# SUMS OF DIVISORS, PERFECT NUMBERS, AND FACTORING

(Extended Abstract)

*Eric Bach* (*)
*Computer Science Division*
*University of California*
*Berkeley, CA 94720*

*Gary Miller*
*Department of Applied Mathematics*
*Massachusetts Institute of Technology*
*Cambridge, MA 02139*

*Jeffrey Shallit* (*)
*Department of Computer Science*
*University of Chicago*
*Chicago, IL 60637*

*Abstract.*

Let $N$ be a positive integer, and let $\sigma(N)$ denote the sum of the positive integral divisors of $N$. We show computing $\sigma(N)$ is equivalent to factoring $N$ in the following sense: there is a random polynomial time algorithm that, given $\sigma(N)$, produces the prime factorization of $N$, and $\sigma(N)$ can be easily computed given the factorization of $N$.

We show that the same sort of result holds for $\sigma_k(N)$, the sum of the $k$-th powers of divisors of $N$.

We give three new examples of problems that are in Gill's complexity class BPP: {*perfect numbers*}, {*multiply perfect numbers*}, and {*amicable pairs*}. These are the first "natural" candidates for BPP - RP.

## I. Introduction.

Factoring is a well-known difficult problem whose precise computational complexity is still unknown, despite recent progress (see [Guy1], [Pol], [Dix]).

The relationship of factoring to other functions in number theory has also been explored. For example, Miller showed that if the Extended Riemann Hypothesis (ERH) is true, then $\phi(N)$ (the number of positive integers less than $N$ and relatively prime to $N$) is equivalent to factoring $N$ in the sense that a polynomial-time method for either problem gives one for the other [Mil]. He also demonstrated a similar equivalence for two other number-theoretic functions, $\lambda(N)$ and $\lambda'(N)$. Long pointed out that if one is willing to use randomization, the ERH assumption in the above results can be eliminated, and further showed that the calculation of orders in the multiplicative group of integers mod $N$ is randomly equivalent to factoring [Lon]. Using this last result, a method for composite-modulus discrete logarithm problems implies a method for factoring [Bac].

However, not every difficult number-theory function is equivalent to factoring; some are apparently *harder*. For example, remarks of Shanks indicate that factorization is reducible to finding the class number of an imaginary quadratic field [Sha] but no reduction in the other direction is known, nor is it even clear that this problem is in NP.

This paper continues in the above tradition and adds another well-known function to the list: $\sigma(N)$, the sum of the positive integral divisors of $N$. Clearly $\sigma(N)$ is readily computed from the factorization of $N$; if

$$N = p_1^{a_1} \cdots p_r^{a_r}$$

then $\sigma(N)$ is given by

$$\sigma(N) = \frac{p_1^{a_1+1} - 1}{p_1 - 1} \cdots \frac{p_r^{a_r+1} - 1}{p_r - 1}$$

$$= (p_1^{a_1} + \cdots + 1) \cdots (p_r^{a_r} + \cdots + 1). \quad (0)$$

We show how to split $N$, given $\sigma(N)$.

Recall that BPP is the class of languages recognized in polynomial time by a probabilistic Turing machine, with error probability bounded by a constant away from 1/2. We show that {perfect numbers}, {multiply perfect numbers}, and {amicable pairs} are in BPP.

## II. Splitting $N$ using $\sigma(N)$: the square-free case.

In this section we assume that

$$N = p_1 p_2 \cdots p_k$$

is the product of one or more distinct primes. This case is somewhat easier than the case where $N$ is divisible by a square, so we give our proofs in detail.

(*) Research sponsored in part by NSF grant MCS 82-04506.

The following procedure will state that $N$ is prime, or with high probability produce a non-trivial divisor of $N$.

(By iteration, if necessary, we eventually produce the complete factorization of $N$.)

Algorithm A.
[Given $\sigma(N)$ with $N$ squarefree, try to split $N$.]

A0. If $\sigma(N) = N + 1$, output $N$ and stop.

A1. If $N$ is even, output the factor $2^k$ and stop.

Repeat until $N$ splits:

A2. Run a single iteration of the randomized version of Miller's algorithm (as described by Long [Lon]), using $\sigma(N)$ as the exponent. If a non-trivial divisor of $N$ is produced, output that divisor and stop.

A3. Choose a random monic quadratic polynomial from $\mathbb{Z}_N[X]$, say, $f(X) = X^2 + bX + c$.

