4. In the continued fraction algorithm explain why there is no need to include in the factor base B any primes p such that $(\frac{n}{p}) = -1$.
5. Following Examples 2 and 3, use the continued fraction algorithm to factor the following numbers: (a) 9509; (b) 13561; (c) 8777; (d) 14429; (e) 12403; (f) 14527; (g) 10123; (h) 12449; (i) 9353; (j) 25511; (k) 17873.

References for § V.4

1. H. Davenport, *The Higher Arithmetic*, 5th ed., Cambridge Univ. Press, 1982.
2. D. Knuth, *The Art of Computer Programming*, Vol. 2, Addison-Wesley, 1973.
3. D. H. Lehmer and R. E. Powers, "On factoring large numbers," *Bull. Amer. Math. Soc.* **37** (1931), 770–776.
4. M. A. Morrison and J. Brillhart, "A method of factoring and the factorization of F_7," *Math. Comp.* **29** (1975), 183–205.
5. C. Pomerance and S. S. Wagstaff, Jr., "Implementation of the continued fraction integer factoring algorithm," *Proc. 12th Winnipeg Conference on Numerical Methods and Computing*, 1983.
6. M. C. Wunderlich, "A running time analysis of Brillhart's continued fraction factoring method," *Number Theory, Carbondale 1979*, Springer Lecture Notes Vol. 751 (1979), 328–342.
7. M. C. Wunderlich, "Implementing the continued fraction factoring algorithm on parallel machines," *Math. Comp.* **44** (1985), 251–260.

5 The quadratic sieve method

The quadratic sieve method for factoring large integers, developed by Pomerance in the early 1980's, for a long time was more successful than any other method in factoring integers n of general type which have no prime factor of order of magnitude significantly less than $\sqrt{n}$. (For integers n having a special form there may be special purpose methods which are faster, and for n divisible by a prime much smaller than $\sqrt{n}$ the elliptic curve factorization method in §VI.4 is faster. Also see the discussion of the number field sieve at the end of the section.)

The quadratic sieve is a variant of the factor base approach discussed in §3. As our factor base B we take the set of all primes $p \le P$ (where P is some bound to be chosen in some optimal way) such that n is a quadratic residue mod p, i.e., $\left(\frac{n}{p}\right) = 1$ for p odd, and $p = 2$ is always included in B. The set of integers S in which we look for B-numbers (recall that a B-number is an integer divisible only by primes in B) will be the same set that we used in Fermat factorization (see §3), namely:

$$S = \{t^2 - n \mid [\sqrt{n}] + 1 \le t \le [\sqrt{n}] + A\}$$

for some suitably chosen bound A.

The main idea of the method is that, instead of taking each $s \in S$ one by one and dividing it by the primes $p \in B$ to see if it is a B-number, we take each $p \in B$ one by one and examine divisibility by p (and powers of p) simultaneously for all of the $s \in S$. The word "sieve" refers to this idea. Here we should recall the "sieve of Eratosthenes," which one can use to make a list of all primes $p \le A$. For example, to list the primes ≤ 1000 one takes the list of all integers ≤ 1000 and then for each $p = 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31$ one discards all multiples of p greater than p — one "lets them fall through a sieve which has holes spaced a distance p apart" — after which the numbers that remain are the primes.

We shall give an outline of a procedure to carry out the method, and then give an example. The particular version described below is only one possible variant, and it is not necessarily the most efficient one. Moreover, our example of a number n to be factored (and also the numbers to be factored in the exercises at the end of the section) will be chosen in the range $\approx 10^6$, so as to avoid having to work with large matrices. However, such n are far too small to illustrate the time advantage of the sieve in finding a large set of B-numbers.

Thus, suppose we have an odd composite integer n.

1. Choose bounds P and A, both of order of magnitude roughly

$$e^{\sqrt{\log n \log\log n}}.$$

Generally, A should be larger than P, but not larger than a fairly small power of P, e.g., $P < A < P^2$.

This function $\exp(\sqrt{\log n \log\log n})$, which we encountered before in this chapter and which is traditionally denoted $L(n)$, has an order of magnitude intermediate between polynomial in $\log n$ and polynomial in n. If $n \approx 10^6$, then $L(n) \approx 400$. In the examples below, we shall choose $P = 50$, $A = 500$.

