega(n)$ is the number of distinct prime divisors of n. Therefore given an algorithm for **C11**, one can obtain a confidence interval for $\omega(n)$ by checking random values.

Rem11$_{94}$ We are unaware of any progress on this problem.

Ref11[AM82].

12 Quadratic non-residue modulo a prime

C12	Input	$p \in \mathbb{N}$.
	Output	$b \in \mathbb{N}$ such that there does not exist $c \in \mathbb{N}$ with $c^2 \equiv b \pmod{p}$, if $p \in \mathit{Primes}$.
O12	Is **C12** in $\mathcal{P}$?	

Rem12$_{86}$ **C12** is easily seen to be in $\mathcal{R}$, since polynomial time algorithms for the corresponding problem of distinguishing quadratic residues from nonresidues can be based on the Jacobi symbol and the law of quadratic reciprocity, or else on Euler's criterion:

$$p \in \mathit{Primes} \text{ and } p \not| a \Rightarrow a^{\frac{p-1}{2}} \equiv \left(\frac{a}{p}\right) \pmod{p} \;.$$

Curiously, Gauss was aware of Euler's criterion, but was apparently unimpressed by its efficiency [Gau86, art. 106]:

> Although it is of almost no practical use, it is worthy of mention because of its simplicity and generality ... But as soon as the numbers we are examining are even moderately large this criterion is practically useless because of the amount of calculation involved.

Under the extended Riemann hypothesis, **C12** is in $\mathcal{P}$ [Mil76]. It is also known that the least quadratic nonresidue is almost always small [Erd61], so **C12** can be solved in deterministic polynomial time for almost all inputs.

Rem12$_{94}$ On the problem of calculating kth power non-residues in $GF(p^n)$, the following is known. On ERH, the algorithm of Huang [Hua85], generalized by Evdokimov [Evd89], constructs a kth power non-residue, in $GF(p^n)$ in deterministic time $(kn \log p)^{O(1)}$. Buchmann and Shoup [BS91], on ERH, construct a kth power non-residue in $GF(p^n)$ in deterministic time $(\log p)^{O(n)}$. Bach [Bac90], on ERH, has given explicit bounds for estimations of the least kth power non-residue. See also **Rem19**$_{94}$.

Ref12[Ank52], [Bac85].

13 Quadratic signature

C13	Input	$\sigma \in \{-1, 1\}^*$.
	Output	The least $p \in \mathit{Primes}$ such that for all i with $1 \le i \le |\sigma|$, $(\frac{p_i}{p}) = \epsilon_i$, where $|\sigma|$, the length of σ, is the number of symbols in σ, p_i is the i^{th} prime, and ϵ_i is the i^{th} symbol of σ.

O13 Is **C13** in $\mathcal{P}$?

Rem13$_{86}$ If n has the form m^2q with q an odd prime and m odd, then for any a with $\gcd(a, n) = 1$ we have $(\frac{a}{n}) = (\frac{a}{q})$. It follows that if **C13** is in $\mathcal{P}$, then n could be partially factored since, assuming the extended Riemann hypothesis, q can be determined by a signature of length $O(\log^2 n)$ [Mil76], [Ank52]. The notion of quadratic signature can be generalized; see [AM82].

Rem13$_{94}$ The concept of quadratic signature has found application in the number field sieve [Adl91].

Ref13[Ank52], [Bac85], [Bac90].

14 Square roots modulo a prime

C14	Input	$a, p \in \mathbf{N}$.
	Output	$x \in \mathbf{N}$ with $x^2 \equiv a \pmod p$ if $p \in \mathit{Primes}$ and such an x exists.

O14 Is **C14** in $\mathcal{P}$?

Rem14$_{86}$ Among the researchers who have presented algorithms for **C14** are [Gau86, art. 319-322], [Ton91], [Leh69], [Sha72], [Ber67], [Rab80b], [AMM77]. It is now known that **C14** is in $\mathcal{R}$. It is also known that **C14** $\le_{\mathcal{P}}$ **C12** and that on the extended Riemann hypothesis, **C14** is in $\mathcal{P}$. There is a natural generalization of **C14** where the exponent 2 is replaced by a fixed k. Another generalization has k as part of the input. For this version there is a random time $O((k \log p)^c)$ algorithm based on known algorithms for **C15**. One can also use a discrete logarithm algorithm (see **Rem21**$_{86}$) to solve this variant, resulting in a random time $O(L(p))$ algorithm, which for large k will be faster.

Rem14$_{94}$ It is an oversight that we did not mention the work of Schoof [Sch85] on this problem in our earlier manuscript. Schoof proved that for fixed a, there exists a deterministic algorithm with running time polynomial in $\log p$. also **Ref16** and [Hua85], [Evd89], [BS91].

15 Polynomial roots modulo a prime

C15	Input	$p \in \mathbf{N}$, $f \in (\mathbf{Z}/p\mathbf{Z})[x]$.
	Output	$a \in \mathbf{Z}$ with $f(a) \equiv 0 \pmod p$ if $p \in \mathit{Primes}$ and such an a exists.

O15 Is **C15** in $\mathcal{P}$?

Rem15$_{86}$ See **Rem14**$_{86}$. **C15** is in $\mathcal{R}$ [Ber70], [CZ81], [Rab80b]. If the extended Riemann hypothesis is assumed and f has abelian Galois group over the rationals, then the problem is in $\mathcal{P}$ [Hua85].

Rem15$_{94}$ If f is fixed the problem appears to remain difficult; however, for certain f progress has been made. When f is linear the problem is trivial. When f is a quadratic there exists a deterministic polynomial time algorithm due to Schoof [Sch85]. When f is a cyclotomic polynomial, there exists a deterministic polynomial time algorithm due to Pila [Pil90].

Ref15[Sho90b], [BS91]. See also **Ref16**.

16 Factoring polynomials modulo a prime

C16	Input	$p \in \mathbf{N}, f \in (\mathbf{Z}/p\mathbf{Z})[x]$.
	Output	irreducible $g_1, \ldots, g_k \in (\mathbf{Z}/p\mathbf{Z})[x]$, and $e_1, \ldots, e_k \in \mathbf{N}$ such that $f = \prod_{i=1}^{k} g_i^{e_i}$, if $p \in \mathit{Primes}$.

O16 Is **C16** in $\mathcal{P}$?

Rem16$_{86}$ See **Rem15**$_{86}$. **C16** is in $\mathcal{R}$ [Ber70], [CZ81], [Rab80b]. The corresponding problem over $\mathbf{Q}$ is in $\mathcal{P}$ [LLL82].

Rem16$_{94}$ Let n denote the

degree of f. Rónyai [Rón88] on ERH gives a deterministic algorithm with running time $(n^n \log p)^{O(1)}$. Evdokimov [Evdar] on ERH gives a deterministic algorithm with running time $(n^{\log n} \log p)^{O(1)}$. In particular, both algorithms are polynomial time if the degree is bounded. For the case $f \in Z[x]$, f irreducible and $Q[x]/(f)$ Abelian over Q, Huang [Hua91] on ERH gives a deterministic polynomial time algorithm. For the case $f \in Z[x]$, f irreducible and $Q[x]/(f)$ Galois over Q, Rónyai on ERH gives a deterministic polynomial time algorithm [Rón89]. For the case $f \in Z[x]$ solvable, Evdokimov [Evd89] on ERH gives a deterministic polynomial time algorithm.

Lenstra [Len90] has shown in many cases the assumption of ERH above may be removed if irreducible polynomials of appropriate degree can be found in deterministic polynomial time.

