Randomized Algorithms in Number Theory

MICHAEL O. RABIN
Hebrew University and Harvard University

AND

JEFFERY O. SHALLIT
University of Chicago

Introduction

Algorithmic problems always played a role in the study of number theory. One of the outstanding early examples is Euclid's algorithm (i.e., computational procedure) for finding the greatest common divisor (gcd) of two integers. His algorithm is efficient in a sense to be explained below. C. F. Gauss, who was deeply interested in computational questions, describes in his *Disquisitiones Arithmeticae* the problems of testing numbers for primality and factoring numbers into prime factors as being among the most important problems in arithmetic.

Not all interesting arithmetical questions are even in principle algorithmically solvable. A celebrated result of Matijasevitch states that there is no algorithm for testing, for every given polynomial Diophantine equation, whether it has a solution. For other problems, the possibility of algorithmic solution seems to be a trivial statement. Consider the problem of finding (for n) a sum of four (integer) squares representation $n = x^2 + y^2 + z^2 + w^2$. Simply enumerate all quadruples $x, y, z, w \leqq \sqrt{n}$, and test for every quadruple whether it solves the equation. Similar remarks apply to the primality testing and factorization problems, as well as to the gcd problem. So what challenge did Gauss see and what is Euclid's accomplishment?

The answer to the above question revolves around the issue of efficiency of algorithms. Finding the sum of squares representation by the simple-minded method will require $O(n^2)$ arithmetical operations. If n has, say, 200 binary digits, we are talking about 2^{400} operations. Similarly, the simple algorithm for testing primality will require $\sqrt{n} \sim 2^{100}$ operations, and will behave worst when n is in fact prime. On the other hand, Euclid's algorithm for the gcd of $m, n \sim 2^{200}$ will require at most 200 operations with integers smaller than 2^{200}, an entirely feasible computation by computer. We can say about Euclid's algorithm that it requires a number of operations linear in the *length* of n (when written in binary notation), i.e., in $\log_2 n$.

Communications on Pure and Applied Mathematics, Vol. XXXIX S239–S256 (1986)

CCC 0010-3640/86/0S0239-18$04.00

A reasonable concept for an efficient algorithm is one which when working on input n, will require $c \log^d n$ (all logarithms in this article are to base 2) arithmetical operations on integers smaller than n, where c and d are sufficiently small to render the algorithm practical, say $c \leqq 50$, $d \leqq 3$ (of course, if d is, say, 1 we can afford a larger c). This somewhat vague notion is a specialization of the "polynomial time solvable" notion popular in theoretical computer science. By arithmetical operations on n, m, we mean finding the binary representations of $m + n$, $m - n$, $m \cdot n$, $[m/n]$ and of the remainder of m when divided by n.

A surprising development of the past decade or so was that for certain number theoretic problems, efficient algorithms can be given by resorting within the computation to random choices of numbers. For example, in [17], [19] and [25], a randomized test for primality is given which tests n by $c \log n$ operations. A novel feature is that there is a small probability that the test will produce a wrong answer. For the exact meaning of these statements, and a report on experimental results the reader may look at [19]. Despite extremely interesting work on ordinary algorithms for primality, there is as yet no provably polynomial time algorithm for primality, and the randomized tests are by far the most efficient.

Another possibility arising when we use randomization, is to give up not assurance of the correctness of the result, but the ability to say exactly how many steps are required. Namely, with lucky random choices, success will come quickly, but the computation can also continue (with a small probability) for a long time. Thus we construct algorithms which for every n require at most an expected number $c \cdot \log^d n$ of operations, and always produce a correct result.

In the present paper we apply these ideas to a number of Diophantine problems involving sums of squares and arising out of classical theorems. Every prime number $p = 4k + 1$ can be represented as a sum $p = x^2 + y^2$ of two squares of integers. In Section 1, we present a randomized algorithm for this problem which requires an expected number $O(\log p)$ of operations. This algorithm appears in [18]. Recently, R. J. Schoof [24] gave a deterministic algorithm which requires $O(\log^6 p)$ operations.

In Section 2, we give two randomized algorithms for finding the representation $n = x^2 + y^2 + z^2 + w^2$. One, taken from [18], does it in an expected number of operations $O(\log^2 n)$. It uses integral quaternions and depends for its proof on the ERH. The second algorithm (see [21]) employs a celebrated theorem of Linnick and requires $O(\log^2 n \log\log n)$ operations. In Section 3, another method, taken from [23] and using quaternions, is given which unlike the first method does not employ any unproven number theoretic assumption. In Section 4, a randomized polynomial time algorithm for the sum of three squares problem is given, following [23]. For the classical literature on sums of squares, see [5], [7], [15], [26].

Each of the algorithms in Sections 2–4 employs two or more randomizations within it. They run very efficiently in practice, and, of course, always produce correct answers. As yet, there is no non-randomized algorithm for any of these problems. An interesting feature is that they involve prime numbers even though they solve equations for arbitrary integers n.

1. Primes as Sums of Two Squares

As a first example of the randomization method let us treat the well-known Fermat-Lagrange theorem to the effect that every prime $p = 4k + 1$ can be represented as a sum of two squares. Our aim is to present an efficient algorithm which for any given prime $p = 4k + 1$ will find integers x, y such that $p = x^2 + y^2$. The algorithm given here was first outlined in [18].

It is well known that if $p = 4k + 1$ is a prime, then there exists an integer $u < \frac{1}{2}p$ such that $u^2 + 1 \equiv 0 \bmod p$. In other words, the equation $t^2 + 1$ has a solution in the finite field Z_p of residues mod p. In [4] and [20], a randomizing method for finding solutions for a general polynomial equation over a finite field is given. For the sake of completeness we shall show how this algorithm specializes to the case of the equation $t^2 + 1 = 0$ (in Z_p).