2. For $t = [\sqrt{n}] + 1, [\sqrt{n}] + 2, \ldots, [\sqrt{n}] + A$, make a column listing the integers $t^2 - n$.

3. For each odd prime $p \le P$, first check that $\left(\frac{n}{p}\right) = 1$ (see §II.2); if not, then throw that p out of the factor base.

4. Assuming that p is an odd prime such that n is a quadratic residue mod p (we'll treat the case $p = 2$ separately), solve the equation $t^2 \equiv n \pmod{p^\beta}$ for $\beta = 1, 2, \ldots$, using the method in Exercise 20 of §II.2. Take increasing values of β until you find that there is no solution t which is congruent modulo p^β to any integer in the range $[\sqrt{n}] + 1 \le t \le [\sqrt{n}] + A$. Let β be the largest integer such that there is some t in this range for which $t^2 \equiv n \pmod{p^\beta}$. Let t_1 and t_2 be two solutions of $t^2 \equiv n \pmod{p^\beta}$ with

$t_2 \equiv -t_1 \pmod{p^\beta}$ (t_1 and t_2 are not necessarily in the range from $[\sqrt{n}]+1$ to $[\sqrt{n}]+A$).

5. Still with the same value of p, run down the list of t^2-n from part 2. In a column under p put a 1 next to all values of t^2-n for which t differs from t_1 by a multiple of p, change the 1 to a 2 next to all values of t^2-n for which t differs from t_1 by a multiple of p^2, change the 2 to a 3 next to all values of t^2-n for which t differs from t_1 by a multiple of p^3, and so on until p^β. Then do the same with t_1 replaced by t_2. The largest integer that appears in this column will be β.

6. As you go through the procedure in 5), each time you put down a 1 or change a 1 to a 2, a 2 to a 3, etc., divide the corresponding t^2-n by p and keep a record of what's left.

7. In the column $p=2$, if $n \not\equiv 1 \bmod 8$, then simply put a 1 next to the t^2-n for t odd and divide the corresponding t^2-n by 2. If $n \equiv 1 \bmod 8$, then solve the equation $t^2 \equiv n \pmod{2^\beta}$ and proceed exactly as in the case of odd p (except that there will be 4 different solutions t_1, t_2, t_3, t_4 modulo 2^β if $\beta \geq 3$).

8. When you finish with all primes $\leq P$, throw out all of the t^2-n except for those which have become 1 after division by all the powers of $p \leq P$. You will have a table of the form in Example 9 in §3, in which the column labeled b_i will have the values of t, $[\sqrt{n}]+1 \leq t \leq [\sqrt{n}]+A$, for which t^2-n is a B-number, and the other columns will correspond to all values of $p \leq P$ for which n is a quadratic residue.

9. The rest of the procedure is exactly as in §3.

Example. Let us try to factor $n = 1042387$, taking the bounds $P = 50$ and $A = 500$. Here $[\sqrt{n}] = 1020$. Our factor base consists of the 8 primes $\{2, 3, 11, 17, 19, 23, 43, 47\}$ for which 1042387 is a quadratic residue. Since $n \not\equiv 1 \pmod 8$, the column corresponding to $p = 2$ alternates between 1 and 0, with a 1 beside all odd t, $1021 \leq t \leq 1520$.