Buchmann and Shoup [BS91] proved, under ERH, that for all $n \in \mathbf{N}$, there exists a deterministic algorithm for **C16** with running time $\sqrt{k}$ times a polynomial in the input size, where k is the largest prime dividing $\phi_n(p)$ and ϕ_n is the n-th cyclotomic polynomial.

Ref16[Ber67], [Ber68], [Knu81, pages 420–441], [LN83, pages 147-185].

17 Irreducible polynomials

C17	Input	$d, p \in \mathbf{N}$.
	Output	irreducible $f \in (\mathbf{Z}/p\mathbf{Z})[x]$ with $degree(f) = d$, if $p \in Primes$.
O17	Is **C17** in $\mathcal{P}$?	

Rem17$_{86}$ **C17** is in $\mathcal{R}$ [Ber68], [Rab80b]. **C17** is in $\mathcal{P}$ if the extended Riemann hypothesis is true [AL86]. There is a $c \in \mathbf{N}$ and a deterministic polynomial time algorithm which on input d, p with $p \in Primes$ outputs an irreducible $f \in (\mathbf{Z}/p\mathbf{Z})[x]$ of degree greater than $cd/\log p$ and less than or equal to d [AL86]. Since irreducible quadratics yield quadratic nonresidues, it is clear that **C12** $\leq_{\mathcal{P}}$**C17**, and also from the results on **C14** that **C14** $\leq_{\mathcal{P}}$**C17**.

Rem17$_{94}$ The result of [AL86] was discovered independently by Evdokimov [Evd89]. Shoup [Sho90a] proved **C17** $\leq_{\mathcal{P}}$**C16**, and gave a deterministic algorithm for finding an irreducible polynomial of degree d over $\mathbf{Z}/p\mathbf{Z}$ in time $\sqrt{p}(d + \log p)^{O(1)}$.

Ref17[Len92].

18 Recognition of a primitive root modulo a prime

C18	Input	$b, p \in \mathbf{N}$.
	Output	1 if b is a generator of $(\mathbf{Z}/p\mathbf{Z})^*$ and $p \in Primes$, 0 if b is not a generator of $(\mathbf{Z}/p\mathbf{Z})^*$ and $p \in Primes$.
O18a	Is **C18** in $\mathcal{P}$?	
O18b	Is **C18** recognized in $\mathcal{R}$?	

Rem18$_{86}$ It is known that **C18** $\leq_{\mathcal{P}}$**C5**, since b is a primitive root modulo p if and only if $p \not| b$ and

$$\forall q[[q \in Primes \;\&\; q \mid p-1] \Rightarrow b^{(p-1)/q} \not\equiv 1 \pmod{p}] \;.$$

A generalization of **C18** where a third input $c \in \mathbf{N}$ is given and the output is 1 if and only if b has order c is also of interest.

Rem18$_{94}$ We are unaware of any progress on this problem. We would like to point out however that under ERH, **C18** $\leq_{\mathcal{P}}$**C21**. To see why, recall that under ERH, the least primitive root modulo p is $\leq c \log^6 p$ for some constant c [Sho90c]. Let g be a suspected primitive root modulo p. We dovetail the following procedures:

process A for $b = 1, 2, \ldots, c \log^6 p$: ask oracle for **C21** to compute an x with $g^x \equiv b \pmod{p}$. If the oracle returns an x keep it only if you confirm that $g^x \equiv b \pmod{p}$. If for all b an x is kept then output "primitive root".

process B for $b = 1, 2, \ldots, c \log^6 p$: ask oracle for **C21** to compute x such that $b^x \equiv g \pmod{p}$. If the oracle returns an x keep it only if you confirm that $b^x \equiv g \pmod{p}$. If for some b an x is kept with $\gcd(x, p-1) > 1$, then output "not a primitive root".

19 Finding a primitive root modulo a prime

C19	Input	$p \in \mathbf{N}$.
	Output	$g \in \mathbf{N}$ such that $1 \leq g \leq p-1$ and g generates $(\mathbf{Z}/p\mathbf{Z})^*$, if $p \in Primes$.
O19	Is **C19** in $\mathcal{P}$?	

Rem19$_{86}$ The density of generators is sufficient that it is easily shown that **C19** $\leq_{\mathcal{R}}$**C18**. If the extended Riemann hypothesis is true, then the least generator is small [Wan61], and **C19** $\leq_{\mathcal{P}}$**C18**. An interesting variant of **C19** involves finding elements of $(\mathbf{Z}/p\mathbf{Z})^*$ of desired order. **C19** has an obvious extension to an arbitrary finite field, or for that matter to any cyclic group.

Rem19$_{94}$ Shoup [Sho90c] proved several results related to this problem. Among other things, he proved under the assumption of the extended Riemann hypothesis that a primitive root for $GF(p^2)$ can be constructed in deterministic polynomial time. Buchmann and Shoup [BS91], on ERH, give a deterministic algorithm, which on input an irreducible f of degree n over $\mathbb{Z}/p\mathbb{Z}$, outputs a generating set for $\mathbb{Z}/p\mathbb{Z}[x]/(f)$ in time $(\log p)^{O(n)}$. As a consequence, if the factorization of $p^n - 1$ is known, then under the assumption of ERH, a primitive root of $GF(p^n)$ can be computed in deterministic polynomial time.

20 Calculation of orders modulo a prime

C20	Input	$a, p \in \mathbb{N}$.
	Output	$k = min\{x \mid x \in \mathbb{N}, a^x \equiv 1 \pmod p\}$, if $p \in$ *Primes* and $\gcd(a,p) = 1$.
O20	Is **C20** in $\mathcal{P}$?	

Rem20$_{86}$ The variant in which p is not required to be prime is random polynomial time equivalent to **C5** [Mil76]. A related question: is the problem of factoring numbers of the form $p - 1$, with p prime, polynomial time reducible to **C20**? If **C6** is in $\mathcal{P}$, then the problem of factoring numbers of the form $p - 1$ with p prime is polynomial time equivalent to factoring.

Rem20$_{94}$ We are unaware of any progress on this problem.

21 Discrete logarithm modulo a prime

C21	Input	$g, b, p \in \mathbb{N}$.
	Output	$x \in \mathbb{N}$ with $g^x \equiv b \pmod p$, if $p \in$ *Primes* and such an x exists.
O21	Is **C21** in $\mathcal{P}$?	

Rem21$_{86}$ If the prime factors of $p - 1$ are less than $\log^c p$ for some constant $c > 0$, then the problem is in $\mathcal{P}$ [PH78]. The fastest known algorithms for solving **C21** have running times of $L(p)$ [COS86]. The resolution of **O21** would have important consequences in cryptography [ElG85], [BM84]. There is an obvious generalization of **C21** to an arbitrary finite field. Bach [Bac84] has asked if the problem of factoring numbers of the form $p - 1$, with p prime, is polynomial time reducible to **C21**.

Rem21$_{94}$ There has been considerable progress on this problem. Pomerance [Pom86] proved that there exists a probabilistic algorithm to compute discrete logarithms in $GF(q)$ with expected running time of $L_q[1/2, \sqrt{2}]$, for the case where q is prime or q is a power of 2. Gordon [Gor93] presented an adaptation of the number field sieve to computing discrete logarithms in $\mathbb{Z}/p\mathbb{Z}$, along with a heuristic argument to suggest an expected running time of $L_p[1/3, c]$ for some positive constant c.

