20. [*M40*] Investigate the accuracy of Euclid's algorithm: What can be said about calculation of the greatest common divisor of polynomials whose coefficients are floating point numbers?

21. [*M25*] Prove that the computation time required by Algorithm C to compute the gcd of two nth degree polynomials over the integers is $O(n^4(\log Nn)^2)$, if the coefficients of the given polynomials are bounded by N in absolute value.

22. [*M23*] Prove Sturm's theorem. [*Hint:* Some sign sequences are impossible.]

23. [*M22*] Prove that if $u(x)$ in (29) has $\deg(u)$ real roots, then we have $\deg(u_{j+1}) = \deg(u_j) - 1$ for $0 \le j \le k$.

24. [*M21*] Show that (19) simplifies to (20) and (23) simplifies to (24).

25. [*M24*] (W. S. Brown.) Prove that all the polynomials $u_j(x)$ in (16) for $j \ge 3$ are multiples of $\gcd(\ell(u), \ell(v))$, and explain how to improve Algorithm C accordingly.

▸ **26.** [*M26*] The purpose of this exercise is to give an analog for polynomials of the fact that continued fractions with positive integer entries give the best approximations to real numbers (exercise 4.5.3-42).

Let $u(x)$ and $v(x)$ be polynomials over a field, with $\deg(u) > \deg(v)$, and let $a_1(x), a_2(x), \ldots$ be the quotient polynomials when Euclid's algorithm is applied to $u(x)$ and $v(x)$. For example, the sequence of quotients in (5) and (6) is $9x^2 + 7$, $5x^2 + 5$, $6x^3 + 5x^2 + 6x + 5$, $9x + 12$. We wish to show that the convergents $p_n(x)/q_n(x)$ of the continued fraction $/a_1(x), a_2(x), \ldots/$ are the "best approximations" of low degree to the rational function $v(x)/u(x)$, where we have $p_n(x) = Q_{n-1}(a_2(x), \ldots, a_n(x))$ and $q_n(x) = Q_n(a_1(x), \ldots, a_n(x))$ in terms of the continuant polynomials of Eq. 4.5.3–4. By convention, we let $p_0(x) = q_{-1}(x) = 0$, $p_{-1}(x) = q_0(x) = 1$.

Prove that if $p(x)$ and $q(x)$ are polynomials such that $\deg(q) < \deg(q_n)$ and $\deg(pu - qv) \le \deg(p_{n-1}u - q_{n-1}v)$, for some $n \ge 1$, then $p(x) = cp_{n-1}(x)$ and $q(x) = cq_{n-1}(x)$ for some constant c. In particular, each $q_n(x)$ is a "record-breaking" polynomial in the sense that no nonzero polynomial $q(x)$ of smaller degree can make the quantity $p(x)u(x) - q(x)v(x)$, for any polynomial $p(x)$, achieve a degree as small as $p_n(x)u(x) - q_n(x)v(x)$.

*4.6.2. Factorization of Polynomials

Let us now consider the problem of *factoring* polynomials, not merely finding the greatest common divisor of two or more of them.

Factoring modulo p. As in the case of integer numbers (Sections 4.5.2, 4.5.4), the problem of factoring seems to be more difficult than finding the greatest common divisor. But factorization of polynomials modulo a prime integer p is not as hard to do as we might expect. It is much easier to find the factors of an arbitrary polynomial of degree n, modulo 2, than to use any known method to find the factors of an arbitrary n-bit binary number. This surprising situation is a consequence of an instructive factorization algorithm discovered in 1967 by Elwyn R. Berlekamp [*Bell System Technical J.* **46** (1967), 1853–1859].

Let p be a prime number; all arithmetic on polynomials in the following discussion will be done modulo p. Suppose that someone has given us a polynomial $u(x)$, whose coefficients are chosen from the set $\{0, 1, \ldots, p-1\}$; we may assume

that $u(x)$ is monic. Our goal is to express $u(x)$ in the form

$$u(x) = p_1(x)^{e_1} \dots p_r(x)^{e_r}, \tag{1}$$

where $p_1(x), \dots, p_r(x)$ are distinct, monic, irreducible polynomials.

As a first step, we can use a standard technique to determine whether any of the exponents $e_1, \dots, e_r$ are greater than unity. If

$$u(x) = u_n x^n + \cdots + u_0 = v(x)^2 w(x), \tag{2}$$

then its "derivative" formed in the usual way (but modulo p) is

$$u'(x) = nu_n x^{n-1} + \cdots + u_1 = 2v(x)v'(x)w(x) + v(x)^2 w'(x), \tag{3}$$

and this is a multiple of the squared factor $v(x)$. Therefore our first step in factoring $u(x)$ is to form

$$\gcd\big(u(x), u'(x)\big) = d(x). \tag{4}$$

If $d(x)$ is equal to 1, we know that $u(x)$ is "squarefree," the product of distinct primes $p_1(x) \dots p_r(x)$. If $d(x)$ is not equal to 1 and $d(x) \neq u(x)$, then $d(x)$ is a proper factor of $u(x)$; the relation between the factors of $d(x)$ and the factors of $u(x)/d(x)$ speeds up the factorization process nicely in this case (see exercise 34). Finally, if $d(x) = u(x)$, we must have $u'(x) = 0$; hence the coefficient u_k of x^k is nonzero only when k is a multiple of p. This means that $u(x)$ can be written as a polynomial of the form $v(x^p)$, and in such a case we have

$$u(x) = v(x^p) = \big(v(x)\big)^p; \tag{5}$$

the factorization process can be completed by finding the irreducible factors of $v(x)$ and raising them to the pth power.

