hep-th/0503159

Fast Factoring of Integers

Gordon Chalmers

e-mail: gordon@quartz.shango.com

Abstract

An algorithm is given to factor an integer with N digits in $\ln^m N$ steps, with m approximately 4 or 5. Textbook quadratic sieve methods are exponentially slower. An improvement with the aid of an a particular function would provide a further exponential speedup.

Factorization of large integers is important to many areas of pure mathematics and has practical applications in applied math including cryptography. This subject has been under intense study for many years [1]; improvements in the methodology are especially desired for computational reasons.

Given an integer N composed of approximately $\ln_{10} N$ digits, standard text-book quadratic sieve methods generate the factorization of the number into primes in roughly

$$e^{a\sqrt{\ln N \ln\ln N}} \tag{1}$$

moves. The steps require manipulations of large integers, of the size N, with bit complexity of approximate $\ln N$. The number a is approximately 2, depending on the variant used [1].

The presentation in this work generates a computational method to obtain the prime factorization in

$$\ln^m N \tag{2}$$

moves with integers of the same size. The factor m is specified by the convergence of the solution to a set of polynomial equations in $\ln N$ variables, which numerically is approximately $m = 3$, after the root selection is chosen from small numbers to large (see, e.g. [2]).

Given a function C_N that counts the number of prime factors of a number, i.e.

$$N = \prod_{j=1}^{r} p_{\sigma(j)}^{k_j} \qquad C_N = \sum_{i=1}^{r} k_j \ , \tag{3}$$

the factorization of the number N could be performed in approximately C_N^m steps. The bound on the number of prime factors of an integer N is set by $\ln_2 N$, the product of the smallest prime number 2. The number of primes smaller than a number N is approximately $N/\ln N$, and the C_N is roughly $\ln\ln N$. Hence, given the function C_N a further exponential improvement is generically given. However, this function would drastically simplify the factorization of large numbers possessing only a few prime factors. The upper bound of $C_N \sim \ln N$ describes the case discussed in the previous paragraph.

Consider a number with exactly C_N factors. This number projects in base x onto the form,

$$N = \sum_{i=0}^{C_N} a_i x^i \ . \tag{4}$$

The polynomial form in (4) admits a product form,

$$N = \prod_{i=1}^{C_N} (c_j x - b_j) \ , \tag{5}$$

with $c_j x - b_j$ integral. The number scales into the form,

$$N = \gamma \prod_{i=1}^{C_N} (\alpha_j x - 1) \ , \tag{6}$$

in which there are C_N numbers α_j and a number γ. The same integer has the prime factorization

$$N = \prod_{i=1}^{C_N} p_{\sigma(i)} \ , \tag{7}$$

with the set $\sigma(j)$ containing possible redundancy, for example, $p_{\sigma(1)} = p_{\sigma(2)} = 2$. Given an integer base x the solution to the numbers b_j generate the prime factors, *as long as the value* C_N *is correct.*

Two examples are given. First, $15 = 3^2 + 2(3) = x(x+2)$, which solves for the prime factors 3 and 5. Second, $10 = 2^3 + 2 = x(x^2+1)$, which solves for the prime factors 2 and 5. In the second example, even though there are two factors, the polynomial is a cubic with a vanishing zeroth order term; the origin of the cubic is that there is a complex root.

The polynomial base form of the number, i.e. $N = \sum a_i x^i$, will not in all cases factor into the form (5) with real coeffcients c_j and b_j. However, because the coefficients are real, the roots will enter in complex conjugate pairs. The product of these complex conjugates form a positive number. In order to test all possible cases, including the presence of a factor being represented as a product of two complex roots, all numbers from C_N to $2C_N$ should be examined. Potential complex roots come in pairs, and the maximum number of factors could take on the form of C_N products of two complex numbers. The cost of the additional complexity is of order unity.