For discrete logarithms over general finite fields, progress has also been made. At the time that we wrote our original paper, we neglected to mention the work of Coppersmith [Cop84], who had published an algorithm for $GF(2^n)$ with a heuristic expected running time bounded by $L_{2^n}[1/3, c]$ for some positive constant c. Lovorn [Lov92] proved a running time of $L_q[1/2, c]$ for some positive constant c when $q = p^n$ with $\log p \le n^{0.98}$. Adleman and DeMarrais [AD93a] gave an algorithm for arbitrary finite fields whose heuristic expected running time is $L_q[1/2, c]$ for some positive constant c. Adleman's function field sieve [Adlar] gives a heuristic expected running time of $L_q[1/3, c]$ for some positive constant c when $q = p^n$ and $\log p \le n^{g(n)}$, where g is any function such that $0 < g(n) < 0.98$ and $\lim_{n\to\infty} g(n) = 0$.

Surveys on the discrete logarithm problem have been published: [vO91], [McC90a], [Odl94].

Historically, advances in integer factoring algorithms have brought corresponding advances in discrete logarithm algorithms. Adleman thinks it is an interesting research problem to establish whether reductions exist between **C5** and **C21**. McCurley finds the evidence for the existence of such reductions to be unconvincing.

In a very recent development Peter Shor [Shoar] has shown that discrete logarithms can be computed in polynomial time on a "quantum computer". It is premature to judge the implications of this development.

Ref21[Odl85], [Sch93], [AD93b].

22 Discrete logarithm modulo a composite

C22	Input	$g, b, n \in \mathbb{N}$.
	Output	$x \in \mathbb{N}$ with $g^x \equiv b \pmod n$, if such an x exists.
O22a	Is **C22** in $\mathcal{P}$?	
O22b	Is **C5** $\le_{\mathcal{P}}$**C22**?	

Rem22$_{86}$ Clearly **C21** $\le_{\mathcal{P}}$**C22**. It is also known that **C5** $\le_{\mathcal{R}}$**C22** [Bac84]. The resolution of **O22** would have consequences in public-key cryptography

[McC88]. There is an obvious generalization to an arbitrary group (see also **C28**).

Rem22$_{94}$ We are unaware of any progress on this problem.

23 Calculation of $\phi(n)$

C23 Input $n \in \mathbb{N}$.
Output $\phi(n)$.

O23 Is **C5** $\leq_{\mathcal{P}}$**C23**?

Rem23$_{86}$ It is known that **C5** $\leq_{\mathcal{R}}$**C23** [Mil76], and it is obvious that **C23** $\leq_{\mathcal{P}}$**C5**. **C5** is known to be random polynomial time equivalent to the problem of computing $\sigma(n)$, the sum of the positive integral divisors of n [BMS84].

Rem23$_{94}$ We are unaware of any progress on this problem. See **Rem7**$_{94}$.

24 Point on an elliptic curve

C24 Input $a, b, p \in \mathbb{N}$.
Output $x, y \in \mathbb{N}$ with $y^2 \equiv x^3 + ax + b \pmod p$, if $p \in$ *Primes* and $p \not| 4a^3 + 27b^2$.

O24 Is **C24** in $\mathcal{P}$?

Rem24$_{86}$ One can show that **C24** is in $\mathcal{R}$, since there is an easy argument to show that **C24** $\leq_{\mathcal{R}}$**C14**: choose random values of x, evaluate the right hand side, and use a random algorithm for **C14** to try to solve for y. A theorem of Hasse implies that the probability of choosing a successful x is approximately $\frac{1}{2}$.

Rem24$_{94}$ We are unaware of any progress on this problem. **C24** has applications in cryptography [Kob87b, p. 162].

25 Binary quadratic congruences

C25 Input $k, m, n \in \mathbb{N}$.
Output $x, y \in \mathbb{N}$ with $x^2 - ky^2 \equiv m \pmod n$, if n is odd and $\gcd(km, n) = 1$.

O25 Is **C25** in $\mathcal{P}$?

Rem25$_{86}$ **C25** is in $\mathcal{R}$ [AEM87]. If the extended Riemann hypothesis and Heath-Brown's conjecture on the least prime in an arithmetic progression are true, then **C25** is in $\mathcal{P}$ [Sha84]. **C25** arose from cryptography [OSS84], [PS87]. In fact, **C25** is only one example of a wide range of questions concerning solutions of $f \equiv 0 \pmod n$, where f is a multivariate polynomial with coefficients in $\mathbb{Z}/n\mathbb{Z}$. Such questions can vary greatly in their complexity as the form of the question changes. We may ask questions about determining if a solution exists, finding a solution, finding the least solution, or finding the number of solutions. We may vary the form of the polynomial or the properties of n (e.g. prime, composite, squarefree). As an example of the variation in complexity, even for the polynomial $f(x) = x^2 - a$ we have the following situation:

1. The problem of deciding from inputs $a, p \in \mathbb{N}$ whether $x^2 - a \equiv 0 \pmod p$ has a solution when p is prime is in $\mathcal{P}$ (see **Rem12**$_{86}$.)
2. The problem of finding from inputs $a, p \in \mathbb{N}$ a solution of $x^2 - a \equiv 0 \pmod p$ when p is prime is in $\mathcal{R}$ (see **Rem14**$_{86}$).
3. The problem of finding from inputs $a, n \in \mathbb{N}$ a solution of $x^2 - a \equiv 0 \pmod n$ is random equivalent to the problem of factoring n (see **Rem10**$_{86}$).
4. The problem of finding from inputs $a, n \in \mathbb{N}$ the least positive integer solution of $x^2 - a \equiv 0 \pmod n$ is $\mathcal{NP}$-hard [MA78].

We therefore view the problem of classifying all problems concerning solutions of $f \equiv 0 \pmod n$ according to their complexity as an important metaproblem.

Rem25$_{94}$ We are unaware of any progress on this problem. There has been marginal progress on the "metaproblem". We regard this area as a very fruitful one for future investigations.

Ref25[vzGKS93]. Some cryptographic problems related to the metaproblem are mentioned in [McC90b]. That paper also contains pointers to other unsolved number-theoretic problems relating to cryptology.

26 Key distribution

C26 Input $g, p, a, b \in \mathbb{N}$.
Output $c \in \mathbb{N}$, where $c \equiv g^{xy} \pmod p$, if $p \in$ *Primes*, g is a primitive root modulo p, $a \equiv g^x \pmod p$, and $b \equiv g^y \pmod p$.

O26 Is **C21** $\leq_{\mathcal{R}}$**C26**?

Rem26$_{86}$ The motivation for this problem comes from cryptography [DH76]. It is obvious that **C26** $\leq_{\mathcal{P}}$**C21**. There is a generalization where p is replaced by a composite n, and we ask only for an output c when a and b are powers of g. For this generalization is the problem equivalent to **C5** or **C22** (see [Bac84], [McC88])?

Rem26$_{94}$ Bert den Boer [dB90] proved that when all prime factors of $\phi(p-1)$ are small, the key distribution problem is as hard as computing discrete logarithms.

Ref26[Odl85], [ElG85].

27 Construction of an elliptic curve group of a given order

C27 Input $p, n \in \mathbf{N}$.
Output $a, b \in \mathbf{N}$ with $\#E_{p,a,b} = n$, if $p \in$ *Primes* and such an a, b exist.

O27 Is C27 in $\mathcal{P}$?

Rem27$_{86}$ There is a polynomial time algorithm that, given p, a, and b with $p \nmid 4a^3 + 27b^2$ computes $\#E_{p,a,b}$ [Sch85].

Rem27$_{94}$ We are unaware of any progress on this problem, however it is known that for some primes p, supersingular curves of order $p+1$ can be constructed efficiently (see [MOV94]).

Ref27[Kob87b], [Kob87a], [Sch85], [Sil86], [Kob91], [Kob91], [Kob88].