Identity (5) may appear somewhat strange to the reader; it is an important fact that is basic to Berlekamp's algorithm and to several other methods we shall discuss. We can prove it as follows: If $v_1(x)$ and $v_2(x)$ are any polynomials modulo p, then

$$\begin{aligned}\big(v_1(x) + v_2(x)\big)^p &= v_1(x)^p + \tbinom{p}{1} v_1(x)^{p-1} v_2(x) \\ &\qquad + \cdots + \tbinom{p}{p-1} v_1(x) v_2(x)^{p-1} + v_2(x)^p \\ &= v_1(x)^p + v_2(x)^p,\end{aligned}$$

since the binomial coefficients $\binom{p}{1}, \dots, \binom{p}{p-1}$ are all multiples of p. Furthermore if a is any integer, we have $a^p \equiv a$ (modulo p) by Fermat's theorem. Therefore when $v(x) = v_m x^m + v_{m-1} x^{m-1} + \cdots + v_0$, we find that

$$\begin{aligned}v(x)^p &= (v_m x^m)^p + (v_{m-1} x^{m-1})^p + \cdots + (v_0)^p \\ &= v_m x^{mp} + v_{m-1} x^{(m-1)p} + \cdots + v_0 \;=\; v(x^p).\end{aligned}$$

The above remarks show that the problem of factoring a polynomial reduces to the problem of factoring a squarefree polynomial. Let us therefore assume that

$$u(x) = p_1(x)p_2(x)\dots p_r(x) \tag{6}$$

is the product of distinct primes. How can we be clever enough to discover the $p_j(x)$'s when only $u(x)$ is given? Berlekamp's idea is to make use of the Chinese remainder theorem, which is valid for polynomials just as it is valid for integers (see exercise 3). If $(s_1, s_2, \dots, s_r)$ is any r-tuple of integers mod p, the Chinese remainder theorem implies that *there is a unique polynomial* $v(x)$ *such that*

$$\begin{gathered} v(x) \equiv s_1 \pmod{p_1(x)}, \quad \dots, \quad v(x) \equiv s_r \pmod{p_r(x)}, \\ \deg(v) < \deg(p_1) + \deg(p_2) + \cdots + \deg(p_r) = \deg(u). \end{gathered} \tag{7}$$

The notation $g(x) \equiv h(x) \pmod{f(x)}$ that appears here is the same as "$g(x) \equiv h(x) \pmod{f(x) \text{ and } p}$" in exercise 3.2.2–11, since we are considering polynomial arithmetic modulo p. The polynomial $v(x)$ in (7) gives us a way to get at the factors of $u(x)$, for if $r \geq 2$ and $s_1 \neq s_2$, we will have $\gcd\big(u(x), v(x) - s_1\big)$ divisible by $p_1(x)$ but not by $p_2(x)$.

Since this observation shows that we can get information about the factors of $u(x)$ from appropriate solutions $v(x)$ of (7), let us analyze (7) more closely. In the first place we can observe that the polynomial $v(x)$ satisfies the condition $v(x)^p \equiv s_j^p = s_j \equiv v(x) \pmod{p_j(x)}$ for $1 \leq j \leq r$, therefore

$$v(x)^p \equiv v(x) \pmod{u(x)}, \qquad \deg(v) < \deg(u). \tag{8}$$

In the second place we have the basic polynomial identity

$$x^p - x \equiv (x-0)(x-1)\dots\big(x-(p-1)\big) \pmod{p} \tag{9}$$

(see exercise 6); hence

$$v(x)^p - v(x) = \big(v(x)-0\big)\big(v(x)-1\big)\dots\big(v(x)-(p-1)\big) \tag{10}$$

is an identity for any polynomial $v(x)$, when we are working modulo p. If $v(x)$ satisfies (8), it follows that $u(x)$ divides the left-hand side of (10), so every irreducible factor of $u(x)$ must divide one of the p relatively prime factors of the right-hand side of (10). In other words, *all* solutions of (8) must have the form of (7), for some $s_1, s_2, \dots, s_r$; *there are exactly* p^r *solutions of* (8).

The solutions $v(x)$ to congruence (8) therefore provide a key to the factorization of $u(x)$. It may seem harder to find all solutions to (8) than to factor $u(x)$ in the first place, but in fact this is not true, since the set of solutions to (8) is closed under addition. Let $\deg(u) = n$; we can construct the $n \times n$ matrix

$$Q = \begin{pmatrix} q_{0,0} & q_{0,1} & \cdots & q_{0,n-1} \\ \vdots & \vdots & & \vdots \\ q_{n-1,0} & q_{n-1,1} & \cdots & q_{n-1,n-1} \end{pmatrix} \tag{11}$$

where

$$x^{pk} \equiv q_{k,n-1}x^{n-1} + \cdots + q_{k,1}x + q_{k,0} \pmod{u(x)}. \tag{12}$$

Then $v(x) = v_{n-1}x^{n-1} + \cdots + v_1x + v_0$ is a solution to (8) if and only if

$$(v_0, v_1, \ldots, v_{n-1})Q = (v_0, v_1, \ldots, v_{n-1}); \tag{13}$$

for the latter equation holds if and only if

$$v(x) = \sum_j v_j x^j = \sum_j \sum_k v_k q_{k,j} x^j \equiv \sum_k v_k x^{pk} = v(x^p) \equiv v(x)^p \pmod{u(x)}.$$

Berlekamp's factoring algorithm therefore proceeds as follows:

B1. Ensure that $u(x)$ is squarefree; i.e., if $\gcd(u(x), u'(x)) \neq 1$, reduce the problem of factoring $u(x)$, as stated earlier in this section.