We describe in detail how to form the column under $p = 3$. We want a solution $t_1 = t_{1,0} + t_{1,1} \cdot 3 + t_{1,2} \cdot 3^2 + \cdots + t_{1,\beta-1} \cdot 3^{\beta-1}$ to $t_1^2 \equiv 1042387 \pmod{3^\beta}$, where $t_{1,j} \in \{0,1,2\}$ (for the other solution t_2 we can take $t_2 = 3^\beta - t_1$). We can obviously take $t_{1,0} = 1$. (For each of our 8 primes the first step — solving $t_1^2 \equiv 1042387 \pmod p$ — can be done quickly by trial and error; if we were working with larger primes, we could use the procedure described at the end of §II.2.) Next, we work modulo 9: $(1+3t_{1,1})^2 \equiv 1042387 \equiv 7 \pmod 9$, i.e., $6t_{1,1} \equiv 6 \pmod 9$, i.e., $2t_{1,1} \equiv 2 \pmod 3$, so $t_{1,1} = 1$. Next, modulo 27: $(1+3+9t_{1,2})^2 \equiv 1042387 \equiv 25 \pmod{27}$, i.e., $16 + 18t_{1,2} \equiv 25 \pmod{27}$, i.e., $2t_{1,2} \equiv 1 \pmod 3$, so $t_{1,2} = 2$. Then modulo 81: $(1+3+18+27t_{1,3})^2 \equiv 1042387 \equiv 79 \pmod{81}$, which leads to $t_{1,3} = 0$. Continuing until 3^7, we find the solution (in the notation of §I.1 for numbers written to the base 3): $t_1 \equiv (210211)_3 \pmod{3^7}$, and $t_2 \equiv (2012012)_3 \pmod{3^7}$. However, there is no t between 1021 and 1520 which is $\equiv t_1$ or t_2 modulo 3^7. Thus, we have $\beta = 6$, and we can take $t_1 = (210211)_3 = 589 \equiv 1318 \pmod{3^6}$ and $t_2 = 3^6 - t_1 = 140 \equiv 1112 \pmod{3^5}$ (note that there is no number in the range from 1021 to 1520 which is $\equiv t_2 \pmod{3^6}$).

We now construct our "sieve" for the prime 3 as follows. Starting from 1318, we take jumps of 3 down until we reach 1021 and up until we reach 1519, each time putting a 1 in the column, dividing the corresponding t^2-n by 3, and recording the result of the division. (Actually, for t odd, the number we divide by 3 is half of t^2-n, since we already divided t^2-n by 2 when we formed the column of alternating 0's and 1's under 2.) Then we do the same with jumps of 9, each time changing the 1 to 2 in the column under 3, dividing the quotient of t^2-n by another 3, and recording the result. We go through the analogous procedure with jumps of 27, 81, 243, and 729 (there is no jump possible for 729 — we merely change the 5 to 6 next to 1318 and divide the quotient of 1318^2—1042387 by another 3). Finally, we go through the same steps with $t_2 = 1112$ instead of $t_1 = 1318$, this time stopping with jumps of 243.

After going through this procedure for the remaining 6 primes in our factor base, we have a 500×8 array of exponents, each row corresponding to a value of t between 1021 and 1520. Now we throw out all rows for which t^2-n has not been reduced to 1 by repeated division by powers of p as we formed our table, i.e., we take only the rows for which t^2-n is a B-number. In the present example $n = 1042387$ we are left with the following table (here blank spaces denote zero exponents):

t	t^2-n	2	3	11	17	19	23	43	47
1021	54	1	3	–	–	–	–	–	–
1027	12342	1	1	2	1	–	–	–	–
1030	18513	–	2	2	1	–	–	–	–
1061	83334	1	1	–	1	1	–	1	–
1112	194157	–	5	–	1	–	–	–	1
1129	232254	1	3	1	1	–	1	–	–
1148	275517	–	2	3	–	–	1	–	–
1175	338238	1	2	–	–	1	1	1	–
1217	438702	1	1	1	2	–	1	–	–
1390	889713	–	2	2	–	1	–	1	–
1520	1268013	–	1	–	1	–	2	–	1

Proceeding as we did in Example 9 in §3, we now look for relations modulo 2 between the rows of this matrix. That is, moving down from the first row, we look for a subset of the rows which sums to an even number in each column. The first such subset we find here is the first three rows, the sum of which is twice the row 1 3 2 1 – – – – . Thus, we obtain the congruence

$$(1021 \cdot 1027 \cdot 1030)^2 \equiv (2 \cdot 3^3 \cdot 11^2 \cdot 17)^2 \pmod{1042387}.$$

But despite our good fortune in finding a set of mod 2 linearly dependent rows so quickly, it turns out that we are not so lucky after all: the two numbers being squared in the above congruence are both $\equiv 111078 \ (mod\ 1042387)$, so we get only the trivial factorization. As we continue down the matrix, we find some other sets of dependent rows, which also fail to give us a nontrivial factorization. Finally, when we are about to give up — and start over again with a larger A — we notice that the last row — corresponding to our very last value of t — is dependent on the earlier rows. More precisely, it is equal modulo 2 to the fifth row. This gives us $(1112 \cdot 1520)^2 \equiv (3^3 \cdot 17 \cdot 23 \cdot 47)^2 \ (mod\ 1042387)$, i.e., $647853^2 \equiv 496179^2 \ (mod\ 1042387)$, and we obtain the nontrivial factor $g.c.d.(647853 - 496179, 1042387) = 1487$.