28 Discrete logarithms in elliptic curve groups

C28 Input $a, b, p \in \mathbf{N}, P, Q \in S_{p,a,b}$
Output $n \in \mathbf{N}$ with $P = nQ$, if $p \in$ *Primes* and such an n exists.

O28 Is **C28** in $\mathcal{P}$?

Rem28$_{86}$ The presumed difficulty of this problem has been used as the basis for a public key cryptosystem and digital signature scheme [Kob87b], [Mil86]. Whereas for the discrete logarithm problem in the multiplicative group modulo a prime there is a subexponential algorithm (see **Rem21**$_{86}$), no such algorithm is known to exist for **C28**. A related problem is given a, b, and p to construct a minimal set of generators for $E_{p,a,b}$.

Rem28$_{94}$ Menezes, Okamoto, and Vanstone [MOV94] used Weil pairing to prove that there exists a probabilistic reduction from **C28** to the problem of computing discrete logarithm in the multiplicative group of a (perhaps high degree) extension of GF(q). For supersingular curves, this reduction can be carried out in random polynomial time, with the result that a probabilistic subexponential algorithm is obtained for **C28** in this special case.

Koblitz [Kob90] has suggested cryptographic uses for the rational subgroups of the Jacobian of a hyperelliptic curve over a finite field. Adleman, Huang, and DeMarrais [AHDar] discovered a heuristic subexponential probabilistic algorithm for the discrete logarithm problem in these subgroups when the genus of the curve is large with respect to the size of the finite field.

29 Shortest vector in a lattice

C29 Input $b_1, \ldots, b_n \in \mathbf{Z}^n$
Output $v \in \Lambda$ with $\|v\|_2 = \min\{\|x\|_2 \mid x \in \Lambda, x \neq 0\}$, where $\Lambda = \mathbf{Z}b_1 \oplus \ldots \oplus \mathbf{Z}b_n$ if $b_1, \ldots, b_n$ span $\mathbf{R}^n$.

O29 Is **C29** $\mathcal{NP}$-hard?

Rem29$_{86}$ The corresponding problems with norms $\|\cdot\|_\infty$ and $\|\cdot\|_1$ are known to be $\mathcal{NP}$-hard [Lag85], [vEB81]. See also **Rem30**$_{86}$.

Rem29$_{94}$ It was an oversight that we did not mention the result of Lenstra [Len83], who proved that if the dimension n is fixed, the shortest vector in a lattice of dimension n can be found in polynomial time.

Ref29[GLS88] and [Lov86] contain nice surveys of this and related topics.

30 Short vector in a lattice

Let $c \in \mathbf{N}$

C30 Input $b_1, \ldots, b_n \in \mathbf{Z}^n$
Output $v \in \Lambda$ with $\|v\|_2 \leq n^c \min\{\|x\|_2 \mid x \in \Lambda, x \neq 0\}$, where $\Lambda = \mathbf{Z}b_1 \oplus \ldots \oplus \mathbf{Z}b_n$ if $b_1, \ldots, b_n$ span $\mathbf{R}^n$.

O30 Does there exist a $c \in \mathbf{N}$ for which **C30** is in $\mathcal{P}$?

Rem30$_{86}$ In [LLL82] it was shown that there is a polynomial time algorithm that produces a vector $v \in \Lambda$ with

$$\|v\|_2 \leq 2^{\frac{n-1}{2}} \min\{\|x\|_2 \mid x \in \Lambda, x \neq 0\} ,$$

and in [Sey87] it was shown that for any $\epsilon > 0$ there is a polynomial time algorithm $\mathcal{A}_\epsilon$ that produces a vector $v \in \Lambda$ with

$$\|v\|_2 \leq (1+\epsilon)^n \min\{\|x\|_2 \mid x \in \Lambda, x \neq 0\} .$$

A number of related problems in simultaneous diophantine approximation are discussed in [Lag85] and [Fru85].

Rem30$_{94}$ We are unaware of any progress on this problem.

Ref30[LLS90], [GLS88], [Lov86].

31 Galois group of a polynomial

C31 Input $f \in \mathbb{Q}[x]$.
Output $n = [K : \mathbb{Q}]$, where K is the splitting field of f.

O31 Is **C31** in $\mathcal{P}$?

Rem31$_{86}$ n is the order of the Galois group associated with f. Polynomial time algorithms exist for determining if n is a power of 2 or if the Galois group is solvable [LM85]. Many other properties of the Galois group can also be determined in polynomial time [Kan85].

Rem31$_{94}$ Landau [Lan85] proved that the Galois group can be computed in deterministic time $O((\#G+\ell)^c)$ for some constant $c > 0$, where ℓ is the length of the input specification of f and K. Further results are discussed in [Len92], but the problem remains open.

32 Class numbers

C32 Input $d \in \mathbb{N}$.
Output $h(-d)$, the order of the group of equivalence classes of binary quadratic forms with discriminant $-d$ under composition.

O32 Is **C32** in $\mathcal{P}$?

Rem32$_{86}$ This is related to classical questions of Gauss [Gau86, art. 303]. It appears that the results of Shanks [Sha72], [Sha71], Schnorr & Lenstra [SL84], Seysen [Sey87], and Schoof [Sch82] establish that **C5** $\leq_{\mathcal{R}}$**C32**, and that ERH implies **C5** $\leq_{\mathcal{P}}$**C32**. It is remarked in [BMS84] that it is not even known if **C32** is in $\mathcal{NP}$. The best known algorithm for computing $h(-d)$ is due to Shanks [Sha71]. The question could also be stated in terms of the class number of orders in the field $\mathbb{Q}(\sqrt{-d})$.

Rem32$_{94}$ McCurley [McC89] proved under ERH that **C32** is in $\mathcal{NP}$. Hafner and McCurley [HM89b] proved under ERH that there exists a probabilistic algorithm with expected running time $L_d[1/2, \sqrt{2}]$ that will compute not only the class number $h(-d)$, but also the structure of the class group. These results were extended to the case of real quadratic fields by Buchmann and Williams [BW89]. Thiel [Thiar] has shown under ERH that verifying the class number belongs to $\mathcal{NP} \cap \mathrm{co}\mathcal{NP}$.

The more general question of computing class numbers and class groups of arbitrary algebraic number fields is also of interest. According to Lenstra [Len92], Buchmann and Lenstra proved that there is a deterministic exponential time algorithm for computing the cardinality and structure of the class group. Buchmann [Buc90] gave a probabilistic subexponential algorithm for a special case of this problem. Lenstra [Len92] outlines an approach to obtaining a probabilistic subexponential algorithm in the general case.

Lenstra's paper [Len92] is an important source for information concerning algorithms and open problems concerning algebraic number fields.

Ref32[Gol85], [Sha72], [Sch82], [Lag80b], [Buc90].

33 Solvability of binary quadratic diophantine equations

C33 Input $a, b, c, d, e, f \in \mathbb{Z}$.
Output 1 if there exists $x, y \in \mathbb{Z}$ with $ax^2+bxy+cy^2+dx+ey+f = 0$ and there does not exist a $g \in \mathbb{Z}$ with $b^2 - 4ac = g^2$,
0 otherwise.

O33a Is **C33** $\mathcal{NP}$-hard?

O33b Is **C33** $\mathcal{NP}$-hard with respect to $\mathcal{R}$?

Rem33$_{86}$ It is known that **C33** is recognized in $\mathcal{NP}$ [Lag79]. Without the constraint that $b^2 - 4ac$ is not a square, the problem is known to be $\mathcal{NP}$-hard [MA78]. Certain variants of **C33** are known to be $\mathcal{NP}$-hard with respect to $\mathcal{R}$ [AM77].

Rem33$_{94}$ We are unaware of any progress on this problem.