A4. Choose a random linear polynomial from $\mathbb{Z}_N[X]$, say, $r(X) = tX + u$ such that $t$ and $u$ are not both 0.

A5. [ Ensure that $r(X) \not\equiv 0 \pmod{q}$ for all primes $q \mid N$ ]. If $\gcd(t,N)$ splits $N$, output that divisor and stop.

A6. Compute $r^{\sigma(N)} \pmod{(f,N)} = dX + e$.

A7. If $\gcd(d,N)$ splits $N$, output that divisor and stop.

A8. If $\gcd(e-1,N)$ splits $N$, output that divisor and stop.

A9. [Failure.] No divisor of $N$ has been produced on this iteration.

In our analysis of Algorithm A, we will find the following group-theoretic lemma useful:

**Lemma A.**
Let $G$ be a finite cyclic group, $|G| = n$. Let $\sigma$ be the homomorphism defined by $\sigma(g) = g^r$. Then $\sigma(G)$ is also a finite cyclic group. We have

$$|\sigma(G)| = \frac{n}{\gcd(n,r)}$$

and if $g' \in \sigma(G)$ then $|\sigma^{-1}(g')| = \gcd(n,r)$. Hence $\sigma(G)$ is the trivial group iff $n \mid r$.

**Proof.**
Easy. Left to the reader.

Here are the ideas behind Algorithm A:

Steps A0 and A1 are self-explanatory.

In step A2, if for every $p_i$ dividing $N$ we have $p_i - 1 \mid \sigma(N)$ then the randomized version of Miller's algorithm will split $N$ in polynomial time.

Hence let us assume that for at least one $p_i$ (call it $p$) we have $p_i - 1 \nmid \sigma(N)$.

Suppose $f$ is a monic quadratic polynomial chosen at random from $\mathbb{Z}_N[X]$. Then a simple argument shows that with probability

$$\frac{1}{2} \cdot \frac{p-1}{p} \tag{1}$$

$f$ is irreducible (mod $p$); so assume it is. (In practice, of course, we choose many different $f$ and perform the algorithm on all of them. With high probability, the algorithm succeeds somewhere.)

Similarly, for a prime $q$, with probability

$$\frac{1}{2} \cdot \frac{q-1}{q} \tag{2}$$

$f$ splits as the product of distinct linear factors (mod $q$), say $f(X) = (X - \beta)(X - \gamma) \pmod{q}$, so assume it does for some $p_j \neq p_i$ ( call it $q$).

We now distinguish two cases:

Case I: $q - 1 \nmid \sigma(N)$.

Case II: $q - 1 \mid \sigma(N)$.

**Lemma B.**
Suppose Case I holds. (This corresponds to step A7 of the algorithm). Then with probability at least $\frac{1}{2}$, $\gcd(d,N)$ splits $N$.

**Proof.**
We show that we always have $d \equiv 0 \pmod{p}$ but $d \not\equiv 0 \pmod{q}$ with probability $\geq \frac{1}{2}$. From this we conclude that $\gcd(d,N)$ splits $N$ with probability $\geq \frac{1}{2}$.

To see that $d \equiv 0 \pmod{p}$ it is enough to see that

$$r(X)^{p+1} \pmod{f} \in \mathbb{Z}_p.$$

This is true since the field extension is as follows:

$$\begin{array}{ccc} r(X) & \in & \mathbb{Z}_p[X]/(f(X)) \\ \Big| & & \Big| \\ r(X)^{p+1} & \in & \mathbb{Z}_p \end{array}$$

Figure 1: Field extensions

Now the $p$-th power automorphism gives all the conjugates of the element $r(X)$. Since $f(X)$ is quadratic, there are only two conjugates: $r$ and $r^p$. Their product must lie in the base field $\mathbb{Z}_p$ since the product of conjugates is a norm (see [Mar]). Thus $d \equiv 0 \pmod{p}$.