B2. Form the matrix Q defined by (11) and (12). This can be done in one of two ways, depending on whether or not p is very large, as explained below.

B3. "Triangularize" the matrix $Q - I$, where $I = (\delta_{ij})$ is the $n \times n$ identity matrix, finding its rank $n - r$ and finding linearly independent vectors $v^{[1]}$, $\ldots$, $v^{[r]}$ such that $v^{[j]}(Q - I) = (0, 0, \ldots, 0)$ for $1 \leq j \leq r$. (The first vector $v^{[1]}$ may always be taken as $(1, 0, \ldots, 0)$, representing the trivial solution $v^{[1]}(x) = 1$ to (8). The "triangularization" needed in this step can be done using appropriate column operations, as explained in Algorithm N below.) *At this point, r is the number of irreducible factors of $u(x)$,* because the solutions to (8) are the p^r polynomials corresponding to the vectors $t_1v^{[1]} + \cdots + t_rv^{[r]}$ for all choices of integers $0 \leq t_1, \ldots, t_r < p$. Therefore if $r = 1$ we know that $u(x)$ is irreducible, and the procedure terminates.

B4. Calculate $\gcd(u(x), v^{[2]}(x) - s)$ for $0 \leq s < p$, where $v^{[2]}(x)$ is the polynomial represented by vector $v^{[2]}$. The result will be a nontrivial factorization of $u(x)$, because $v^{[2]}(x) - s$ is nonzero and has degree less than $\deg(u)$, and by exercise 7 we have

$$u(x) = \prod_{0 \leq s < p} \gcd(v(x) - s, u(x)) \tag{14}$$

whenever $v(x)$ satisfies (8).

If the use of $v^{[2]}(x)$ does not succeed in splitting $u(x)$ into r factors, further factors can be obtained by calculating $\gcd(v^{[k]}(x) - s, w(x))$ for $0 \leq s < p$ and all factors $w(x)$ found so far, for $k = 3, 4, \ldots$, until r factors are obtained. (If we choose $s_i \neq s_j$ in (7), we obtain a solution $v(x)$ to (8) that distinguishes $p_i(x)$ from $p_j(x)$; some $v^{[k]}(x) - s$ will be divisible by $p_i(x)$ and not by $p_j(x)$, so this procedure will eventually find all of the factors.)

If p is 2 or 3, the calculations of this step are quite efficient; but if p is more than 25, say, there is a much better way to proceed, as we shall see later. ∎

As an example of this procedure, let us now determine the factorization of

$$u(x) = x^8 + x^6 + 10x^4 + 10x^3 + 8x^2 + 2x + 8 \tag{15}$$

modulo 13. (This polynomial appears in several of the examples in Section 4.6.1.) A quick calculation using Algorithm 4.6.1E shows that $\gcd(u(x), u'(x)) = 1$; therefore $u(x)$ is squarefree, and we turn to step B2. Step B2 involves calculating the Q matrix, which in this case is an 8×8 array. The first row of Q is always $(1, 0, 0, \ldots, 0)$, representing the polynomial $x^0 \bmod u(x) = 1$. The second row represents $x^{13} \bmod u(x)$, and, in general, $x^k \bmod u(x)$ may readily be determined as follows (for relatively small values of k): If

$$u(x) = x^n + u_{n-1}x^{n-1} + \cdots + u_1 x + u_0$$

and if

$$x^k \equiv a_{k,n-1}x^{n-1} + \cdots + a_{k,1}x + a_{k,0} \ \text{(modulo } u(x)\text{)},$$

then

$$\begin{aligned} x^{k+1} &\equiv a_{k,n-1}x^n + \cdots + a_{k,1}x^2 + a_{k,0}x \\ &\equiv a_{k,n-1}(-u_{n-1}x^{n-1} - \cdots - u_1 x - u_0) + a_{k,n-2}x^{n-1} + \cdots + a_{k,0}x \\ &= a_{k+1,n-1}x^{n-1} + \cdots + a_{k+1,1}x + a_{k+1,0}, \end{aligned}$$

where

$$a_{k+1,j} = a_{k,j-1} - a_{k,n-1}u_j. \tag{16}$$

In this formula $a_{k,-1}$ is treated as zero, so that $a_{k+1,0} = -a_{k,n-1}u_0$. The simple "shift register" recurrence (16) makes it easy to calculate $x^1, x^2, x^3, \ldots \bmod u(x)$. Inside a computer, this calculation is of course generally done by maintaining a one-dimensional array $(a_{n-1}, \ldots, a_1, a_0)$ and repeatedly setting $t \leftarrow a_{n-1}$, $a_{n-1} \leftarrow (a_{n-2} - tu_{n-1}) \bmod p$, $\ldots$, $a_1 \leftarrow (a_0 - tu_1) \bmod p$, and $a_0 \leftarrow (-tu_0) \bmod p$. (We have seen similar procedures in connection with random number generation; cf. Eq. 3.2.2–10.) For the example polynomial $u(x)$ in (15), we obtain the following sequence of coefficients of $x^k \bmod u(x)$, using arithmetic modulo 13:

k	$a_{k,7}$	$a_{k,6}$	$a_{k,5}$	$a_{k,4}$	$a_{k,3}$	$a_{k,2}$	$a_{k,1}$	$a_{k,0}$
0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0
2	0	0	0	0	0	1	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	1	0	0	0	0	0
6	0	1	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0
8	0	12	0	3	3	5	11	5
9	12	0	3	3	5	11	5	0
10	0	4	3	2	8	0	2	8
11	4	3	2	8	0	2	8	0
12	3	11	8	12	1	2	5	7
13	11	5	12	10	11	7	1	2