34 Solvability of anti-Pellian equation

C34 Input $d \in \mathbb{N}$.
Output 1 if there exist $x, y \in \mathbb{Z}$ with $x^2 - dy^2 = -1$,
0 otherwise.

O34 Is **C34** in $\mathcal{P}$?

Rem34$_{86}$ There exist choices of d for which the smallest solution of $x^2 - dy^2 = -1$ cannot be written down in polynomial space [Lag79]. It is known that **C34** is in $\mathcal{NP}$ [Lag80a]. If the factorization of d is provided as part of the input, then the problem is recognized in $\mathcal{R}$, and if in addition we assume the extended Riemann hypothesis, then the problem is in $\mathcal{P}$ [Lag80a].

Rem34$_{94}$ We are unaware of any progress on this problem.

35 Greatest common divisors in parallel

C35 Input $a, b \in \mathbf{N}$.
Output $\gcd(a, b)$.

O35 Is **C35** in $\mathcal{NC}$?

Rem35$_{86}$ The best known results for computing greatest common divisors in parallel are contained in [BK83], [CG] and [KMR87]. One may ask a similar question for the modular exponentiation problem: given $a, b, n \in \mathbf{N}$, compute $a^b \pmod n$. For a definition of $\mathcal{NC}$ see [Coo85] or [Coo81].

Rem35$_{94}$ Polylog depth, subexponential size circuits for both integer GCD and modular exponentiation have been obtained by Adleman and Kompella [AK88].

Ref35[KMR84].

36 Integer multiplication in linear time

C36 Input $a, b \in \mathbf{N}$.
Output ab.

O36 Does there exist an algorithm to solve **C36** that uses only $O(\log(ab))$ bit operations ?

Rem36$_{86}$ The best known algorithm is due to Schönhage and Strassen and uses $O(\log(ab) \cdot \log\log(ab) \cdot \log\log\log(ab))$ bit operations [SS71].

Rem36$_{94}$ We are unaware of any progress on this problem.

Ref36[Knu81, pages 278-301]

Acknowledgments

I wish to reiterate my thanks to Kevin McCurley for permission to use our open problems list in this paper. In the course of maintaining the open problems list, we have benefited from conversations with many researchers. We would especially like to thank Jonathan DeMarrais, Andrew Granville, Ming-Deh Huang, Neal Koblitz, Jeff Lagarias and Gary Miller for their contributions.

References

[AD93a] Leonard M. Adleman and Jonathan DeMarrais. A subexponential algorithm for discrete logarithms over all finite fields. In Douglas R. Stinson, editor, *Advances in Cryptology: Crypto '93*, volume 773 of *Lecture Notes in Computer Science*, pages 147–158, New York, 1993. Springer-Verlag.

[AD93b] Leonard M. Adleman and Jonathan DeMarrais. A subexponential algorithm for discrete logarithms over all finite fields. *Mathematics of Computation*, 61:1–15, 1993. Extended abstract in [AD93a].

[Adl91] Leonard M. Adleman. Factoring numbers using singular integers. In *Proceedings of the 23th Annual Symposium on Theory of Computing*, pages 64–71, 1991.

[Adlar] Leonard M. Adleman. The function field sieve. In *Proceedings of the 1994 Algorithmic Number Theory Symposium*, Lecture Notes in Computer Science. Springer-Verlag, to appear.

[AEM87] Leonard M. Adleman, Dennis Estes, and Kevin S. McCurley. Solving bivariate quadratic congruences in random polynomial time. *Mathematics of Computation*, 48:17–28, 1987.

[AH92] Leonard M. Adleman and Ming-Deh Huang. *Primality testing and two dimensional Abelian varieties over finite fields*, volume 1512 of *Lecture Notes in Mathematics*. Springer-Verlag, 1992.

[AHDar] Leonard M. Adleman, Ming-Deh A. Huang, and Jonathan DeMarrais. A subexponential algorithm for discrete logarithms in the rational subgroup of the Jacobian of a hyperelliptic curve over a finite field. In *Proceedings of the 1994 Algorithmic Number Theory Symposium*, Lecture Notes in Computer Science. Springer-Verlag, to appear.

[AHU74] Alan Aho, John Hopcroft, and Jeffrey Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, MA, 1974.

[AK88] Leonard M. Adleman and Kireeti Kompella. Using smoothness to achieve parallelism. In *Proceedings of the 20th ACM Symposium on Theory of Computing*, pages 528–538, 1988.

[AL86] Leonard M. Adleman and H. W. Lenstra, Jr. Finding irreducible polynomials over finite fields. In *Proceedings of the 18th Annual Symposium on Theory of Computing*, pages 350–355, New York, 1986. Association for Computing Machinery.

[AM77] Leonard M. Adleman and K. Manders. Reducibility, randomness, and intractibility. In *Proc. 9th Annual ACM Symposium On Theory Of Computing*, pages 151–163, New York, 1977. Association for Computing Machinery.

[AM82] Leonard M. Adleman and R. McDonnell. An application of higher reciprocity to computational number theory. In *Proceedings of the 22nd Annual Symposium on Foundations of Computer Science*, pages 100–106. IEEE Computer Society, 1982.

[AM86] Leonard M. Adleman and Kevin S. McCurley. Open problems in number-theoretic complexity. In *Discrete Algorithms and Complexity (Proceedings of the Japan-US Joint Seminar on Discrete Algorithms and Complexity Theory)*, pages 237–262. Academic Press, 1986.

[AMM77] Leonard M. Adleman, K. Manders, and Gary L. Miller. On taking roots in finite fields. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, pages 175–178, Rhode Island, 1977. IEEE Computer Society.

[AMar] Leonard M. Adleman and Kevin S. McCurley. Open problems in number-theoretic complexity ii. In *Proceedings of the 1994 Algorithmic Number Theory Symposium*, Lecture Notes in Computer Science. Springer-Verlag, to appear.

[Ank52] N. Ankeny. The least quadratic non-residue. *Annals of Mathematics*, 55:65–72, 1952.

[APR83] Leonard M. Adleman, Carl Pomerance, and Robert Rumely. On distinguishing prime numbers from composite numbers. *Annals of Mathematics*, 117:173–206, 1983.

[Bac84] Eric Bach. Discrete logarithms and factoring. Technical Report UCB/CSD 84/186, University of California, Computer Science Division (EECS), University of California, Berkely, California, June 1984.

[Bac85] Eric Bach. *Analytic Methods in the Analysis and Design of Number Theoretic Algorithms.* MIT Press, Cambridge, 1985.

[Bac88] Eric Bach. How to generate factored random numbers. *SIAM Journal of Computing*, 17:179–193, 1988.

[Bac90] Eric Bach. Explicit bounds for primality testing and related problems. *Mathematics of Computation*, 55:355–380, 1990.

[BBS86] Lenore Blum, Manuel Blum, and Michael Shub. A simple unpredictable pseudo-random number generator. *SIAM Journal of Computing*, 15:364–383, 1986.

[Ber67] Elwyn Berlekamp. Factoring polynomials over finite fields. *Bell System Technical Journal*, 46:1853–1859, 1967.

[Ber68] Elwyn Berlekamp. *Algebraic Coding Theory.* McGraw-Hill, New York, 1968.

[Ber70] Elwyn Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24:713–735, 1970.

[BK83] Richard Brent and H. Kung. Systolic VLSI arrays for linear time gcd computation. In F. Anceau and E. Aas, editors, *VLSI 83*, pages 145–154. IFIP, Elsevevier, 1983.

[BM84] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal of Computing*, 13:850–864, 1984.