Let the two roots of $t^2 + 1 = 0$ be u_1 and $u_2 = p - u_1$, then $u_1 \neq u_2$. For any residue $0 \leqq b < p$ consider the polynomial

$$f_b(t) = (t - b)^2 + 1 = t^2 - 2bt + b^2 + 1.$$

The roots of $f_b(t) = 0$ are $u_1 + b$ and $u_2 + b$.

We have $\frac{1}{2}(p - 1) = 2k$, so that the equation $t^{2k} - 1 = 0$ is satisfied by exactly the $2k$ quadratic residues in Z_p. Thus if, for example, $u_1 + b$ is a quadratic residue while $u_2 + b$ is a nonresidue, then we have

$$(1.1) \qquad \left(f_b(t), t^{2k} - 1\right) = t - u_1 - b,$$

where $(f(t), g(t))$ denotes the gcd of these polynomials. It follows that if we succeed in finding a $b \in Z_p$ with the above property, then we can compute u_1 from (1.1).

It would seem that computing the gcd in (1.1) requires $O(2k) = O(p)$ operations, but this is not so. Let $2k = d^{d_1} + \cdots + d^{d_m}$ be the binary representation of $2k$. Then $d_i \leqq \log_2 p$, $1 \leqq i \leqq m$, and $m \leqq \log_2 p$. Denoting $r = 2^d$, we can compute $t^r \bmod f_b(t)$ by computing the sequence

$$g_1 = t^2 \bmod f_b(t), \quad g_2 = g_1^2 \bmod f_b(t), \cdots, g_d = g_{d-1}^2 \bmod f_b(t).$$

Note that each $g_i(t)$ is a linear polynomial of the form $ct + e$, $c, e \in Z_p$, so that computing g_{i+1} from g_i requires a fixed number of operations. Thus computing all the powers $t^{2^i} \bmod f_b(t)$, $0 \leqq i \leqq \log_2 p$, requires $O(\log p)$ operations. Now, to obtain $h(t) = t^{2k} \bmod f_b(t)$ we just have to perform m multiplications $\bmod f_b(t)$ of linear polynomials, using the binary representation of $2k$.

Once $h(t)$ has been obtained, the gcd in (1.1) is computed by a fixed number of operations, since $(f_b(t), t^{2k} - 1) = (f_b(t), h(t) - 1)$.

How do we find a $b \in Z_p$ such that $h_1 + b$ is a quadratic residue while $h_2 + b$ is not, or vice-versa, so that (1.1) will hold? Call $\alpha, \beta \in Z_p$, $\alpha \neq 0$, $\beta \neq 0$, of *different type* if one is a quadratic residue while the other is not.

THEOREM [20]. *Let $\alpha_1, \alpha_2 \in Z_p$, $\alpha_1 \neq \alpha_2$; then*

$$\tfrac{1}{2}(p-1) = c\big(\{\delta \mid \delta \in Z_p, \alpha_1 + \delta \text{ and } \alpha_2 + \delta \text{ are of different type}\}\big),$$

The proof appears in [20], where the theorem is stated for arbitrary finite fields and is used for solving equations in any finite field.

We can now give a randomized algorithm for solving the equation $t^2 + 1 = 0$ in Z_p. Choose $b \in Z_p$ *randomly* and compute $((t-b)^2 + 1, t^{2k} - 1)$. By the above theorem, in an *expected* number of two tries we find a $b \in Z_p$ for which $u_1 + b$ and $u_2 + b$ are of different type so that (1.1) holds (for u_1 or u_2). Thus $u_1 + b$, and hence u_1 with $u_1^2 + 1 = 0 \bmod p$ is found in an expected number of operations $O(\log p)$.

Having found an integer $u < p$ so that $u^2 + 1 = mp$, we solve $p = x^2 + y^2$ by following one of the well-known proofs of the Fermat-Lagrange theorem, using Gaussian integers.

The ring GI consists of all complex numbers $a + ib$, $a, b \in Z$. Denote $N(a+ib) = a^2 + b^2$. It is readily seen that for $z, w \in GI$, $w \neq 0$, we can find $q, r \in GI$ such that $z = q \cdot w + r$ and $N(r) < \frac{1}{2}N(w)$. The computation requires a fixed number of operations with ordinary integers smaller than $\max(N(z), N(w))$. From the availability of a division with remainder algorithm with $N(r) < \frac{1}{2}N(w)$, it follows that Euclid's algorithm for computing the gcd (z_1, z_2), $z_1, z_2 \in GI$, will terminate in $\log_2(\max(N(z_1), N(z_2))$ steps.

Recall that we found a $u < p$ satisfying $(u+i)(u-i) = u^2 + 1 = mp$. Compute $(u + i, p) = x + iy$. It is readily seen that $p = x^2 + y^2$. The previous arguments establish the following

THEOREM 1.1. *For a prime $p = 4k + 1$ we can find the representation $p = x^2 + y^2$ by an $O(\log_2 p)$ expected number of arithmetical operations with integers smaller than p.*

2. Sum of Four Squares Using Primes

Every integer n can be expressed as a sum

$$n = x^2 + y^2 + z^2 + w^2 \tag{2.1}$$

of four integral squares. One elementary proof of this result proceeds from the fact that by the Euler identity, if n_1 and n_2 are each a sum of four squares, then so is $n_1 n_2$ (see Section 3). Hence it suffices to establish the result for primes p. This is done by showing the existence of a solution for $x^2 + y^2 + 1 = mp$, and proving by contradiction that the smallest m_1, for which $m_1 p$ is a sum of four squares, is $m_1 = 1$.