Therefore the second row of Q is $(2, 1, 7, 11, 10, 12, 5, 11)$. Similarly we may determine $x^{26} \bmod u(x)$, ..., $x^{91} \bmod u(x)$, and we find that

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 7 & 11 & 10 & 12 & 5 & 11 \\ 3 & 6 & 4 & 3 & 0 & 4 & 7 & 2 \\ 4 & 3 & 6 & 5 & 1 & 6 & 2 & 3 \\ 2 & 11 & 8 & 8 & 3 & 1 & 3 & 11 \\ 6 & 11 & 8 & 6 & 2 & 7 & 10 & 9 \\ 5 & 11 & 7 & 10 & 0 & 11 & 7 & 12 \\ 3 & 3 & 12 & 5 & 0 & 11 & 9 & 12 \end{pmatrix},$$

$$Q - I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 7 & 11 & 10 & 12 & 5 & 11 \\ 3 & 6 & 3 & 3 & 0 & 4 & 7 & 2 \\ 4 & 3 & 6 & 4 & 1 & 6 & 2 & 3 \\ 2 & 11 & 8 & 8 & 2 & 1 & 3 & 11 \\ 6 & 11 & 8 & 6 & 2 & 6 & 10 & 9 \\ 5 & 11 & 7 & 10 & 0 & 11 & 6 & 12 \\ 3 & 3 & 12 & 5 & 0 & 11 & 9 & 11 \end{pmatrix}. \tag{17}$$

That finishes step B2; the next step of Berlekamp's procedure requires finding the "null space" of $Q - I$. In general, suppose that A is an $n \times n$ matrix over a field, whose rank $n - r$ is to be determined; suppose further that we wish to determine linearly independent vectors $v^{[1]}$, $v^{[2]}$, ..., $v^{[r]}$ such that $v^{[1]}A = v^{[2]}A = \cdots = v^{[r]}A = (0, \ldots, 0)$. An algorithm for this calculation can be based on the observation that any column of A may be multiplied by a nonzero quantity, and any multiple of one of its columns may be added to a different column, without changing the rank or the vectors $v^{[1]}, \ldots, v^{[r]}$. (These transformations amount to replacing A by AB, where B is a nonsingular matrix.) The following well-known "triangularization" procedure may therefore be used.

Algorithm N (*Null space algorithm*). Let A be an $n \times n$ matrix, whose elements a_{ij} belong to a field and have subscripts in the range $0 \le i, j < n$. This algorithm outputs r vectors $v^{[1]}$, ..., $v^{[r]}$, which are linearly independent over the field and satisfy $v^{[j]}A = (0, \ldots, 0)$, where $n - r$ is the rank of A.

N1. [Initialize.] Set $c_0 \leftarrow c_1 \leftarrow \cdots \leftarrow c_{n-1} \leftarrow -1$, $r \leftarrow 0$. (During the calculation we will have $c_j \ge 0$ only if $a_{c_j j} = -1$ and all other entries of row c_j are zero.)

N2. [Loop on k.] Do step N3 for $k = 0, 1, \ldots, n - 1$, and then terminate the algorithm.

N3. [Scan row for dependence.] If there is some j in the range $0 \le j < n$ such that $a_{kj} \ne 0$ and $c_j < 0$, then do the following: Multiply column j of A by $-1/a_{kj}$ (so that a_{kj} becomes equal to -1); then add a_{ki} times column j to column i for all $i \ne j$; finally set $c_j \leftarrow k$. (Since it is not difficult to show that $a_{sj} = 0$ for all $s < k$, these operations have no effect on rows 0, 1, ..., $k - 1$ of A.)

On the other hand, if there is no j in the range $0 \le j < n$ such that $a_{kj} \neq 0$ and $c_j < 0$, then set $r \leftarrow r + 1$ and output the vector

$$v^{[r]} = (v_0, v_1, \ldots, v_{n-1})$$

defined by the rule

$$v_j = \begin{cases} a_{ks}, & \text{if } c_s = j \ge 0; \\ 1, & \text{if } j = k; \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

∎

An example will reveal the mechanism of this algorithm. Let A be the matrix $Q - I$ of (17) over the field of integers modulo 13. When $k = 0$, we output the vector $v^{[1]} = (1, 0, 0, 0, 0, 0, 0, 0)$. When $k = 1$, we may take j in step N3 to be either 0, 2, 3, 4, 5, 6, or 7; the choice here is completely arbitrary, although it affects the particular vectors that are chosen to be output by the algorithm. For hand calculation, it is most convenient to pick $j = 5$, since $a_{15} = 12 = -1$ already; the column operations of step N3 then change A to the matrix