[BM92] Ernest F. Brickell and Kevin S. McCurley. An interactive identification scheme based on discrete logarithms and factoring. *Journal of Cryptology*, 5:29–40, 1992.

[BMS84] Eric Bach, Gary L. Miller, and Jeffrey O. Shallit. Sums of divisors, perfect numbers, and factoring. In *Proceedings of the 16th Annual Symposium on Theory of Computing*, New York, 1984. Association for Computing Machinery.

[Bom74] Enrico Bombieri. Le grand crible dans la théorie analytique des nombres. Avec une sommaire en anglais. *Astérisque*, 18, 1974.

[BS91] Johannes Buchmann and Victor Shoup. Constructing non-residues in finite fields and the extended Riemann hypothesis. In *Proceedings of the 23th Annual Symposium on Theory of Computing*, pages 72–79, 1991.

[Buc90] Johannes Buchmann. Complexity of algorithms in algebraic number theory. In R.A. Mollin, editor, *Proceedings of the First Conference of the Canadian Number Theory Association*, pages 37–53, Berlin, 1990. De Gruyter.

[BW89] Johannes Buchmann and Hugh C. Williams. On the existence of a short proof for the value of the class number and regulator of a real quadratic field. In Richard A. Mollin, editor, *Proceedings of the NATO Advanced Study Institute on Number Theory and Applications*, pages 327–345, The Netherlands, 1989. Kluwer.

[CG] B. Chor and O. Goldreich. An improved parallel algorithm for integer GCD. *Algorithmica.* To Appear.

[Chi89] A. L. Chistov. The complexity of constructing the ring of integers of a global field. *Dokl. Akad. Nauk. SSSR*, 306:1063–1067, 1989. English translation: Soviet Math. Dokl. 39 (1989), 597-600.

[CL84] H. Cohen and H. W. Lenstra, Jr. Primality testing and Jacobi sums. *Mathematics of Computation*, 42:297–330, 1984.

[Coo81] Stephen Cook. Towards a complexity theory of synchronous parallel computation. *Enseignment Math.*, 27:99–124, 1981.

[Coo85] Stephen Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64:2–22, 1985.

[Cop84] Don Coppersmith. Fast evaluation of discrete logarithms in fields of characteristic two. *IEEE Transactions on Information Theory*, 30:587–594, 1984.

[Cop90] Don Coppersmith. Modifications to the number field sieve. Technical Report RC16264, IBM TJ Watson Research Center, Yorktown Heights, New York, 1990.

[COS86] Don Coppersmith, Andrew Odlyzko, and Richard Schroeppel. Discrete logarithms in GF(p). *Algorithmica*, 1:1–15, 1986.

[Cou93] Jean-Marc Couveignes. Computing a square root for the number field sieve. In A. K. Lenstra and H. W. Lenstra, Jr., editors, *The development of the number field sieve*, number 1554 in Lecture Notes in Mathematics, pages 95–102. Springer-Verlag, 1993.

[Cra36] H. Cramér. On the order of magnitude of the difference between consecutive prime numbers. *Acta Arithmetica*, 2:23–46, 1936.

[CZ81] David Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 36:587–592, 1981.

[dB90] Bert den Boer. Diffie-Hellman is as strong as discrete log for certain primes. In *Advances in Cryptology: Proceedings of Crypto '88*, volume 403 of *Lecture Notes in Computer Science*, pages 530–539, New York, 1990. Springer-Verlag.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976.

[Dic71] Leonard E. Dickson. *History of the Theory of Numbers*, volume 1. Chelsea Books, New York, 1971.

[Dix81] John D. Dixon. Asymptotically fast factorization of integers. *Mathematics of Computation*, 36:255–260, 1981.

[EH71] P. D. T. A. Elliot and H. Halberstam. The least prime in an arithmetic progression. In *Studies in Pure Mathematics*, pages 69–61. Academic Press, London, 1971.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.

[Erd61] P. Erdös. Remarks on number theory, I. *Mat. Lapok*, 12:10–17, 1961.

[Evd89] S. A. Evdokimov. Factoring a solvable polynomial over a finite field and generalized Riemann hypothesis. *Zapiski Nauchn. Semin. Leningr. Otdel. Matem. Inst. Acad. Sci. USSR*, 176:104–117, 1989. In Russian.

[Evdar] S. A. Evdokimov. Factorization of polynomials over finite fields. In *Proceedings of the 1994 Algorithmic Number Theory Symposium*, Lecture Notes in Computer Science, Berlin, to appear. Springer-Verlag.

[Fru85] M. A. Frumkin. Complexity questions in number theory. *J. Soviet Math.*, 29:1502–1517, 1985. Translated from Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Matematicheskogo Instituta im. V. A. Steklova AN SSSR, vol. 118 (1982), 188-210.

[Für85] Martin Fürer. Deterministic and Las Vegas primality testing algorithms. In *Proceedings of ICALP 1985*, 1985.

[Gau86] Karl Friedrich Gauss. *Disquisitiones Arithmeticæ*. Springer-Verlag, New York, 1986. Reprint of the 1966 English translation by Arthur A. Clarke, S.J., Yale University Press, revised by William C. Waterhouse. Original 1801 edition published by Fleischer, Leipzig.

[Gil77] John Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal of Computing*, 4:675–695, 1977.

[GJ79] Michael Garey and David Johnson. *Computers and Intractibility: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, 1979.

[GK86] Shafi Goldwasser and Joe Kilian. Almost all primes can be quickly certified. In *Proceedings of the 18th Annual Symposium on Theory of Computing*, pages 316–329, New York, 1986. Association for Computing Machinery.

[GLS88] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the 14th Annual Symposium on Theory of Computing*, pages 365–377, New York, 1982. Association for Computing Machinery.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Science*, 28:270–299, 1984.

[Gol85] Dorian Goldfeld. Gauss' class number problem for imaginary quadratic fields. *Bulletin of the American Mathematical Society*, 13:23–38, 1985.

[Gor93] Daniel M. Gordon. Discrete logarithms in GF(p) using the number field sieve. *SIAM Journal of Discrete Mathematics*, 6:124–138, 1993.

[Guy77] Richard Guy. How to factor a number. *Congressus Numeratium*, XXVII:49–89, 1977. Proceedings of the Fifth Manitoba Conference on Numerical Mathematics, University of Manitoba.

[Har] Juris Hartmanis. Godel, von neumann and the p=?np problem.

[HB78] D. R. Heath-Brown. Almost-primes in arithmetic progressions and short intervals. *Mathematical Proceedings of the Cambridge Philosophical Society*, 83:357–375, 1978.

[HM89a] James Lee Hafner and Kevin S. McCurley. On the distribution of running times of certain integer factoring algorithms. *Journal of Algorithms*, 10:531–556, 1989.

[HM89b] James Lee Hafner and Kevin S. McCurley. A rigorous subexponential algorithm for computation of class groups. *Journal of the American Mathematical Society*, 2:837–850, 1989.

[Hua85] Ming-Deh A. Huang. Riemann hypothesis and finding roots over finite fields. In *Proceedings of the 17th Annual Symposium on Theory of Computing*, pages 121–130, New York, 1985. Association for Computing Machinery.

[Hua91] Ming-Deh A. Huang. Generalized riemann hypothesis and factoring polynomials over finite fields. *Journal of Algorithms*, 12:464–481, 1991.

[Kan85] William Kantor. Polynomial-time algorithms for finding elements of prime order and Sylow subgroups. *Journal of Algorithms*, 4:478–514, 1985.

[KMR84] Ravi Kannan, Gary L. Miller, and L. Rudolph. Sublinear parallel algorithm for computing the greatest common divisor of two integers. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science*, pages 7–11. IEEE Computer Society, 1984.