Now let us show that $d \not\equiv 0$ with probability $\geq 1/2$. By the Chinese Remainder Theorem, we have the isomorphism

$$\mathbb{Z}_q[X]/(f(X)) \cong \mathbb{Z}_q[X]/(X-\beta) \oplus \mathbb{Z}_q[X]/(X-\gamma)$$

Indeed, we can make this isomorphism explicit. There exist fixed $w_1X + w_2$ and $v_1X + v_2 \in \mathbb{Z}_q[X]$ such that every linear $r(X) \in \mathbb{Z}_q[X]$ can be written uniquely as

$$r(X) \equiv c_1(w_1X + w_2) + c_2(v_1X + v_2) \pmod{q} \quad (3)$$

Here the $c_1$ and $c_2$ are in $\mathbb{Z}_q$ and depend on $r(X)$. Step A5 of the procedure above ensures that $c_1$ and $c_2$ are not both 0.

Now

$$r(X)^{\sigma(N)} \equiv c_1{}^{\sigma(N)}(w_1X + w_2) + c_2{}^{\sigma(N)}(v_1X + v_2) \pmod{q}$$

so

$$d \equiv c_1{}^{\sigma(N)}w_1 + c_2{}^{\sigma(N)}v_1 \pmod{q}$$

It is easy to see that $w_1$, $v_1 \not\equiv 0 \pmod{q}$, so if $d \equiv 0$ we must have

$$-c_1{}^{\sigma(N)}w_1v_1^{-1} \equiv c_2{}^{\sigma(N)} \pmod{q} \quad (4)$$

Now let us count the number of pairs $(c_1, c_2)$ for which this can happen.

If $c_1 \equiv 0$, then for (3) to hold we must have $c_2 \equiv 0$, and by step A5 this cannot happen. A similar argument holds if $c_2 \equiv 0$.

Now if both $c_1$, $c_2 \not\equiv 0 \pmod{q}$ then $c_1$, $c_2 \in \mathbb{Z}_q^*$, which is a cyclic group. Hence we may apply Lemma A to see that for any fixed value of $c_1$, the number of $c_2$ satisfying equation (1) is $\gcd(q-1, \sigma(N))$. But by the hypothesis for Case I, this is $\leq \frac{q-1}{2}$. Hence the total number of pairs for which (4) can hold is $\leq (q-1)^2/2$. Dividing this by $q^2 - 1$ total pairs $(c_1, c_2)$ with $c_1$, $c_2$ not both 0, we get $d \equiv 0 \pmod{q}$ with probability

$$\leq \frac{1}{2}\,\frac{q-1}{q+1}$$

Hence with probability $\geq \frac{1}{2}$, we have $d \not\equiv 0 \pmod{q}$.

This completes the proof of Lemma B.

Now we turn to case II, where $q - 1 \mid \sigma(N)$. We have the following Lemma:

**Lemma C.**

Suppose Case II holds. (This corresponds to step A8 of the algorithm). Then with probability at least $\frac{1}{2}\frac{q-1}{q+1}$, $\gcd(e-1,N)$ splits $N$.

**Proof.**

First, we show that, with high probability, we have $e \equiv 1 \pmod{q}$.

Equation (1) implies that

$$e \equiv c_1{}^{\sigma(N)}w_2 + c_2{}^{\sigma(N)}v_2 \pmod{q}$$

Now with high probability ( $\frac{q-1}{q+1}$ , to be precise ) neither $c_1$ nor $c_2$ equals 0, so let us assume this is true. Then both $c_1$ and $c_2$ lie in $\mathbb{Z}_q^*$, so both $c_1{}^{\sigma(N)}$ and $c_2{}^{\sigma(N)}$ must equal 1. Now it easily verified that $w_2 + v_2 \equiv 1 \pmod{q}$; hence $e \equiv 1 \pmod{q}$ with high probability.

Now let's consider what $e$ looks like (mod $p$). Considered mod $p$, $r(X)$ lies in the group of invertible elements of a finite field of $p^2$ elements. (It is invertible by the check in step A5). We have already shown that

$$r(X)^{\sigma(N)} \equiv dX + e \pmod{p}$$

and $d \equiv 0 \pmod{p}$. Using Lemma A, then, the probability that $e \equiv 1 \pmod{p}$ is

$$\frac{\gcd(p^2-1, \sigma(N))}{p^2-1}$$

But since $p - 1 \nmid \sigma(N)$, certainly $p^2 - 1 \nmid \sigma(N)$ and so $\gcd(p^2 - 1, \sigma(N)) \leq \frac{p^2-1}{2}$.

Thus $\gcd(e-1,N)$ splits $N$ with probability at least

$$\frac{1}{2}\,\frac{q-1}{q+1} \quad (6)$$

and the proof of Lemma C is complete.