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \textcircled{12} & 0 & 0 \\ 11 & 6 & 5 & 8 & 1 & 4 & 1 & 7 \\ 3 & 3 & 9 & 5 & 9 & 6 & 6 & 4 \\ 4 & 11 & 2 & 6 & 12 & 1 & 8 & 9 \\ 5 & 11 & 11 & 7 & 10 & 6 & 1 & 10 \\ 1 & 11 & 6 & 1 & 6 & 11 & 9 & 3 \\ 12 & 3 & 11 & 9 & 6 & 11 & 12 & 2 \end{pmatrix}.$$

(The circled element in column "5", row "1", is used here to indicate that $c_5 = 1$. Remember that Algorithm N numbers the rows and columns of the matrix starting with 0, not 1.) When $k = 2$, we may choose $j = 4$ and proceed in a similar way, obtaining the following matrices, which all have the same null space as $Q - I$:

$k = 2$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \textcircled{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & \textcircled{12} & 0 & 0 & 0 \\ 8 & 1 & 3 & 11 & 4 & 9 & 10 & 6 \\ 2 & 4 & 7 & 1 & 1 & 5 & 9 & 3 \\ 12 & 3 & 0 & 5 & 3 & 5 & 4 & 5 \\ 0 & 1 & 2 & 5 & 7 & 0 & 3 & 0 \\ 11 & 6 & 7 & 0 & 7 & 0 & 6 & 12 \end{pmatrix}$$

$k = 3$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \textcircled{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & \textcircled{12} & 0 & 0 & 0 \\ 0 & \textcircled{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 9 & 8 & 9 & 11 & 8 & 8 & 5 \\ 1 & 10 & 4 & 11 & 4 & 4 & 0 & 0 \\ 5 & 12 & 12 & 7 & 3 & 4 & 6 & 7 \\ 2 & 7 & 2 & 12 & 9 & 11 & 11 & 2 \end{pmatrix}$$

$0 \le j < n$ such that
he vector

(18)

∎

ι. Let A be the matrix
$k = 0$, we output the
ke j in step N3 to be
ırbitrary, although it
ɔy the algorithm. For
nce $a_{15} = 12 = -1$
to the matrix

ıere to indicate that
and columns of the
se $j = 4$ and proceed
ll have the same null

$= 3$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \textcircled{12} & 0 & 0 \\ \textcircled{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 11 & 8 & 8 & 5 \\ 4 & 4 & 0 & 0 \\ 3 & 4 & 6 & 7 \\ 9 & 11 & 11 & 2 \end{pmatrix}$$

$k = 4$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \textcircled{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & \textcircled{12} & 0 & 0 & 0 \\ 0 & \textcircled{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{12} \\ 1 & 10 & 4 & 11 & 4 & 4 & 0 & 0 \\ 8 & 2 & 6 & 10 & 11 & 11 & 0 & 9 \\ 1 & 6 & 4 & 11 & 2 & 0 & 0 & 10 \end{pmatrix}$$

$k = 5$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \textcircled{12} & 0 & 0 \\ 0 & 0 & 0 & 0 & \textcircled{12} & 0 & 0 & 0 \\ 0 & \textcircled{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \textcircled{12} \\ \textcircled{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 5 & 5 & 0 & 9 \\ 12 & 9 & 0 & 0 & 11 & 9 & 0 & 10 \end{pmatrix}$$

Now every column that has no circled entry is completely zero; so when $k = 6$ and $k = 7$ the algorithm outputs two more vectors, namely

$$v^{[2]} = (0, 5, 5, 0, 9, 5, 1, 0), \qquad v^{[3]} = (0, 9, 11, 9, 10, 12, 0, 1).$$

From the form of matrix A after $k = 5$, it is evident that these vectors satisfy the equation $vA = (0, \ldots, 0)$. Since the computation has produced three linearly independent vectors, $u(x)$ must have exactly three irreducible factors.

Finally we can go to step B4 of the factoring procedure. The calculation of $\gcd\big(u(x), v^{[2]}(x) - s\big)$ for $0 \le s < 13$, where $v^{[2]}(x) = x^6 + 5x^5 + 9x^4 + 5x^2 + 5x$, gives $x^5 + 5x^4 + 9x^3 + 5x + 5$ as the answer when $s = 0$, and $x^3 + 8x^2 + 4x + 12$ when $s = 2$; the gcd is unity for other values of s. Therefore $v^{[2]}(x)$ gives us only two of the three factors. Turning to $\gcd\big(v^{[3]}(x) - s,\ x^5 + 5x^4 + 9x^3 + 5x + 5\big)$, where $v^{[3]}(x) = x^7 + 12x^5 + 10x^4 + 9x^3 + 11x^2 + 9x$, we obtain the value $x^4 + 2x^3 + 3x^2 + 4x + 6$ when $s = 6$, $x + 3$ when $s = 8$, and unity otherwise. Thus the complete factorization is