This proof does not lead to an efficient algorithm for solving (2.1). First, we would need a factorization of n into its prime factors and there is, as yet, no efficient factorization algorithm. Secondly, the proof that $x^2 + y^2 + 1 = mp$ is solvable proceeds by a counting argument which would reduce to an exhaustive search algorithm. Similar remarks apply to the attempts to turn other proofs into algorithms.

The earliest polynomial time algorithm for the sum of four squares problem appeared in [18]. Briefly, it proceeded as follows. For an odd n, consider the sequence

$$q = mn - 1, \qquad m \equiv 2 \mod 4, \quad m < n^3. \tag{2.2}$$

Every q in (2.2) satisfies $q \equiv 1 \mod 4$. From an unproven version of the extended Riemann hypothesis (ERH) it follows that the relative density of primes in the sequence (2.2) is $O(1/\log n)$. Choose $m < n^3$ randomly and test q for primality. In expected time $O(\log n)$, a prime $q = mn - 1 \equiv 1 \mod 4$ is found. Now express q as a sum of two squares by the method of Section 1. Thus $q = mn - 1 = u^2 + v^2$, $mn = u^2 + v^2 + 1$. From the last equation, a solution of (2.1) is found by using integral quaternions (see Section 3) in a manner similar to the use of Gaussian integers in Section 1. In [10], R. Kannan uses the shortest vector in a lattice algorithm, for giving another reduction from a solution of $mn = u^2 + v^2 + 1$ to a solution of (2.1).

The proof of correctness depends on ERH. Randomization is used twice, in finding a prime $q = mn - 1$, and in expressing q as a sum of two squares. The prime q is used to "piggyback" the solution for the number n.

We shall now present the solution in [21] which does not require any unproven number theoretic propositions. Consider the equation

$$n = x^2 + y^2 + p, \qquad p \text{ a prime.} \tag{2.3}$$

In a profound paper [12], Linnik proved a conjecture of Hardy and Littlewood concerning the asymptotic number of solutions of (2.3) as a function of n. As a corollary of that result we have that for a certain constant $0 < A$, there exists a number n_0 such that

$$n_0 < n \Rightarrow \text{number of solutions of (2.3)} > \frac{A \cdot n}{\log n \log\log n}. \tag{2.4}$$

Thus the obvious plan for expressing n as a sum of four squares is to find a representation (2.3) with $p = 4k + 1$, and to express p as $p = z^2 + w^2$. We can actually ensure that the prime p will have the required form.

LEMMA 2.1. *Let $n = 2(2m + 1)$. If x, y, p solve (2.3), then $p = 2$ or $p = 4k + 1$.*

The proof follows at once from the fact that a square u^2 has residue 0 or 1 mod 4.

THEOREM 2.2. *There is a randomizing algorithm for expressing a number n as a sum of four squares which requires an expected number $O((\log n)^2 \log\log n)$ of operations with integers smaller than n for $n_0 < n$.*

Proof: Assume first that $n = 2(2k+1)$. Choose randomly $x, y \leqq \sqrt{n}$. Employ the randomizing algorithm of Section 1 to $p = n - x^2 - y^2$, which by Lemma 2.1 satisfies $p = 4k + 1$ (the case $p = 2$ leads to $n = x^2 + y^2 + 1 + 1$). If p is prime, then this will produce a solution $p = z^2 + w^2$, so that $n = x^2 + y^2 + z^2 + w^2$.

However, since p may actually be composite, in which case the randomized algorithm for solving $u^2 + 1 = 0 \bmod p$ might not terminate, we introduce a small modification. Namely, in trying to solve $u^2 + 1 = mp$, we make just *one* random choice of $b \in Z_p$. If we do not solve the congruence $u^2 + 1 = 0 \bmod p$ by using $f_b(t)$ (see Section 1), then we discard p and again randomly choose $x, y \leqq \sqrt{n}$.

To treat the general n, we distinguish two cases. If n is odd, then apply the above algorithm to $m = 2n$. Having found

$$m = x^2 + y^2 + z^2 + w^2,$$

we observe that x, y, z, w must fall into two pairs of integers with equal residues mod 2, say $x \equiv y \bmod 2$ and $z \equiv w \bmod 2$. We then have

$$n = \tfrac{1}{2}m = \left(\tfrac{1}{2}(x+y)\right)^2 + \left(\tfrac{1}{2}(x-y)\right)^2 + \left(\tfrac{1}{2}(z+w)\right)^2 + \left(\tfrac{1}{2}(z-w)\right)^2.$$

Assume n is even, $n = 2^d(2k+1)$. If d is odd, then $n = s \cdot 2(2k+1)$, where s is a square. Applying the algorithm to $2(2k+1)$, we find a solution for (2.1). Finally, if d is even, find a sum of four squares representation for $2n$ by the previous remark, and derive a solution for n by the method of the previous paragraph.

It remains to analyze the expected number of operations. If suffices to treat the case $n = 2(2k+1)$. It follows from (2.4) that for $A \cdot n/\log n \log\log n$ pairs out of the n pairs $x, y \leqq \sqrt{n}$, the number $p = n - x^2 - y^2$ is a prime. Thus in an expected number at most $A^{-1}\log n \log\log n$ of random choices of pairs x, y, a prime p will be encountered. The probability of succeeding to solve $u^2 + 1 = 0 \bmod p$ with one random choice of $b \in Z_p$ is $\frac{1}{2}$, by the theorem in Section 1. Hence in an expected number smaller than $2A^{-1}\log n \log\log n$ of choices of pairs x, y, $p = n - x^2 - y^2 = z^2 + w^2$ will be solved. The computation associated with each choice of a pair requires $O(\log n)$ operations, so that the total expected number of operations required for solving (2.1) is $O(\log^2 n \log\log n)$.