The expansion of (5) generates a set of algebraic equations relating the integer coefficients b_j to those in a_j. The form is,

$$\begin{aligned} \gamma\alpha_1\alpha_2\ldots\alpha_{C_N} &= a_{C_N} \\ &\ldots \\ \gamma &= a_0 \ , \end{aligned} \tag{8}$$

in which the combinations

$$c_j x - b_j \tag{9}$$

must converge to integers or into pairs with the product being an integer, and for the maximal factorization to prime numbers p (of which there are an approximate $N/\ln N$ of them for the number N). The determination of the numbers α_j must be rational as $(c_j/b_j x - 1) = n/b_j$, with n integral. The other case of interest is when c_j/b_j is complex and the relevant condition is $|(c_j/b_j x - 1)|^2 = n^2/b_j^2$; this is not satisified in general by complex rational numbers. However, one may square the number N and then all terms in the product must be rational.

The number γ must be an integer or the square of γ must be an integer, according to the presence of complex terms (roots) which square to an integer. If the solution does not satisfy these criteria, then there is not a valid factorization N into integers. Given rational solutions $\alpha_j = c_j/b_j$ and the $\gamma = \prod b_j$, the straightforward multiplication of γ into the C_N factors generates the factorization into $N_1 N_2 \ldots N_{C_N}$, via eliminating the denominators in the individual terms of the rational numbers. The complex root case allows the numbers to be determined as $N_j = N_{j,+}N_{j,-}$.

Solving these equations generates the prime factorization of the integer N into the set of primes $p_{\sigma(j)}$. Numerically, solving a set of equations in n variables typically has convergence of n^3 if the initial starting values are chosen correctly.

In the case of C_N not known, but bounded by $\ln N$, all cases of interest from the test cases of $\tilde{C}_N = 1$ to $\tilde{C}_N = \ln N$ may be examined, at the cost of duplicating the process by the bound $\ln N$. Typical true values of C_N are expected for generic numbers to be smaller than the bound, e.g. $\ln\ln N$. The cases from C_N to $2C_N$ must also be examined in order to take into account the pairs of complex conjugates.

Computationally exploring all of the cases from $\tilde{C}_N = 1$ to $\tilde{C}_N = 2\ln N$ (e.g. $\sim C_N$) for integer bases x finds all product forms of the integer N into products

$$N = N_1 N_2 \qquad N = N_1 N_2 N_3 \tag{10}$$

$$N = N_1 N_2 \dots N_{C_N} = \prod_{j=1}^{C_N} p_{\sigma(j)} \ . \tag{11}$$

Solving for b_j and c_j (e.g. $\alpha_j = b_j/c_j$) in terms of a_j generates either integral values or non-integral values, or pairs or complex conjugates. In the case of integral values for all the b_j parameters, the base x is resubstituted into the factors $c_j x - b_j$ of the total product,

$$N = \prod^{\tilde{C}_N} (c_j x - b_j) = \gamma \prod (\alpha_j x - 1) \ , \tag{12}$$

to find the values of the individual N_n. The integral solutions generate the various factorizations. In the case of complex conjugate pairs, the integrality is tested by multiplying the individual terms,

$$(c_j x - b_j)(c_j^* x - b_j^*) \sim (\alpha_j x - 1)(\alpha_j^* x - 1) \ , \tag{13}$$

and determining if it is an integral (the latter has to be rational). The maximum $\tilde{C}_N$ that results in integral values of $c_j x - b_j$ gives the prime factorization. Non-integral solutions to $c_j x - b_j$ do not generate integer factorization of the number N into $\tilde{C}_N$ numbers.

In the computation for the roots, the parameters α_j and γ are determined. The integrality of the $c_j x - b_j$ translates into α_j being a rational number (when not a complex root allowing the complex $c_j x - b_j$ square to be an integer). The rationality allows the denominators of all the α_j parameters to be extracted and used to multiply the prefactor γ. The case of the complex roots may be examined also by first taking the complex square and examining for integrality; the denominators must also be taken out of the products.

Computationally testing all cases from $\tilde{C}_N = 1$ to $\ln N$ finds all product forms of the potentially large integer N. The factorization process entails three steps: 1) projecting the number N into base x, 2) solving the system of algebraic equations for integral $c_j x - b_j$, 3) substituting the base x back into the $c_j x - b_j$ for factor determination.