Putting Lemmas B and C together, we get

**Theorem 1.**

With probability at least $\frac{1}{30}$, a single iteration of steps A2-A8 splits $N$.

**Proof.**

Putting together the worst cases above, we multiply the probabilities given in equations (1), (2), (6) with $p = 5, q = 3$ to get the result.

A brief remark is in order. Algorithm A will work even if we have a non-zero *multiple* of $\sigma(N)$ instead of $\sigma(N)$ itself. The only difference is that in step A0 we must use a polynomial-time prime test on $N$; for example, the probabilistic test given in [SS].

## III. Factoring $N$ using $\sigma(N)$: What to do about repeated factors.

This section serves two purposes: We generalize the algorithm in section (II) to the case when $N$ is not necessarily squarefree, and we show how to obtain the *complete* factorization of $N$, given $\sigma(N)$.

Before presenting the algorithm, we need some preliminary remarks on factorization. Ordinarily, we factor a number by splitting it into two pieces, then

splitting each of them, and continuing until we arrive at prime powers. For reasons that will become apparent later, we can only recursively split factors of $N$ that contain all the available copies of some prime $p$ dividing $N$. To refer to splittings that are useful in this sense, we say that a factorization $N = N_1 N_2 \cdots N_r$ *segregates* $p$ if $\nu_p(N_i) = \nu_p(N)$ for some $i$, where by $\nu_p(k)$ we mean the exponent of the highest power of $p$ that divides $N$.

A factorization segregates every prime if and only if the elements are pairwise relatively prime. The following procedure produces such a factorization.

Factor refinement procedure:

(At all times we have $N = M_1^{e_1} \cdots M_r^{e_r}$, possibly needing further processing).

R0. Let the initial factorization be $N = M_1 M_2$.

While factors remain with $\gcd(M_i, M_j) > 1$:

R1. Set $g = \gcd(M_i, M_j)$.

R2. Replace $M_i^{e_i}, M_j^{e_j}$ in the list by $g^{e_i + e_j}$, $(M_i/g)^{e_i}$, $(M_j/g)^{e_j}$.

R3. If necessary, remove units from the list and combine powers of equal numbers.

The properties of the refinement procedure are given by

**Lemma D.**
The factor refinement procedure terminates in at most $\log_2 N$ iterations, with all the $M_i$'s relatively prime. If the initial factorization is nontrivial and segregates some $p \mid N$, then on termination $r \geq 2$.

**Proof.**
Left to the reader.

Now assume that we want to split $N = p_1^{e_1} \cdots p_k^{e_k}$; the algorithm below uses a guess for one of the $\alpha_i$'s, say $\alpha$. Since $\alpha_i \leq \log_2 N$, we can try all possible $\alpha$'s without spoiling the polynomial time bound.

Algorithm B:
[ Try to split $N$ given $\sigma(N)$ and $\alpha$ ].

B0. If $N$ is a prime power, output $N = p^k$ and stop.

B1. If $N$ is even, output a relatively prime factorization $N = 2^k \cdot M$ and stop.

Repeat:

B2. Pick a random monic $f \in \mathbb{Z}_N[X]$ of degree $\alpha + 1$.

B3. Pick a random $r \in \mathbb{Z}_N[X]$ of degree $\leq \alpha$.

B4. Compute $r^{\sigma(N)} \bmod f = d_\alpha X^\alpha + \cdots + d_1 X + e$.

B5. For each $i$, $1 \leq i \leq \alpha$, let $g_i = \gcd(d_i, N)$.

B6. Let $h = \gcd(e - 1, N)$.

B7. Try to split $N$ using $\sigma(N)$ as an exponent for $\mathbb{Z}_n^*$, yielding a possibly trivial factor $l$.

B8. If any of the factors $g_i$, $h$, or $l$ lead to a relatively prime factorization, output the results and stop.

We now prove that with reasonable probability, Algorithm B produces a relatively prime factorization.

First, note that if $N$ is a prime power or even, we get a good factorization. It can also be shown that if

$$\operatorname{lcm}(p_1 - 1, \cdots, p_r - 1) \mid \sigma(N)$$

then with probability at least 1/2 Miller's algorithm splits $N$ and segregates one of the primes dividing $N$. Therefore we may as well assume that $N$ is odd, has at least two prime factors, and there is some $p \mid N$ with $p - 1 \nmid \sigma(N)$. Let $q$ be any other prime dividing $N$.