Based on some plausible conjectures, one can show that the expected running time of the quadratic sieve factoring method is asymptotically

$$O\left(e^{(1+\epsilon)\sqrt{\log n \log\log n}}\right)$$

for any $\epsilon > 0$. There is a fairly large space requirement, also of the form $\exp(C\sqrt{\log n \log\log n})$. For a detailed discussion of time and space requirements for the quadratic sieve (and several other) factoring algorithms, see Pomerance's article in the volume *Computation Methods in Number Theory*.

The number field sieve. Until recently, all of the contenders for the best general purpose factoring algorithm had running time of the form

$$\exp\big(O(\sqrt{\log n \log\log n})\big).$$

Some people even thought that this function of n might be a natural lower bound on the running time. However, during the last few years a new method — called the *number field sieve* — has been developed that has a heuristic running time that is much better (asymptotically), namely:

$$\exp\big(O((\log n)^{1/3}(\log\log n)^{2/3})\big).$$

In practice, it appears to be the fastest method for factoring numbers that are at or beyond the current (1994) upper limits of what can be factored, i.e., > 150 digits.

In some respects, the number field sieve factoring algorithm is similar to the earlier algorithms that attempt to combine congruences so as to obtain a relation of the form $x^2 \equiv y^2 \ (mod\ n)$. However, one uses a "factor base" in the ring of integers of a suitably chosen algebraic number field. Thus, along with the basic machinery of the quadratic sieve, this factoring method uses algebraic number theory. It is perhaps the most complicated factoring algorithm known. We shall give only an overview.

The basic requirements of the algorithm can be briefly described as follows. Given an integer n to be factored, choose a degree d and find n as

the value at some integer m of an irreducible monic integer polynomial of degree d:

$$n = f(m) = m^d + a_{d-1}m^{d-1} + a_{d-2}m^{d-2} + \cdots + a_1 m + a_0,$$

where m and the a_k are integers that are $O(n^{1/d})$. One way to find such a polynomial is to let m be the integer part of the d-th root of n and then expand n to the base m. For 125-digit numbers an analysis of the algorithm suggests that d should be 5, so that m and the coefficients will have about 25 digits.

The number field sieve then searches (by a sieving process similar to the quadratic sieve) for as many pairs (a, b) as possible such that both $a + bm$ and also

$$b^d f(-a/b) = (-a)^d + a_{d-1}(-a)^{d-1}b + a_{d-2}(-a)^{d-2}b^2 + \cdots - a_1 a b^{d-1} + a_0 b^d$$

are smooth over a given factor base (i.e., are divisible only by primes in the factor base). The details of how this is done and how this leads to a factorization of n can be found in the book *The Development of the Number Field Sieve* cited in the references below. In order for this procedure to succeed, the proportion of smooth numbers among values of the polynomial f should be approximately the same as the proportion of smooth numbers among all numbers of the same size. Although this is likely to be true, and is true in all examples that have been computed, it seems to be a very hard assertion to prove. Since the estimate of running time depends on this unproved conjecture, it is a heuristic estimate. While perhaps of little consequence in practice for factoring actual numbers, this circumstance points to some important open problems in the analysis of the theoretical asymptotic complexity of factoring.

The author would like to thank Joe Buhler for providing the above brief summary of the number field sieve for this book.

Exercises

1. In the example, find all linear dependence relations mod 2 between the rows of the matrix, and show that if $P = 50$ and $A \leq 499$ one cannot get a nontrivial factorization of 1042387 by this method.
2. Let $n \longrightarrow \infty$, and suppose that P and A are always chosen to have the same order of magnitude (for example, suppose that there are positive constants c_1 and c_2 such that $c_1 \leq \log A / \log P \leq c_2$). Asymptotically, what is the most time-consuming part of steps 1)-7) in the above version of the quadratic sieve? Give a big-O estimate for the number of bit operations required by that step.
3. Use the method in this section with $P = 50$ and $A = 500$ to factor: (a) 1046603, (b) 1059691, and (c) 998771.