[KMR87] Ravi Kannan, Gary L. Miller, and L. Rudolph. Sublinear parallel algorithm for computing the greatest common divisor of two integers. *SIAM Journal of Computing*, 16:7–16, 1987. Extended abstract in [KMR84].

[Knu81] Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts, second edition, January 1981.

[Kob87a] Neal Koblitz. *A Course in Number Theory and Cryptography*. Number 114 in Graduate Texts in Mathematics. Springer-Verlag, New York, 1987.

[Kob87b] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48:203–209, 1987.

[Kob88] Neal Koblitz. Primality of the number of points on an elliptic curve over a finite field. *Pacific Journal of Mathematics*, 131:157–165, 1988.

[Kob90] Neal Koblitz. A family of Jacobians suitable for discrete log cryptosystems. In S. Goldwasser, editor, *Advances in Cryptology: Proceedings of Crypto '88*, volume 403 of *Lecture Notes in Computer Science*, pages 94–99, Berlin, 1990. Springer-Verlag.

[Kob91] Neal Koblitz. Constructing elliptic curve cryptosystems in characteristic 2. In A. J. Menezes and S. A. Vanstone, editors, *Advances in Cryptology: Proceedings of Crypto '90*, volume 537 of *Lecture Notes in Computer Science*, pages 156–167, Berlin, 1991. Springer-Verlag.

[KP94] Sergei Konyagin and Carl Pomerance. On primes recognizable in deterministic polynomial time. preprint, May 1994.

[Lag79] Jeffrey C. Lagarias. Succinct certificates for the solvability of binary quadratic diophantine equations. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, pages 47–54. IEEE Computer Society, 1979.

[Lag80a] Jeffrey C. Lagarias. On the computational complexity of determining the solvability or unsolvability of the equation $x^2 - dy^2 =$

−1. *Transactions of the American Mathematical Society*, 260:485–508, 1980.

[Lag80b] Jeffrey C. Lagarias. Worst-case complexity bounds for algorithms in the theory of integral quadratic forms. *Journal of Algorithms*, 1:142–186, 1980.

[Lag85] Jeffrey C. Lagarias. The computational complexity of simultaneous diophantine approximation problems. *SIAM Journal of Computing*, 14:196–209, 1985.

[Lan85] Susan Landau. Polynomial time algorithms for Galois groups. In J. Fitch, editor, *Proceedings of EUROSAM '84*, volume 174 of *Lecture Notes in Computer Science*, pages 225–236, New York, 1985. Springer-Verlag.

[Lan88] Susan Landau. Some remarks on computing the square parts of integers. *Information and Computation*, 78:246–253, 1988.

[Leh69] D. H. Lehmer. Computer technology applied to the theory of numbers. In *Studies in Number Theory*, pages 117–151. Mathematical Association of America, 1969. Distributed by Prentice Hall, Englewood Cliffs, NJ.

[Len81] H. W. Lenstra, Jr. Primality testing algorithms (after Adleman, Rumely, and Williams). In *Séminaire Bourbaki 1980/81, Exposé 576*, number 901 in Lecture Notes in Mathematics, pages 243–257. Springer-Verlag, Berlin, 1981.

[Len83] H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[Len87] H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126:649–673, 1987.

[Len90] H. W. Lenstra, Jr. Algorithms for finite fields. In *Number Theory and Cryptography*, volume 154 of *London Mathematical Society Lecture Note Series*, pages 76–85. Cambridge University Press, Cambridge, 1990.

[Len92] H. W. Lenstra, Jr. Algorithms in algebraic number theory. *Bulletin of the American Mathematical Society*, 26:211–244, 1992.

[LL93] Arjen K. Lenstra and H. W. Lenstra, Jr., editors. *The development of the number field sieve*. Number 1554 in Lecture Notes in Mathematics. Springer-Verlag, 1993.

[LLL82] Arjen K. Lenstra, H. W. Lenstra, Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[LLS90] Jeffrey C. Lagarias, H. W. Lenstra, Jr., and Claus-Peter Schnorr. Korkine-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10:333–348, 1990.

[LM85] Susan Landau and Gary L. Miller. Solvability by radicals is in polynomial time. *Journal of Computer and System Science*, 30:179–208, 1985.

[LN83] Rudolf Lidl and Harald Niederreiter. *Finite Fields*. Addison-Wesley, Reading, MA, 1983.

[Lov86] László Lovász. *An Algorithmic Theory of Numbers, Graphs, and Convexity*. Number 50 in CBMS-NSF Regional Conference Series in Applied Mathematics. Society of Industrial and Applied Mathematicians, Philadelphia, PA, 1986.

[Lov92] Renet Lovorn. *Rigorous Subexponential Algorithms for Discrete Logarithms over Finite Fields*. PhD thesis, University of Georgia, May 1992.

[LP92] H. W. Lenstra, Jr. and Carl Pomerance. A rigorous time bound for factoring integers. *Journal of the American Mathematical Society*, 5:483–516, 1992.

[MA78] K. Manders and Leonard M. Adleman. NP-complete decision problems for binary quadratics. *Journal of Computer and System Science*, 16:168–184, 1978.

[MB75] Michael Morrison and John Brillhart. A method of factoring and the factorization of F_7. *Mathematics of Computation*, 29:183–205, 1975.

[McC88] Kevin S. McCurley. A key distribution system equivalent to factoring. *Journal of Cryptology*, 1:95–105, 1988.

[McC89] Kevin S. McCurley. Cryptographic key distribution and computation in class groups. In Richard A. Mollin, editor, *Proceedings of the NATO Advanced Study Institute on Number Theory and Applications*, pages 459–479, The Netherlands, 1989. Kluwer.

[McC90a] Kevin S. McCurley. The discrete logarithm problem. In Pomerance [Pom90], pages 49–74.

[McC90b] Kevin S. McCurley. Odds and ends from cryptology and computational number theory. In Pomerance [Pom90], pages 145–166.

[Mil76] Gary L. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Science*, 13:300–317, 1976.

[Mil86] Victor Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology: Proceedings of Crypto '85*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426, Berlin, 1986. Springer-Verlag.

[MOV94] Alfred J. Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions in Information Theory*, ???, 1994. Extended abstract in Proceedings of the 23rd ACM Symposium on Theory of Computing, 1991, ACM, pp. 80–89.

[oC91] U.S. Department of Commerce. A proposed federal information processing standard for digital signature standard. In *Federal Register*, volume 56, no. 169, pages 42980–42982. U.S. GPO, August 1991.

[Odl85] Andrew Odlyzko. The discrete logarithm problem and its cryptographic significance. In *Advances in Cryptology: Proceedings of Eurocrypt '84*, volume 209 of *Lecture Notes in Computer Science*, pages 224–314, Berlin, 1985. Springer-Verlag.

[Odl94] Andrew Odlyzko. Discrete logarithms and smooth polynomials. In Gary L. Mullen and Peter Shiue, editors, *Finite Fields: Theory, Applications, and Algorithms*, Contemporary Mathematics Series, Providence, RI, 1994. American Mathematical Society.

[OSS84] H. Ong, Claus-Peter Schnorr, and Adi Shamir. An efficient signature scheme based on quadratic equations. In *Proceedings of the 16th Annual Symposium on Theory of Computing*, pages 208–216, New York, 1984. Association for Computing Machinery.

[PH78] Stephen Pohlig and Martin Hellman. An improved algorithm for computing discrete logarithms over GF(p) and its cryptographic significance. *IEEE Transactions on Information Theory*, 24:106–110, 1978.

[Pil90] Jonathan Pila. Frobenius maps of abelian varieties and finding roots of unity in finite fields. *Mathematics of Computation*, 55:745–763, 1990.