$$u(x) = (x^4 + 2x^3 + 3x^2 + 4x + 6)(x^3 + 8x^2 + 4x + 12)(x + 3). \tag{19}$$

Let us now estimate the running time of Berlekamp's method when an nth degree polynomial is factored modulo p. First assume that p is relatively small, so that the four arithmetic operations can be done modulo p in essentially a fixed length of time. (Division modulo p can be converted to multiplication, by storing a table of reciprocals as suggested in exercise 9; for example, when working modulo 13, we have $\frac{1}{2} = 7$, $\frac{1}{3} = 9$, etc.) The computation in step B1 takes $O(n^2)$ units of time; step B2 takes $O(pn^2)$. For step B3 we use Algorithm N, which requires $O(n^3)$ units of time at most. Finally, in step B4 we can observe that the calculation of $\gcd\big(f(x), g(x)\big)$ by Euclid's algorithm takes $O\big(\deg(f)\deg(g)\big)$ units of time; hence the calculation of $\gcd\big(v^{[j]}(x) - s,\ w(x)\big)$ for fixed j and s and for all factors $w(x)$ of $u(x)$ found so far takes $O(n^2)$ units. Step B4 therefore requires $O(prn^2)$ units of time at most. *Berlekamp's procedure factors an arbitrary polynomial of degree n, modulo p, in $O(n^3 + prn^2)$ steps,* when p is a small prime; and exercise 5 shows that the average number of factors, r, is approximately $\ln n$.

Thus the algorithm is much faster than any known methods of factoring n-digit numbers in the p-ary number system.

Of course, when n and p are small, a trial-and-error factorization procedure analogous to Algorithm 4.5.4A will be even faster than Berlekamp's method. Exercise 1 implies that it is a good idea to cast out factors of small degree first when p is small, before going to any more complicated procedure, even when n is large.

When p is large, a different implementation of Berlekamp's procedure would be used for the calculations. Division modulo p would not be done with an auxiliary table of reciprocals; instead the method of exercise 4.5.2–15, which takes $O((\log p)^2)$ steps, would probably be used. Then step B1 would take $O(n^2(\log p)^2)$ units of time; similarly, step B3 takes $O(n^3(\log p)^2)$. In step B2, we can form $x^p \bmod u(x)$ in a more efficient way than (16) when p is large: Section 4.6.3 shows that this value can essentially be obtained by using $O(\log p)$ operations of "squaring $\bmod u(x)$," i.e., going from $x^k \bmod u(x)$ to $x^{2k} \bmod u(x)$. The squaring operation is relatively easy to perform if we first make an auxiliary table of $x^m \bmod u(x)$ for $m = n, n+1, \ldots, 2n-2$; if

$$x^k \bmod u(x) = c_{n-1}x^{n-1} + \cdots + c_1x + c_0,$$

then

$$x^{2k} \bmod u(x) = (c_{n-1}^2 x^{2n-2} + \cdots + (c_1c_0 + c_1c_0)x + c_0^2) \bmod u(x),$$

where $x^{2n-2}, \ldots, x^n$ can be replaced by polynomials in the auxiliary table. The total time to compute $x^p \bmod u(x)$ comes to $O(n^2(\log p)^3)$ units, and we obtain the second row of Q. To get further rows of Q, we can compute $x^{2p} \bmod u(x)$, $x^{3p} \bmod u(x)$, ..., simply by multiplying repeatedly by $x^p \bmod u(x)$, in a fashion analogous to squaring $\bmod u(x)$; step B2 is completed in $O(n^2(\log p)^3)$ units of time. Thus steps B1, B2, and B3 take a total of $O(n^2(\log p)^3 + n^3(\log p)^2)$ units of time; these three steps tell us the number of factors of $u(x)$.

But when p is large and we get to step B4, we are asked to calculate a greatest common divisor for p different values of s, and that is out of the question if p is even moderately large. This hurdle was first surmounted by Hans Zassenhaus [*J. Number Theory* **1** (1969), 291–311], who showed how to determine all of the "useful" values of s (see exercise 14); but an even better way to proceed was found by Zassenhaus and Cantor in 1980. If $v(x)$ is any solution to (8), we know that $u(x)$ divides $v(x)^p - v(x) = v(x) \cdot (v(x)^{(p-1)/2} + 1) \cdot (v(x)^{(p-1)/2} - 1)$. This suggests that we calculate

$$\gcd(u(x), v(x)^{(p-1)/2} - 1); \tag{20}$$

with a little bit of luck, (20) will be a nontrivial factor of $u(x)$. In fact, we can determine exactly how much luck is involved, by considering (7). Let $v(x) \equiv s_j$ (modulo $p_j(x)$) for $1 \le j \le r$; then $p_j(x)$ divides $v(x)^{(p-1)/2} - 1$ if and only if

$s_j^{(p-1)/2} \equiv 1 \pmod p$. We know that exactly $(p-1)/2$ of the integers s in the range $0 \le s < p$ satisfy $s^{(p-1)/2} \equiv 1 \pmod p$, hence about half of the $p_j(x)$ will appear in the gcd (20). More precisely, if $v(x)$ is a random solution of (8), where all p^r solutions are equally likely, the probability that the gcd (20) equals $u(x)$ is exactly

$$\big((p-1)/2p\big)^r,$$

and the probability that it equals 1 is $\big((p+1)/2p\big)^r$. The probability that a nontrivial factor will be obtained is therefore

$$1 - \left(\frac{p-1}{2p}\right)^r - \left(\frac{p+1}{2p}\right)^r = 1 - \frac{1}{2^{r-1}}\left(1 + \binom{r}{2}p^{-2} + \binom{r}{4}p^{-4} + \cdots\right) \ge \frac{4}{9},$$

for all $r \ge 2$ and $p \ge 3$.