We hope that $f$ is irreducible mod $p$, but reducible and squarefree mod $q$; if this is the case, we call $f$ **suitable**. We then have

$$\mathbb{Z}_q[X]/(f) \cong GF(q^{k_1}) \oplus \cdots \oplus GF(q^{k_r}) \qquad (7)$$

with $r \geq 2$. There is a nice relation between the representations on both sides of this isomorphism, given by

**Lemma E.**
If $a(X) \in \mathbb{Z}_q[X]/(f)$ has degree $\leq \alpha$, then $a$'s nonconstant terms vanish identically if and only if all the projections are equal and are contained in $GF(q)$.

**Proof.**
Each condition is satisfied by $q$ elements, and the first condition implies the second.

We now need some probability estimates:

**Lemma F.**

A monic polynomial $f$ of degree $a+1$ is suitable with probability at least

$$\frac{1}{\alpha+1}\left(1-\frac{1}{\sqrt{p}}\right)\left(1-\frac{1}{q}-\frac{1}{\alpha+1}\right)$$

**Proof.**

First, $f$ is irreducible mod $p$ with probability at least

$$\frac{1}{\alpha+1}\left(1-\frac{1}{\sqrt{p}}\right)$$

(see [Ber, p.80]). Second, $f$ is irreducible mod $q$ with probability at most $\frac{1}{\alpha+1}$, and has a repeated factor with probability exactly $\frac{1}{q}$ (see [Car]).

We also want $r(X)$ not to vanish mod $q$; the chances of this are estimated by the following lemma.

**Lemma G.**

Given that $f$ is suitable, $r(x) \not\equiv 0 \pmod{q}$ with probability at least $1-\frac{1}{q^2}$.

**Proof.**

Clear.

Now assume that $p^\alpha \| N$, $f$ is suitable, and $r \not\equiv 0 \pmod{q}$. We are interested in two events, either one of which gives us a splitting that segregates some prime:

B5 succeeds: there is some $i$ for which $q \nmid d_i$. (Note that $p \mid d_i$ for all $i$, since in analogy with Figure 1 of section II, the map $x \to x^{\sigma(N)}$ takes $GF(p^{\alpha+1})$ into $GF(p)$).

B6 succeeds: $q \mid e-1$, but $p \nmid e-1$.

Now let $(c_1, c_2, \cdots, c_r)$ denote the projections of $r^{\sigma(N)}$ given by the isomorphism (7) above; for convenience assume that all the $c_i$'s are contained in some common field. Since by Lemma E, B5 always succeeds when some but not all $c_i = 0$, we have

$$\mathbf{Pr}\,[\text{ B5 or B6 succeeds }] \geq$$

$$\mathbf{Pr}\,[\text{ B5 or B6 succeeds } |\text{ all } c_i \neq 0\,]$$

Because of this, we may as well assume that all $c_i \not\equiv 0$; as before, consider two cases.

Case 1: Raising to the $\sigma(N)$ power does not annihilate $(\mathbb{Z}_q[X]/(f))^*$; then the image of this homomorphism is a direct product of cyclic groups, say

$$C_1 \otimes C_2 \otimes \cdots \otimes C_r$$

and we may assume, without loss of generality, that $C_1$ is non-trivial. If $C_1 \cap C_2 \neq C_1$, or if $C_1 \cap C_2 \neq C_2$, then $c_1$ and $c_2$ are distinct with probability at least $1/2$, because they have to be in $C_1 \cap C_2$ to be equal. Thus assume $C_1 = C_2$. Then the probability that $c_1 = c_2$ is

$$\frac{1}{\#(C_1 \cap C_2)} \geq 1/2.$$

In this case, by Lemma E, B5 succeeds with probability at least $1/2$.

Case 2: Raising to the $\sigma(N)$ power annihilates $(\mathbb{Z}_q[X]/(f))^*$; then by the hypothesis on $p$, the image of this homomorphism is non-trivial mod $p$ ( $p-1$ doesn't divide $\sigma(N)$, so $p^{\alpha+1}-1$ doesn't either ). Then B6 is successful with probability at least $1/2$.