A more detailed analysis shows that the constant A in (2.4) is large, say $1 \leqq A$, hence A^{-1} is small so that the expected number of choices until $p = n - x^2 - y^2$ is encountered is not large. In actual implementation, the algorithm for solving (2.1) runs very quickly for large values of n, as well as for small values of n (even though n_0 in (2.4) has not been calculated).

Finally, we wish to address a difficulty arising from the fact that, until the n_0 in (2.4) is computed, we do not know whether (2.3) has solutions for, say, every $10^6 < n$. It may happen for an $n < n_0$ that $p = n - x^2 - y^2$ is never a prime and the algorithm of Theorem 2.2 will not terminate. As mentioned just now, in practice the algorithm runs quickly for small values of n as well. Even without knowledge of n_0 we can, however, arrange our computation so that it will always terminate and have an asymptotic expected number of operations as in Theorem 2.2.

Consider the lexicographic ordering (x_i, y_i, z_i) of all the triples $x, y, z \leqq \sqrt{n}$. Interleve after every attempt in the algorithm of Theorem 1.1 to solve $u^2 + 1 = 0 \bmod p$, $p = n - x^2 - y^2$, a test for one triple whether $n - x_i^2 - y_i^2 - z_i^2$ is a square of an integer, taking the triples in the lexicographic order. Obviously, the algorithm will always solve (2.1) and for $n_0 < n$ will do so in the expected $O(\log^2 n \log\log n)$ number of operations. Thus we have proved

THEOREM 2.3. *There exists a constant* $0 < c$ *so that, for every integer* n, *a sum of four squares representation can be found in an expected number* $c \cdot \log^2 n \log\log n$ *of arithmetical operations.*

3. Sums of Four Squares Through Integral Quaternions

In this section, we give another method for obtaining four-square representations through the theory of *quaternions*. The method runs in random polynomial time.

Before we describe the method, we need some facts about quaternions.

The algebra of *rational quaternions* $\mathbf{H}(\mathbf{Q})$ can be viewed as a subset of $\mathbf{Q}^4$. It is defined by

$$\mathbf{H}(\mathbf{Q}) = \{h_1 + h_2 i + h_3 j + h_4 k \mid h_n \in \mathbf{Q}\},$$

where i, j, and k are the *coordinates* and satisfy $i^2 = j^2 = k^2 = -1$ and $ij = k$, $jk = i$, $ki = j$. Multiplication is defined by extending through linearity.

$\mathbf{H}(\mathbf{Q})$ contains a noncommutative subring the *Hurwitz integral quaternions*, $\mathbf{H}$, which are defined by

$$\mathbf{H} = \{h_1 + h_2 i + h_3 j + h_4 k \mid \text{all } h_n \text{ belong either to } \mathbf{Z} \text{ or } \mathbf{Z} + \tfrac{1}{2}\}.$$

The *conjugate* function is defined by

$$\operatorname{conj}(h_1 + h_2 i + h_3 j + h_4 k) = h_1 - h_2 i - h_3 j - h_4 k.$$

The *norm* $N(h)$ is given by

$$N(h) = \operatorname{conj}(h)h = h_1^2 + h_2^2 + h_3^2 + h_4^2;$$

the norm is multiplicative. The applicability of quaternions to the four-square problem is now easy to see; given four-square representations for a and b, we can easily find a four-square representation for ab by using quaternion multiplication.

The *units* are the 24 elements with norm 1:

$$\pm 1, \pm i, \pm j, \pm k, \pm \tfrac{1}{2} \pm \tfrac{1}{2}i \pm \tfrac{1}{2}j \pm \tfrac{1}{2}k.$$

We say h is an *associate* of g if $g = \varepsilon h$ for some unit ε. Every member of $\mathbf{H}$ with half-integer coordinates has an associate with integer coordinates. Inverses are given by $h^{-1} = \operatorname{conj}(h)(N(h))^{-1}$.

We say f is a *right divisor* of g if $g = af$. Given $g, h \in \mathbf{H}$, an element $f \in \mathbf{H}$ is said to be a *greatest common right divisor* (gcrd) of g and h if f is a right divisor of both g and h, and every right divisor of g and h is a right divisor of f. Any two integral quaternions have a gcrd which is unique up to a unit factor. For more information about the ring $\mathbf{H}$, see [9], 20.6–20.9.

Now we can give the outline of our algorithm to write M as the sum of four squares:

(1) We write $M = 2^e n$ with n odd. We can easily obtain a four-square representation for 2^e, and it now suffices to obtain the representation for n, since we may then obtain a representation for M using quaternion multiplication.

(2) Find a, b such that $a^2 + b^2 \equiv -1 \bmod n$

(3) Replace a (respectively b) by $n - a$ (respectively $n - b$), if necessary, to ensure that $a, b < \frac{1}{2}|n|$. Compute $g = \text{gcrd}\ (a + bi + j, n)$ in $\mathbf{H}$.

(4) If $N(g) = n$, then an associate of g gives a four-square representation for n. Otherwise $N(g) = kn$, where $k|n$, and so we have found a divisor of n. Apply the algorithm recursively to k and n/k, obtaining four-square representations; then combine the results using quaternion multiplication to obtain a four-square representation for n.

The remainder of this section is devoted to showing that this algorithm is correct and can be made to run in random polynomial time.