It is therefore a good idea to replace step B4 by the following procedure, unless p is quite small: Set $v(x) \leftarrow a_1 v^{[1]}(x) + a_2 v^{[2]}(x) + \cdots + a_r v^{[r]}(x)$, where the coefficients a_j are randomly chosen in the range $0 \le a_j < p$. Let the current partial factorization of $u(x)$ be $u_1(x) \ldots u_t(x)$ where t is initially 1. Compute

$$g_i(x) = \gcd\big(u_i(x), v(x)^{(p-1)/2} - 1\big)$$

for all i such that $\deg(u_i) > 1$; replace $u_i(x)$ by $g_i(x) \cdot \big(u_i(x)/g_i(x)\big)$ and increase the value of t, whenever a nontrivial gcd is found. Repeat this process for different choices of $v(x)$ until $t = r$.

If we assume (as we may) that only $O(\log r)$ random solutions $v(x)$ to (8) will be needed, we can give an upper bound on the time required to perform this alternative to step B4. It takes $O\big(r(\log p)^2\big)$ steps to compute $v(x)$; and if $\deg(u_i) = d$, it takes $O\big(d^2(\log p)^3\big)$ steps to compute $v(x)^{(p-1)/2} \bmod u_i(x)$ and $O\big(d^2(\log p)^2\big)$ further steps to compute $\gcd\big(u_i(x), v(x)^{(p-1)/2} - 1\big)$. Thus the total time is $O\big(n^2(\log p)^3 \log r\big)$.

For further discussion, see the articles by E. R. Berlekamp, *Math. Comp.* **24** (1970), 713–735, and Robert T. Moenck, *Math. Comp.* **31** (1977), 235–250.

Distinct-degree factorization. We shall now turn to a somewhat simpler way to find factors modulo p. The ideas we have studied so far in this section involve many instructive insights into computational algebra, so the author does not apologize to the reader for presenting them; but it turns out that the problem of factorization modulo p can actually be solved without relying on so many concepts.

In the first place we can make use of the fact that an irreducible polynomial $q(x)$ of degree d is a divisor of $x^{p^d} - x$, and it is not a divisor of $x^{p^c} - x$ for $c < d$; see exercise 16. We can therefore cast out the irreducible factors of each degree separately, by adopting the following strategy.

D1. Rule out squared factors, as in Berlekamp's method. Also set $v(x) \leftarrow u(x)$, $w(x) \leftarrow$ "x", and $d \leftarrow 0$. (Here $v(x)$ and $w(x)$ are variables that have polynomials as values.)

D2. (At this point $w(x) = x^{p^d} \bmod v(x)$; all of the irreducible factors of $v(x)$ are distinct and have degree $> d$.) If $d+1 > \frac{1}{2}\deg(v)$, the procedure terminates since we either have $v(x) = 1$ or $v(x)$ is irreducible. Otherwise increase d by 1 and replace $w(x)$ by $w(x)^p \bmod v(x)$.

D3. Find $g_d(x) = \gcd\big(w(x) - x, v(x)\big)$. (This is the product of all the irreducible factors of $u(x)$ whose degree is d.) If $g_d(x) \neq 1$, replace $v(x)$ by $v(x)/g_d(x)$ and $w(x)$ by $w(x) \bmod v(x)$; and if the degree of $g_d(x)$ is greater than d, use the algorithm below to find its factors. Return to step D2. ∎

This procedure determines the product of all irreducible factors of each degree d, and therefore it tells us how many factors there are of each degree. Since the three factors of our example polynomial (19) have different degrees, they would all be discovered without any need to factorize the polynomials $g_d(x)$.

The distinct degree factorization technique was known to several people in 1960 [cf. S. W. Golomb, L. R. Welch, A. Hales, "On the factorization of trinomials over GF(2)," Jet Propulsion Laboratory memo 20–189 (July 14, 1959)], but there seem to be no references to it in the "open literature." Previous work by Š. Schwarz, *Quart. J. Math.*, Oxford (2) **7** (1956), 110–124, had shown how to determine the number of irreducible factors of each degree, but not their product, using the matrix Q.

To complete the method, we need a way to split the polynomial $g_d(x)$ into its irreducible factors when $\deg(g_d) > d$. Michael Rabin pointed out in 1976 that this can be done by doing arithmetic in the field of p^d elements. David G. Cantor and Hans Zassenhaus discovered in 1979 that there is an even simpler way to proceed, based on the following identity: If p is any odd prime, we have

$$\begin{aligned} g_d(x) = \gcd\big(g_d(x), t(x)\big) \cdot \gcd\big(g_d(x), t(x)^{(p^d-1)/2} + 1\big) \\ \cdot \gcd\big(g_d(x), t(x)^{(p^d-1)/2} - 1\big) \end{aligned} \tag{21}$$

for all polynomials $t(x)$, since $t(x)^{p^d} - t(x)$ is a multiple of all irreducible polynomials of degree d. (We may regard $t(x)$ as an element of the field of size p^d, when that field consists of all polynomials modulo an irreducible $f(x)$ as in exercise 16.) Now exercise 29 shows that $\gcd\big(g_d(x), t(x)^{(p^d-1)/2}\big)$ will be a nontrivial factor of $g_d(x)$ about 50 per cent of the time, when $t(x)$ is a random polynomial of degree $\leq 2d - 1$; hence it will not take long to discover all of the factors. We may assume without loss of generality that $t(x)$ is monic, since integer multiples of $t(x)$ make no difference except possibly to change $t(x)^{(p^d-1)/2}$ into its negative. Thus in the case $d = 1$, we can take $t(x) = x + s$, where s is chosen at random. [See *SIAM J. Computing* **9** (1980), 273–280; *Math. Comp.*, to appear.]