Using lemmas F and G and the above discussion, we have

**Lemma H.**

For some $\alpha \leq \log_2 N$, the above algorithm produces a relatively prime factorization of $N$ with probability at least

$$\frac{1}{33(\alpha+1)}.$$

We observe that, as in the previous section, $\sigma(N)$ can be replaced by any multiple of $\sigma(N)$ with no change in the above result. Since $\sigma(mn) = \sigma(m)\,\sigma(n)$ for relatively prime $m$ and $n$, we can use $\sigma(N)$ to factor the relatively prime pieces output by the algorithm until we reach a prime power. Thus we get

**Theorem 2.**

Given $\sigma(N)$, we can produce the complete factorization of $N$ in random polynomial time.

An interesting corollary is that the function $r_4(N)$, the number of ways to write $N$ as the sum of four integer squares, is (randomly) equivalent to factoring. This follows easily from a theorem of Lagrange (see, for example, [HW, Theorem 386]).

## IV. Generalizations to $\sigma_k(N)$.

A natural generalization of $\sigma(N)$ is summing the $k$-th powers of divisors of $N$, i. e.

$$\sigma_k(N) = \sum_{d \mid N} d^k$$

$$= (p_1^{ke_1} + \cdots + p_1^k + 1)\cdots(p_r^{ke_r} + \cdots + p_r^k + 1)$$

where $N = p_1^{e_1} \cdots p_r^{e_r}$.

We have

**Theorem 3.**

Computing $\sigma_k(N)$ is (randomly) equivalent to factoring, for any fixed integer $k \neq 0$.

**Proof (sketch).**

If $k$ is negative then

$$\sigma_k(N) = N^k \sigma_{-k}(N)$$

so it suffices to consider positive $k$.

The essential idea is that the map $x \to x^{\sigma_k(N)}$ takes $GF(p^{k(\alpha+1)})$ into $GF(p^k)$, when $p^\alpha \| N$.

Algorithm C.
[ Try to split $N$ given $\sigma_k(N)$.]

C0. If $N$ is even or a prime power, output a factor and stop.

Set $\alpha \leftarrow 1$, and repeat until $N$ splits:

C1. Try to split $N$ using $\sigma_k(N)$ as an exponent for $\mathbb{Z}_N^*$.

C2. [ Construct $GF(p^k)$. ] Pick a random monic $a \in \mathbb{Z}_N[Y]$ of degree $k$; let $R$ denote $\mathbb{Z}_N[Y]/(a)$.

C3. Pick a random monic $f \in R[X]$ of degree $\alpha + 1$.

C4. Pick a random $r \in R[X]$ of degree $\leq \alpha$.

C5. Compute $r^{\sigma(N)} \pmod f =$
$d_\alpha(Y) X^\alpha + \cdots + d_1(Y)X + e(Y)$.

C6. For each $i$, $1 \leq i \leq \alpha$, and each coefficient $t$ of $d_i(Y)$, see if $\gcd(t,N)$ splits $N$.

C7. See if $\gcd(e(0) - 1, N)$ splits $N$.

C8. [Failure]. If $\alpha + 1 < \log_3 N$, set $\alpha \leftarrow \alpha + 1$; else set $\alpha \leftarrow 1$.

There is only one new observation to make here: we want $a(Y)$ to be irreducible modulo two distinct divisors of $N$, and this happens with probability about $1/k^2$. Since $k \leq \log_2 \sigma_k(N)$, we only expect to wait a polynomial-bounded time until this happens. In all other respects, Algorithm C behaves just like algorithm B. The details are left to the reader.

## V. Some classes of numbers that can be factored quickly.

The reduction of factoring to computing $\sigma(N)$ discussed in the previous sections is interesting from a complexity theory point of view, but it also has some practical applications: it allows us to quickly factor those numbers $N$ for which $\sigma(N)$ is easily computable.

Consider the equation $\sigma(N) = 2N$. Numbers satisfying this equation are known as **perfect numbers**; there is an enormous literature about such numbers dating back as far as Euclid. If we define $s(N)$ to be the sum of the "aliquot parts" of a number $N$, i. e. the sum of all divisors of $N$ except $N$ itself, then for perfect $k$ we have $s(k) = k$; the ancients read mystic significance into the fact that a perfect number exactly equalled the sum of its "parts".

Even perfect numbers are exactly those of the form $2^{n-1}(2^n - 1)$ where $2^n - 1$ is prime; Euclid proved that this condition is sufficient, and the necessity was proved a few millenia later by Euler. No one knows if there are any odd perfect numbers, but if there are, they must satisfy many stringent conditions (see, e.g., [teR]). We now add one more: they are all easy to factor!