First, we show the following

THEOREM 3.1. *Suppose n is odd and* $\gcd(k, n) = 1$. *Then we can find a solution to $x^2 + y^2 \equiv k \bmod n$ in random polynomial time.*

Proof: The idea is quite simple. We choose w, z at random from $\mathbb{Z}_n$. Then if $w^2 + z^2 = r$, we have the congruence

$$(x^2 + y^2)(w^2 + z^2) \equiv kr \bmod n,$$

where x and y are sought. Now we claim $kr \bmod n$ will essentially be "randomly" distributed mod n and hence will "frequently" be a number for which we can quickly find a two-square representation, i.e., a prime power p^a congruent to $1 \bmod 4$. Now we write $p = u^2 + v^2$; from

$$(x^2 + y^2)(w^2 + z^2) \equiv u^2 + v^2 \bmod n,$$

we find

$$x \equiv (uw + vz)(w^2 + z^2)^{-1} \bmod n,$$

$$y \equiv (vw - uz)(w^2 + z^2)^{-1} \bmod n.$$

Although the main ideas are simple, formalizing them requires some work. We need some lemmas:

LEMMA 3.2. *Let $n = \prod_{i=1}^{k} p_i^{e_i}$, n odd. Let D be an integer relatively prime to n. Then the number of distinct pairs $(x, y) \in \mathbb{Z}_n \times \mathbb{Z}_n$ that satisfy*

$$x^2 - Dy^2 \equiv a \bmod n \tag{3.1}$$

for $a \in \mathbb{Z}_n^*$ is

$$\prod_{i=1}^{k} p_i^{e_i - 1}\left(p_i - \left(\frac{D}{p_i}\right)\right),$$

where $\left(\frac{D}{p}\right)$ is the Legendre symbol.

Proof: First we show the result in the case where $n = p$, an odd prime. Let X be an indeterminate. Consider the ring

$$R = \mathbb{Z}[X]/(X^2 - D, p).$$

There is a natural representation for elements of R in the form $x + y\sqrt{D}$ with $x, y \in \mathbb{Z}_p$. Let us write R^* for those elements of R with $x^2 - Dy^2 \not\equiv 0 \bmod p$. Consider the map $N\colon R^* \to \mathbb{Z}_p^*$ defined by

$$N(x + y\sqrt{D}) = x^2 - Dy^2.$$

It is readily verified that N is multiplicative. It is well known that N maps R^* *onto* $\mathbb{Z}_p^*$. Hence N is a (group) homomorphism; and therefore $x^2 - Dy^2$ takes on each value of $\mathbb{Z}_p^*$ equally often.

What is the structure of R^*? If $X^2 - D$ is irreducible mod p, i.e., $\left(\frac{D}{p}\right) = -1$, then R is isomorphic to $GF(p^2)$; thus $|R^*| = p^2 - 1$ and so there are $(p^2 - 1)/(p - 1) = p + 1$ solutions to (3.1) for each a. On the other hand, if $X^2 - D$ is reducible, then R^* is isomorphic to $\mathbb{Z}_p \times \mathbb{Z}_p$; thus $|R^*| = (p - 1)^2$ and so there are $(p - 1)^2/(p - 1) = p - 1$ solutions to (3.1) for each a.

This proves the lemma for the case $n = p$. Now a simple argument based on Hensel lifting shows that for each solution mod p^k there are p solutions mod p^{k+1}. Hence the lemma is true for prime powers. Finally, an application of the Chinese remainder theorem proves the result for all odd n. This completes the proof of the lemma.

COROLLARY. *If w and z are chosen randomly from $\mathbb{Z}_n$, then $w^2 + z^2$ is invertible* mod n *with probability at least* $\varphi(n)^2/n^2 > 1/(5 \log\log n)^2$.

Proof: Use the lemma with $D = -1$. We find that there are

$$\prod_{i=1}^{k} p_i^{e_i - 1}\left(p_i - \left(\frac{D}{p_i}\right)\right)$$

solutions for each $a \in \mathbb{Z}_n^*$; hence there are at least a total of

$$\varphi(n) \prod_{i=1}^{k} p_i^{e_i - 1}(p_i - 1) = \varphi(n)^2$$

solutions. An estimate of Rosser and Schoenfeld gives the last inequality (see [3]).

Thus from the above results, we need to try at most $25(\log\log n)^2$ pairs (w, z) on average until we find a pair (w, z) such that $w^2 + z^2$ is invertible mod n; and in this case $w^2 + z^2$ will in fact be a random element from $\mathbb{Z}_n^*$. Thus kr mod n will also be a random element from $\mathbb{Z}_n^*$. To complete the proof of Theorem 3.1, it remains to prove that a random element of $\mathbb{Z}_n^*$ is "likely" to be a prime power $p^a \equiv 1 \bmod 4$; we can quickly find a two-square representation for such numbers.

We have the following

LEMMA 3.3. *Let*

$$B_n = \{1 \leqq y \leqq n : y = p^a,\ p \text{ prime},\ y \equiv 1 \bmod 4,\ \text{and } \gcd(y, n) = 1\};$$

$1 \in B_n$ *by definition. Write* $A(n) = \operatorname{card}(B_n)$, *the number of elements in the set* B_n. *Then*

$$A(n) > \frac{1}{10} \frac{n}{\log n}$$

for all $n \geqq 2$.