Computationally the first step requires specifying the base x and projecting the number onto it. The base x is specified by the two equations,

$$\ln x > \frac{\ln N}{\tilde{C}_N + 1} \qquad \ln x \leq \frac{\ln N}{\tilde{C}_N} \ . \tag{14}$$

Following the determination of x, the coefficients a_j are determined via starting with $a_{\tilde{C}_N} \leq x$, $j = \tilde{C}_N$, $N_{\tilde{C}_N} = N$ and following the procedure,

$$N_{j-1} = N_j - a_j x^j \tag{15}$$

with

$$\gamma_{j-1} = N_{j-1}/x^{j-1} \ . \tag{16}$$

Take $a_{j-1} = [\gamma_{j-1}]$ with a rounding down of γ_{j-1}; if $a_{j-1} \geq x$ then the procedure stops and the remaining a_i are set to $a_i = 0$ with $i < j-1$. Otherwise, the subtraction process continues. The procedure costs at most $3\tilde{C}_N$ operations with numbers of at most size N (bit size $\ln N$). Due to the bound on $\tilde{C}_N$, this process is at most of the size $3 \ln N$ operations, one of which is division.

The next step requires the solution of the algebraic equations for b_j in terms of a_j. There are $\tilde{C}_N + 1$ equations in $\tilde{C}_N + 1$ variables. The initial roots are chosen from the lowest prime neighborhood around $p = c_j x - b_j$, to larger. This procedure is natural for the root determinations in the case of the exact C_N. Another case is in choosing cascades in the range 10^n to 10^{n+1} for $n \leq \ln_{10} N$. Convergence is not analyzed, but for well chosen starting values, the number of iterations is typically N_v^3; this is $\tilde{C}_N^3$ for the case of $\tilde{C}_N$ variables and equations. The bound is $\ln^3 N$. Roughly, if the number of operations per iteration is $\tilde{C}_N^2$ (i.e. evaluating a set of similar polynomials) and there are $\tilde{C}_N$ roots, then the steps would number as $\tilde{C}_N^6$.

If any roots converge to a non-integral value of b_j, then the integer N does not factor into $\tilde{C}_N$ numbers. This shortens the number of iterations and steps. The process of determining the factorization of N into the products of two to C_N numbers requires $\ln^m N$ steps, with m denoting an average value from the root selection process, the number of variables at each step, the root solving and iteration process including shortcuts such as information from lower $\tilde{C}_N$ examples, and an averaging of the shortening the algorithm during the process of lower unknown roots or non-integerness. Perhaps, the average results in $m \sim 4$ or 5, less than $\ln^6 N$.

To compare with the textbook quadratic sieve method, take the logarithm of the steps for both this method and the former,

$$a\sqrt{\ln N \ln\ln N} \qquad m \ln\ln N \tag{17}$$

which is,

$$a^2 \ln N \ln\ln N \qquad m^2 \ln^2 \ln N \ . \tag{18}$$

The gain is clearly an exponential. The terms compare as $a^2 \ln N = m^2 \ln\ln N'$ and $N' = \exp\exp(a^2/m^2 N)$. Consider $N = 10^{1000}$: the numbers are an approximate $\exp\,(a^2 12000)^{1/2}$ vs. $\exp\,(m6)^{1/2}$.

In addition to prime factorization, the product form of the integer into various products of factors is determined; this is an additional byproduct of the procedure and its computational cost. Furthermore, an explicit knowlege of C_N, the number of prime factors of a number, would provide a further exponential speedup.

The procedure here may be adapted to find various forms of number decompositions. An example is to find the form of a number written as a sum of products of primes.

References

[1] R. Crandall and C. Pomerance, *Prime Numbers, A Computational Perspective*, Springer-Verlag Inc., (2001).

[2] *Encylopedic Dictionary of Mathematics*, Iwanamic Shoten Publishes, Tokyo, 3rd Ed., (1985), English Transl. MIT Press (1993).