Looking at this in another way, we can prove that the set {perfect numbers} is recognizable in (two-sided) random polynomial time. By {perfect numbers}, of course, we mean the set

$$\{ x \in (0, 1)^* : x \text{ (interpreted in binary) is perfect } \}.$$

Given $N$, *assume* that $\sigma(N) = 2N$. Run the algorithm of section III with the appropriate polynomial time bound; the result is a (purportedly complete) factorization of $N$. Now check to see if $N$ is indeed perfect by using equation (0).

We end up accepting if $N$ is perfect, or if we accidentally produced an incorrect factorization (i. e. one where our probabilistic prime test said all the factors were prime, but some really weren't). But such an accident happens only $\epsilon$ of the time, and we can fix $\epsilon$ ahead of time.

We end up rejecting if $N$ is not perfect, or if we accidentally produced an incorrect factorization as above, or if the algorithm of section III failed to produce any factorization at all in our (pre-fixed) time bound. Again, this happens only $\epsilon$ of the time.

Gill's complexity class BPP denotes the class of languages that are recognized by probabilistic Turing machines in polynomial time, with a (two-sided) probability of error bounded by a constant away from 1/2. Thus the above remarks show

**Theorem 4.**

{perfect numbers} $\in$ BPP.

For discussion of random complexity classes, see [Gil].

Theorem 4 gives the first "natural" candidate for BPP - RP. Of course, it is possible to construct examples like

$$L = \{\, x \# y : x \text{ is prime and } y \text{ is composite} \,\}.$$

$L \in$ BPP, but it is somewhat "artificial", since it may be written as the product of two languages, one of which is known to be in RP, and one which is known to be in co-RP.

Nevertheless, Theorem 4 may in fact be less interesting than it appears at first glance; if there are no odd perfect numbers (as is widely believed), then the clever Lucas-Lehmer test (see [Knu]) combined with the Euclid-Euler result for even perfect numbers gives a *deterministic* polynomial time algorithm to recognize the language {perfect numbers}.

However, there are well-studied generalizations of perfect numbers for which no deterministic tests are known. For example, numbers such that $\sigma(N) = 3N$ are sometimes called **sous-doubles**; examples are 120 and 672. It is easy to see that an argument like that in Theorem 4 shows that {sous-doubles} $\in$ BPP.

A still-larger class is {**multiply perfect numbers**}; i. e., those numbers $N$ for which $N \mid \sigma(N)$. To show that {multiply perfect numbers} $\in$ BPP, we need the following lemma:

**Lemma J.**

$$\sigma(N) < 5N \log\log N \quad \text{for } N \geq 3.$$

**Proof.**

A well-known theorem (e. g. [HW, Thm. 329]) states that

$$\frac{\sigma(N)\,\phi(N)}{N^2} \leq 1$$

A result of Rosser and Schoenfeld [RS] is

$$\frac{N}{\phi(N)} < e^C \log\log N + \frac{3}{\log\log N}$$

for $N \geq 3$. Here $C$ is Euler's constant.

Combining these two inequalities, we get

$$\frac{\sigma(N)}{N} < (e^C + 3) \log\log N$$

for $N > e^e$. From this the result easily follows.

Lemma J shows that we can determine if $N$ is multiply perfect with fewer than 5 log log $N$ iterations of Algorithm B. This can be done in random polynomial time, so we have proved

**Theorem 5.**

{multiply perfect numbers} $\in$ BPP.

The multiply perfect numbers less than $10^9$ are as follows:

1, 6, 28, 120, 496, 672, 8128, 30240, 32760,
523776, 2178540, 23569920, 33550336, 45532800,
142990848, 459818240.

See, for example, [Carm]. It is not known whether or not there are an infinite number of multiply perfect numbers. However, there are some density results that give upper bounds; for example, Hornfeck and Wirsing have shown [HoW] that if $m(x)$ denotes the number of multiply perfect numbers $\leq x$, then

$$m(x) = O(e^{\frac{c \log x \log\log\log x}{\log\log x}})$$

To give still another example, consider the pairs $(M, N)$ such that

$$\sigma(M) = \sigma(N) = M + N$$

Such numbers are known as **amicable pairs**. Jacob gave Esau 220 goats and 220 sheep [Gen], and some scholars have interpreted this as showing that the Hebrews knew about $\sigma(N)$. Amicable pairs also have an enormous amount of literature (see [LM]). An argument similar to those above gives

**Theorem 6.**

{amicable pairs} $\in$ BPP.