Proof: We use some recent results of Livingston [11]. Let

$$(*) \qquad \psi(x; k, l) = \sum_{\substack{p^a \leqq x \\ p \text{ prime} \\ p^a \equiv l \bmod k}} \log p.$$

She has shown that

$$\psi(m; 4, 1) > \frac{2n}{5}$$

for all $n \geqq 37$. Thus it follows that for all $n \geqq 37$ there are at least

$$\frac{2}{5}\frac{n}{\log n}$$

terms in the sum $(*)$, i.e., prime powers $p^a \leqq n$ with $p^a \equiv 1 \bmod 4$. Since there are at most $\log_2 n$ prime powers smaller than or equal to n which are *not* relatively prime to n, we see that

$$A(n) > \frac{2}{5}\frac{n}{\log n} - \log_2 n.$$

Now it is easily verified that

$$\frac{2}{5}\frac{n}{\log n} - \log_2 n > \frac{1}{10}\frac{n}{\log n}$$

for all $n \geqq 104$; on the other hand, it is easily verified by explicit computation that

$$A(n) > \frac{1}{10}\frac{n}{\log n}$$

for $2 \leqq n \leqq 103$. This completes the proof of the lemma.

Thus by choosing w and z at random, we will find an integer $kr \bmod n$ of the required form with probability at least 1 in $10 \log n$. This completes the proof.

COMMENT. Pollard and Schnorr [15] have given an algorithm for solving the more general congruence $x^2 - Dy^2 \equiv k \bmod n$, which runs quickly assuming some unproved hypotheses. Also see [22], [6], [1].

Now we turn our attention to step (3) of the algorithm. We need to show that we can compute greatest common right divisors in $\mathbf{H}$ in polynomial time. Algorithms for computing the gcrd are given in many texts; for example, [9]. Unfortunately, however, if we follow these traditional presentations, it seems difficult to prove a polynomial bound for the run time. Hence, we alter the presentation somewhat.

THEOREM 3.4. *Given $g, h \in \mathbf{H}$, we can compute a greatest common right divisor f with all integer coordinates in polynomial time.*

Proof: We summarize our argument in a series of lemmas; the proofs given are only sketches.

LEMMA 3.5. *Given a point $x \in \mathbf{H}(\mathbf{Q})$, there exists $h \in \mathbf{H}$ satisfying $N(x - h) \leqq \frac{1}{2}$.*

Proof: Consider the vertices of the unit hypercube in $\mathbb{R}^4$, and the vertex at the center $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. It is easily verified that spheres of radius $\sqrt{2}/2$ centered at each of these 17 points cover the entire unit hypercube. Hence there is an $h \in \mathbf{H}$ satisfying $N(x - h) \leqq \frac{1}{2}$.

We could of course find the appropriate h by testing each of the 17 points relevant to the particular hypercube containing x. However, there is a simpler algorithm:

```
function Nearest-Integral-Quaternion($x$);
    for $n := 1$ to 4 do $h_n := \lfloor x_n + \frac{1}{2} \rfloor$;
    for $n := 1$ to 4 do
       if $x_n \geqq h_n$ then $r_n := h_n + \frac{1}{2}$
          else $r_n := h_n - \frac{1}{2}$;
    if $N(x - h) \leqq N(x - r)$ then return($h$)
    else return($r$);
```

Lemma 3.6. (Division algorithm for quaternions). *Given $h, d \in \mathbf{H}$, there exists $q, r \in \mathbf{H}$ with $h = qd + r$ and $N(r) \leqq \frac{1}{2}N(d)$.*

Proof: Let q be nearest-integral-quaternion (hd^{-1}). Put $r = h - qd$. Then from Lemma 3.5, we have

$$N(hd^{-1} - q) \leqq \tfrac{1}{2}$$

and therefore, by the multiplicativity of the norm, we get

$$N(r) = N(h - qd) \leqq \tfrac{1}{2}N(d).$$

Lemma 3.8. (Euclidean algorithm for **H**). *Given $g, h \in \mathbf{H}$, we can find a greatest common right divisor d in time which is a polynomial in* $\log_2 \max(N(g), N(h))$.

Proof: Consider the following algorithm:

```
function gcrd($g$, $h$);
while $g \neq 0$ do begin
    $t := g$;
    Using Lemma 3.6, write $g = qh + r$;
    $g := r$;
    $h := t$
    end;
return($h$);
```

It is easily verified that this algorithm actually produces a greatest common right divisor.

By Lemma 3.6, we reduce the norm by at least a factor of $\frac{1}{2}$ at each step, so in at most

$$\log_2 \max(N(g), N(h))$$

steps, the algorithm will terminate.

LEMMA 3.9. *Given $h \in \mathbf{H}$ with half-integer coordinates, there exists an associate εh with integer coordinates.*

Proof: The following algorithm produces the appropriate associate:

```
**function** Integral-Associate($h$);
**for** $n$ := 1 **to** 4 **do** $a_n := 2\left\lfloor \frac{h_n + 1}{2} \right\rfloor$;
**return**($h$ conj($h - a$));
```

For example, see [9], Theorem 371.

Combining Lemmas 3.5–3.9 proves Theorem 3.4.

It remains to verify the correctness of steps (3) and (4). By our choice of a and b, we know that $n | N(a + bi + j)$ and $N(a + bi + j) < n^2$. Since $N(g) | a^2 + b^2 + 1$ and $N(g) | n^2$, either $N(g) = n$ or $N(g) = kn$, where $k | n$. In the first case, we are done. In the second case, we obtain a non-trivial factorization of n and can continue the algorithm on each piece. Since this splitting of n can occur at most $\log_2 n$ times, the algorithm runs in polynomial time. Thus we have proved

THEOREM 3.10. *There is an algorithm for finding a representation $n = x^2 + y^2 + z^2 + w^2$ in random polynomial time.*

4. Expressing Numbers as the Sum of Three Squares

In this section, we discuss some methods for expressing n as the sum of three integer squares and as the sum of three triangular numbers.

Gauss proved that n can be expressed as the sum of three squares if and only if it is not of the form $4^a(8k + 7)$, where $a, k \geqq 0$. (From this, it also follows that every number can be expressed as the sum of three triangular numbers.) For an "elementary" proof, see [2] or [15].