[Pla79] D. A. Plaisted. Fast verification, testing, and generation of large primes. *Theoretical Computer Science*, 9:1–16, 1979.

[Pom82] Carl Pomerance. Analysis and comparison of some integer factoring methods. In H. W. Lenstra, Jr. and R. Tijdeman, editors, *Computational Methods in Number Theory, Part I*, number 154 in Math. Centre Tract, pages 89–139. Math. Centre, Amsterdam, 1982.

[Pom86] Carl Pomerance. Fast, rigorous factorization and discrete logarithm algorithms. In *Discrete Algorithms and Complexity (Proceedings of the Japan-US Joint Seminar on Discrete Algorithms and Complexity Theory)*, pages 119–143. Academic Press, 1986.

[Pom90] Carl Pomerance, editor. *Cryptography and Computational Number Theory*, volume 42 of *Proceedings of Symposia in Applied Mathematics*. American Mathematical Society, Providence, 1990.

[Pom81] Carl Pomerance. Recent developments in primality testing. *Mathematical Intelligencer*, 3:97–105, 1980/81.

[Pra75] Vaughn Pratt. Every prime has a succinct certificate. *SIAM Journal of Computing*, 4:214–220, 1975.

[PS87] John Pollard and Claus-Peter Schnorr. Solution of $x^2 + ky^2 \equiv m \pmod n$, with application to digital signatures. *IEEE Transactions on Information Theory*, 22:702–709, 1987.

[PSS88] Janos Pintz, William L. Steiger, and Endre Szemerédi. Two infinite sets of primes with fast primality tests. In *Proceedings of the 20th Annual Symposium on Theory of Computing*, pages 504–509. Association for Computing Machinery, 1988. Journal version in [PSS89].

[PSS89] Janos Pintz, William L. Steiger, and Endre Szemerédi. Infinite sets of primes with fast primality tests and quick generation of large primes. *Mathematics of Computation*, 53:399–406, 1989. Extended abstract in [PSS88].

[Rab79] Michael O. Rabin. Digitalized signatures and public-key functions as intractible as factorization. Technical Report TR-212, Massachussetts Institute of Technology, Laboratory for Computer Science, 1979.

[Rab80a] Michael O. Rabin. Probabilistic algorithm for testing primality. *Journal of Number Theory*, 12:128–138, 1980.

[Rab80b] Michael O. Rabin. Probabilistic algorithms in finite fields. *SIAM Journal of Computing*, 9:273–280, 1980.

[Rie85a] Hans Riesel. Modern factorization methods. *BIT*, 25:205–222, 1985.

[Rie85b] Hans Riesel. *Prime Numbers and Computer Methods for Factorization.* Birkhäuser, Boston, 1985.

[Rón88] L. Rónyai. Factoring polynomials over finite fields. *Journal of Algorithms*, 9:391–400, 1988.

[Rón89] L. Rónyai. Galois groups and factoring polynomials over finite fields. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*, pages 99–104. IEEE Computer Society, 1989.

[RSA78] Ronald Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[Sch82] Rene Schoof. Quadratic fields and factorisation. In H. W. Lenstra, Jr. and R. Tijdeman, editors, *Computational Methods in Number Theory, Part I*, number 154 in Math. Centre Tract. Math. Centre, Amsterdam, 1982.

[Sch85] R. Schoof. Elliptic curves over finite fields and the computation of square roots modulo p. *Mathematics of Computation*, 44:483–494, 1985.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.

[Sch93] Oliver Schirokauer. Discrete logarithms and local units. *Philisophical Transactions of the Royal Society of London (A)*, 345:409–423, 1993.

[Sey87] Martin Seysen. A probabilistic factorization algorithm with quadratic forms of negative discriminant. *Mathematics of Computation*, 48:757–780, 1987.

[Sha71] Daniel Shanks. Class number, a theory of factorization, and genera. In *Analytic Number Theory*, volume 20 of *Proceedings of Symposia in Pure Mathematics*, pages 415–440. American Mathematical Society, 1971.

[Sha72] Daniel Shanks. Five number-theoretic algorithms. In *Proceedings of the Second Manitoba Conference on Numerical Mathematics*, pages 51–70, 1972.

[Sha84] Jeffrey O. Shallit. An exposition of Pollard's algorithm for quadratic congruences. Technical Report 84-006, University of Chicago, Department of Computer Science, December 1984.

[Sho90a] Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54:435–447, 1990. Extended abstract in Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science (1988), pp. 283–290.

[Sho90b] Victor Shoup. On the deterministic complexity of factoring polynomials over finite fields. *Information Processing Letters*, 33:261–267, 1990.

[Sho90c] Victor Shoup. Searching for primitive roots in finite fields. In *Proceedings of the 22th Annual Symposium on Theory of Computing*, pages 546–554, 1990.

[Shoar] Peter W. Shor. Polynomial time algorithms for discrete logarithms and factoring on a quantum computer. In *Proceedings of the 1994 Algorithmic Number Theory Symposium*. Springer-Verlag, to appear.

[Sil86] Joseph H. Silverman. *The Arithmetic of Elliptic Curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, 1986.

[SL84] Claus-Peter Schnorr and H. W. Lenstra, Jr. A Monte Carlo factoring algorithm with linear storage. *Mathematics of Computation*, 43:289–311, 1984.

[Sor90] Jonathan Sorenson. Counting the integers factorable via cyclotomic methods. Technical Report 919, University of Wisconsin, Deparment of Computer Sciences, 1990.

[SS71] Alfred Schönhage and Volker Strassen. Schnelle Multiplikation grosser Zahlen. *Computing*, 7:281–292, 1971.

[SS77] R. Solovay and Volker Strassen. A fast Monte-Carlo test for primality. *SIAM Journal of Computing*, 6:84–85, 1977.

[SS85] Jeffrey O. Shallit and Adi Shamir. Number-theoretic functions which are equivalent to number of divisors. *Information Processing Letters*, 20:151–153, 1985.

[Thiar] C. Thiel. Verifying the class number belongs to $\mathcal{NP}\cap\text{co}\mathcal{NP}$. In *Proceedings of the 1994 Algorithmic Number Theory Symposium*. Springer-Verlag, to appear.

[Ton91] A. Tonelli. Bemerkung ber die aufl sung quadratischer Congruenzen. *Göttinger Nachrichten*, pages 314–346, 1891.

[vEB81] P. van Emde Boas. Another NP-complete partition problem and the complexity of computing short vectors in lattices. Technical Report 81-04, Mathematics Department, University of Amsterdam, 1981.

[vO91] Paul van Oorschot. A comparison of practical public key cryptosystems based on integer factorization and discrete logarithms. In Gustavus J. Simmons, editor, *Contemporary Cryptology: The Science of Information Integrity*, IEEE Proceedings, pages 280–322. IEEE, 1991.

[vzGKS93] Joachim von zur Gathen, Marek Karpinski, and Igor Shparlinski. Counting curves and their projections. preprint, March 1993.

[Wan61] Y. Wang. On the least primitive root of a prime. *Scientia Sinica*, 10:1–14, 1961.

[Wil78] Hugh C. Williams. Primality testing on a computer. *Ars Combinatorica*, 5:127–185, 1978.

[Wil84] Hugh C. Williams. Factoring on a computer. *Mathematical Intelligencer*, 6:29–36, 1984.

[WS91] H.C. Williams and J.O. Shallit. Factoring integers before computers. In *Proceedings of Symposium in Applied Mathematics*, pages 1–51, 1991.

[Yao82] Andrew C. Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, pages 80–91. IEEE Computer Society, 1982.