It is not known whether or not there are an infinite number of amicable pairs $(M, N)$, but Erdős conjectures that the number of such pairs, $A(x)$, with $M < N < x$ is at least $cx^{1-\epsilon}$ [Guy2].

Using our methods, it is possible to show that many other types of numbers (for example the "betrothed numbers" of Isaacs [Guy2, p. 33]) can be recognized in two-sided random polynomial time.

In Theorems 4-6 above, we have given three problems in BPP. The two-sidedness of these problems is due to the dependence on primality testing; if we had a deterministic polynomial-time prime test, we would be able to show that {perfect numbers}, {multiply perfect numbers}, and {amicable numbers} are in RP. No such prime test is currently known, although there is an "almost-polynomial-time" one [APR].

## VI. Acknowledgements.

We are pleased to acknowledge the use of the computer algebra program VAXIMA, which allowed us to

confront our early ideas with the harsh reality of specific examples.

We would also like to express our deep appreciation to Manuel Blum, who created an environment eminently suitable to conducting research.

## REFERENCES

[APR] Leonard M. Adleman, Carl Pomerance, and Robert S. Rumely, "On distinguishing prime numbers from composite numbers", *Ann. Math.* **117** (1983) 173-206.

[Bac] Eric Bach, "Discrete logarithms and factoring", to appear.

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

[Carl] Leonard Carlitz, "The arithmetic of polynomials in a Galois field", *Amer. Journ. Math.*, **54** (1932) 39-50.

[Carm] R. D. Carmichael, "A table of multiply perfect numbers", *Bull. Amer. Math. Soc.* **13** (1907) 383-386.

[Dix] J. D. Dixon, "Asymptotically fast factorization of integers", *Math. Comp.* **36** (1981) 255-260.

[Gen] *Genesis*, xxxii, 14.

[Gil] John Gill, "Computational complexity of probabilistic Turing machines", *Siam J. Comput.* **6** (1977) 675-695.

[Guy1] Richard K. Guy, "How to factor a number", *Proc. Fifth Manitoba Conf. on Numerical Math.*, Winnipeg, 1976, 49-89.

[Guy2] Richard K. Guy, *Unsolved Problems in Number Theory*, Springer-Verlag, New York, 1981.

[HoW] Bernhard Hornfeck and Eduard Wirsing, "Über die Haufigkeit vollkommener Zahlen", *Math. Annalen* **133** (1957) 431-438.

[HW] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*, Oxford, 1971.

[Knu] Donald E. Knuth, *The Art of Computer Programming*, V. II (Seminumerical Algorithms), 2nd edition, Addison-Wesley, Reading, Mass. (1981) 391-394.

[LM] Elvin J. Lee and Joseph S. Madachy, "The history and discovery of amicable numbers", *Journ. Rec. Math.* **5** (1972) 77-93, 153-173, 231-249.

[Lon] Douglas L. Long, "Random equivalence of factorization and computation of orders", to appear, *Theoretical Computer Science.*

[Mar] Daniel A. Marcus, *Number Fields*, Springer-Verlag, New York, 1977.

[Mil] Gary Miller, "Riemann's hypothesis and tests for primality", *J. Comp. System Sci.* **13** (1976) 300-317.

[Pol] J. M. Pollard, "Theorems on factorization and primality testing", *Proc. Cambridge Phil. Soc.***76** (1974) 521-528.

[RS] J. Barkley Rosser and Lowell Schoenfeld, "Approximate formulas for some functions of prime numbers", *Ill. Journ. Math.* **6** (1962) 64-94.

[Sha] Daniel Shanks, "Class number, a theory of factorization, and genera", *Proceedings of Symposia in Pure Mathematics*, V. 20 (1969 Number Theory Institute), American Mathematical Society (1971) 415-440.

[SS] R. Solovay and V. Strassen, "A fast Monte-Carlo test for primality", *Siam J. Computing* **6** (1977) 84-5.

[teR] H. J. J. te Riele, "Perfect numbers and aliquot sequences", in Computational Methods in Number Theory, Amsterdam Math. Centre Tracts **154** (1982) 141-157.