If 4 divides n, then from a representation $\frac{1}{4}n = x^2 + y^2 + z^2$ we easily get $n = (2x)^2 + (2y)^2 + (2z)^2$. Thus without loss of generality, we may assume that $n \neq 0 \bmod 4$.

Our results can be summarized as follows:

(a) If p is a prime and $p \not\equiv 7 \bmod 8$, then we can express p as the sum of three squares in random polynomial time.

(b) If n is an integer, $n \neq 4^a(8k + 7)$, then there is a procedure that, *assuming some reasonable conjectures*, will produce a three-square representation in random polynomial time, for all n large enough. In practice, the procedure runs quickly.

(c) If $8n + 3$ is a prime, then there is a random polynomial time algorithm to express n as the sum of three triangular numbers. If $8n + 3$ is composite, then we can express n as the sum of three triangular numbers in random polynomial time, *assuming a reasonable conjecture.*

THEOREM 4.1. *If p is a prime, $p \not\equiv 7 \bmod 8$, then there is an algorithm to express p as the sum of three squares in random polynomial time.*

Proof: If $p = 2$ or p is of the form $4k + 1$, then we can express p as the sum of two squares (and hence as the sum of three squares) in a random polynomial time using the algorithms of Section 1. The remaining case is $p \equiv 3 \bmod 8$. In this case, we use the following facts:

(a) $\mathbb{Z}[\sqrt{-2}\,]$ is an effective Euclidean domain. (That is, we can find the gcd of two elements in polynomial time.)

(b) -2 is a quadratic residue of primes of the form $8k + 3$.

Using these facts, we can mimic the techniques of Section 2 and express p (of the form $8k + 3$) as $x^2 + 2y^2$ in random polynomial time. Then $p = x^2 + y^2 + y^2$. The details are left to the reader.

COROLLARY . *If $8n + 3$ is a prime, then we can express n as the sum of three triangular numbers in random polynomial time.*

Proof: Use Theorem 4.1 to express $8n + 3$ as the sum of three squares; then congruence conditions imply that each of the squares is odd. Hence,

$$8n + 3 = (2x + 1)^2 + (2y + 1)^2 + (2z + 1)^2$$

and so

$$n = \tfrac{1}{2}x(x + 1) + \tfrac{1}{2}y(y + 1) + \tfrac{1}{2}z(z + 1) = T(x) + T(y) + T(z).$$

This completes the proof.

In the case where n is not a prime, we use a trick similar to the one of Section 2. If $n \equiv 1$ or $2 \bmod 4$, we try to write n as the sum of a prime p and a square. Then it can be shown that either $p = 2$ or $p \equiv 1 \bmod 4$; in either case, we can express p as the sum of two squares in random polynomial time. If $n \equiv 3 \bmod 8$, then we try to write n as the sum of a square and *twice* a prime.

These techniques depend on the number of such representations, which are estimated by some reasonable conjectures. The first is Hardy and Littlewood's Conjecture H [8].

CONJECTURE 4.2. *Every sufficiently large number n is either a square or the sum of a prime and a square. The asymptotic behavior of the number $N(n)$ of representations is conjectured to be*

$$N(n) \sim \frac{\sqrt{n}}{\log n} C(n),$$

where $C(n)$ is defined by

$$C(n) = \prod_{p \text{ an odd prime}} 1 - \frac{\left(\frac{n}{p}\right)}{p-1}.$$

Some comments are in order. First, except for the infinite product term, this result is essentially what we would expect. For there are about $\sqrt{n}$ candidates for p, and the naive argument says each of these $\sqrt{n}$ candidates has about a one in $\log n$ chance of being prime.

Second, we would like to estimate the size of the infinite product term. Following the ideas in [13], it can be shown that, assuming the extended Riemann Hypothesis, there exists a constant A such that

$$C(n) > \frac{A}{\log\log n}.$$

Third, Conjecture H has been proved for "almost all" n. See [14].

In fact, the only non-squares congruent to 1 or 2 mod 4 and less than 1,000,000 which do *not* have representations as the sum of a square and a prime are:

$$5, 10, 13, 34, 37, 58, 61, 85, 130, 214, 226, 370, 526,$$

$$706, 730, 829, 1414, 1549, 1906, 2986, 7549, 9634.$$

These are probably *all* the exceptions.

The second conjecture we need is similar to the first; however, it has apparently not been explicitly stated before:

CONJECTURE 4.3. *Every number of the form $8k + 3$, $k \geq 1$, can be expressed as the sum of a square and twice a prime. The asymptotic behavior of the number of* *representation is conjectured to be*

$$M(k) \sim \frac{\sqrt{k}}{2\log k}C(k),$$

where $C(k)$ is the infinite product given in Conjecture 4.2.

The first statement of this conjecture has been verified for all $k \leqq 125{,}000$.

Assuming the truth of Conjectures 4.2 and 4.3, we can now give our algorithm for expressing n as the sum of three squares:

```
function three-squares($n$);
if $n = 0 \bmod 4$ then begin
    $(x, y, z)$ = three-squares($\frac{1}{4}n$);
    return($2x, 2y, 2z$);
    end
else if $n \equiv 7 \bmod 8$ then write('No representation!') and stop.
else if $n \equiv 3 \bmod 8$ then begin
    repeat
        $x :=$ random($\lfloor\sqrt{n}\rfloor$);
        $p := \frac{1}{2}(n - x^2)$
    until $p$ is a probable prime;
    $(y, z) :=$ two-squares($p$);
    return($x, y + z, y - z$)
    end
else if $n$ is a perfect squares, $n = d^2$,
    then return($d, 0, 0$)
else if $n \equiv 1 \bmod 4$ or $n \equiv 2 \bmod 4$ then begin
    repeat
        $x :=$ random($\lfloor\sqrt{n}\rfloor$);
        $p := n - x^2$
    until $p$ is a probable prime;
    $(y, z) :=$ two-squares($p$);
    return($x, y, z$)
    end;
```

THEOREM 4.4. *The preceding algorithm is correct and with high probability returns an expression of n as the sum of three squares, for all n sufficiently large.*

Proof: The only thing necessary to verify is that, in the two cases we are asked to express p as the sum of two squares, p is either 2 or of the form $4k + 1$.

First, the case $n \equiv 3 \bmod 8$; if we write

$$n = x^2 + 2p,$$

then by considering both sides mod 8 we see that $2p \equiv 2$, 3, or 7 mod 8. Thus $p \equiv 1$ or 5 mod 8, and therefore $p \equiv 1 \bmod 4$. Note that if $p = y^2 + z^2$, then $2p = (y+z)^2 + (y-z)^2$.

Second, the case $n \equiv 1$ or 2 mod 4: if we write

$$n = x^2 + p,$$

then by considering both sides mod 4 we see that $p \equiv 0$, 1, or 2 mod 4; so either $p = 2$ or $p \equiv 1 \bmod 4$. In either case, we can express p as the sum of two squares.

COROLLARY. *Assuming Conjectures* 4.2 *and* 4.3, *we can write n as the sum of three triangular numbers in random polynomial time.*

Acknowledgment. The research of the first author was supported by NSF Grant DCR-81-21431.

Bibliography

[1] Adleman, L. M., Estes, D., and McCurley, K. S., *Solving bivariate quadratic congruences in random polynomial time*, to appear.

[2] Ankeny, N. C., *Sums of three squares*, Proc. Amer. Math. Soc. 8 1957, pp. 316–319.

[3] Bach, E., Miller, G., and Shallit, J., *Sums of divisors, perfect numbers, and factoring*, Proc. 16th ACM Symposium on the Theory of Computing, 1984, pp. 183–190.

[4] Berlekamp, E. R., *Factoring polynomials over large finite fields*, Math. Comput., 24, 1970, pp. 713–735.

[5] Brillhart, John, *Note on representing a prime as a sum of two squares*, Math. Comp. 26 1972, 1011–1013.

[6] Estes, D., Adleman, L. M., Kompella, K., McCurley, K. S., and Miller, G. L., *Breaking the Ong-Schnorr-Shamir signature scheme for quadratic number fields*, to appear.

[7] Grace, J. H., *The four-square theorem*, J. Lond. Math. Soc. 2, 1927, pp. 3–8.

[8] Hardy, G. H., and Littlewood, J. E., *Some problems of 'partitio numerorum'; III: On the expression of a number as a sum of primes*, Acta Math. 44, 1923, pp. 1–70.

[9] Hardy, G. H., and Wright, E. M., *An Introduction to the Theory of Numbers*, Oxford, Clarendon Press, 1971.

[10] Kannan, R., *Lattices, basis reduction, and the shortest vector problem*, preprint.

[11] Livingston, M. L., *Explicit estimates for the ψ-function for primes in arithmetic progression*, S. I. U. E. Preprints in Mathematics #69, Southern Illinois University at Edwardsville, Edwardsville, IL, (March, 1986).

[12] Linnik, Ju. V., *An asymptotic formula in an additive problem of Hardy-Littlewood*, Izv. Akad. Nauk SSSR Ser. Mat. 24, 1960, pp. 629-706.

[13] Littlewood, J. E., *On the class-number of the corpus $P(\sqrt{-k})$*, Proc. London Math. Soc. 28, 1928, pp. 358–372.

[14] Miech, R. J., *On the equation $n = p + x^2$*, Trans. Amer. Math. Soc. 130, 1968, pp. 494–512.

[15] Mordell, L. J., *On the representation of a number as a sum of three squares*, Rev. Math. Pures Appl. 3, 1958, pp. 25–27.

[16] Pollard J. M., and Schnorr, C. P., *Solution of $x^2 + ky^2 \equiv m(\text{mod } n)$, with application to digital signatures*, to appear, Math. Comp.

[17] Rabin, M. O., *Probabilistic algorithms*, in *Algorithms and Complexity, Recent Results and New Direction* (J. F. Traub, ed.), Academic Press, New York, 1976, pp. 21–40.

[18] Rabin, M. O., *Efficient Algorithms*, Lecture Notes MIT, 1977, transcribed by M. Lui.

[19] Rabin, M. O., *Probabilistic tests for primality*, J. of Number Theory, 12, 1980, pp. 128–138.

[20] Rabin, M. O., *Probabilistic algorithms in finite fields*, SIAM J. on Computing, 9, 1980, pp. 273–280.

[21] Rabin, M. O., *Efficient Algorithms*, Lecture Notes Harvard University, 1980 transcribed by V. Hadzilacos.

[22] Shallit, J. O., *An exposition of Pollard's algorithm for quadratic congruences*, University of Chicago, Department of Computer Science, Technical Report 84-006, (Dec. 1984).

[23] Shallit, J. O., *Random polynomial time algorithms for sums of squares*, University of Chicago, Department of Computer Science, Technical Report 85-001, (Jan. 1985).

[24] Schoof, R. J., *Elliptic curves over finite fields and the computation of square roots* mod p, Math. of Comp., 44, 1985, pp. 483–494.

[25] Solovay, R. and Strassen, V., *A fast Monte-Carlo test for primality*, SIAM J. Comput. 6, 1977, pp. 84–85.

[26] Taussky, Olga, *Sums of squares*, Am. Math. Monthly 79, 1970, pp. 805–830.