Sometimes this procedure will in fact succeed for $d > 1$ when only linear polynomials $t(x)$ are used. For example, there are eight irreducible polynomials $f(x)$ of degree 3, modulo 3, and they will all be distinguished by calculating

$\gcd(f(x), (x+s)^{13} - 1)$ for $0 \le s < 3$:

$f(x)$	$s=0$	$s=1$	$s=2$
$x^3 + 2x + 1$	1	1	1
$x^3 + 2x + 2$	$f(x)$	$f(x)$	$f(x)$
$x^3 + x^2 + 2$	$f(x)$	$f(x)$	1
$x^3 + x^2 + x + 2$	$f(x)$	1	$f(x)$
$x^3 + x^2 + 2x + 1$	1	$f(x)$	$f(x)$
$x^3 + 2x^2 + 1$	1	$f(x)$	1
$x^3 + 2x^2 + x + 1$	1	1	$f(x)$
$x^3 + 2x^2 + 2x + 2$	$f(x)$	1	1

Exercise 31 contains a partial explanation of why linear polynomials can be effective; however, when the number of irreducible polynomials of degree d exceeds 2^p, it is clear that there will exist irreducibles that cannot be distinguished by linear choices of $t(x)$.

An alternative to (21) that works when $p = 2$ is discussed in exercise 30.

Factoring over the integers. It is somewhat more difficult to find the complete factorization of polynomials with integer coefficients when we are *not* working modulo p, but some reasonably efficient methods are available for this purpose.

Isaac Newton gave a method for finding linear and quadratic factors of polynomials with integer coefficients in his *Arithmetica Universalis* (1707). This method was extended by an astronomer named Friedrich von Schubert in 1793, who showed how to find all factors of degree n in a finite number of steps; see M. Cantor, *Geschichte der Mathematik* **4** (Leipzig: Teubner, 1908), 136–137. L. Kronecker rediscovered von Schubert's method independently about 90 years later; but unfortunately the method is very inefficient when n is five or more. Much better results can be obtained with the help of the "mod p" factorization methods presented above.

Suppose that we want to find the irreducible factors of a given polynomial

$$u(x) = u_n x^n + u_{n-1} x^{n-1} + \cdots + u_0, \qquad u_n \neq 0,$$

over the integers. As a first step, we can divide by the greatest common divisor of the coefficients; this leaves us with a *primitive* polynomial. We may also assume that $u(x)$ is squarefree, by dividing out $\gcd(u(x), u'(x))$ as in exercise 34.

Now if $u(x) = v(x)w(x)$, where each of these polynomials has integer coefficients, we obviously have $u(x) \equiv v(x)w(x)$ (modulo p) for all primes p, so there is a nontrivial factorization modulo p unless p divides $\ell(u)$. An efficient algorithm for factoring $u(x)$ modulo p can therefore be used in an attempt to reconstruct possible factorizations of $u(x)$ over the integers.

For example, let

$$u(x) = x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5. \tag{22}$$

We have seen above in (19) that

$$u(x) \equiv (x^4+2x^3+3x^2+4x+6)(x^3+8x^2+4x+12)(x+3) \text{ (modulo 13);} \quad (23)$$

and the complete factorization of $u(x)$ modulo 2 shows one factor of degree 6 and another of degree 2 (see exercise 10). From (23) we can see that $u(x)$ has no factor of degree 2, so it must be irreducible over the integers.

This particular example was perhaps too simple; experience shows that most irreducible polynomials can be recognized as such by examining their factors modulo a few primes, but it is *not* always so easy to establish irreducibility. For example, there are polynomials that can be properly factored modulo p for all primes p, with consistent degrees of the factors, yet they are irreducible over the integers (see exercise 12).

Almost all polynomials are irreducible over the integers, as shown in exercise 27. But we usually aren't trying to factor a random polynomial; there is probably some reason to expect a nontrivial factor or else the calculation would not have been attempted in the first place. We need a method that identifies factors when they are there.

In general if we try to find the factors of $u(x)$ by considering its behavior modulo different primes, the results will not be easy to combine; for example, if $u(x)$ actually is the product of four quadratic polynomials, it will be hard to match up their images with respect to different prime moduli. Therefore it is desirable to stick to a single prime and to see how much mileage we can get out of it, once we feel that the factors modulo this prime have the right degrees.

One idea is to work modulo a very *large* prime p, big enough so that the coefficients in any true factorization $u(x) = v(x)w(x)$ over the integers must actually lie between $-p/2$ and $p/2$. Then all possible integer factors can be "read off" from the mod p factors we know how to compute.

Exercise 20 shows how to obtain fairly good bounds on the coefficients of polynomial factors. For example, if (22) were reducible it would have a factor $v(x)$ of degree ≤ 4, and the coefficients of v would be at most 34 in magnitude by the results of that exercise. So all potential factors of $u(x)$ will be fairly evident if we work modulo any prime $p > 68$. Indeed, the complete factorization modulo 71 is

$$(x+12)(x+25)(x^2-13-7)(x^4-24x^3-16x^2+31x-12),$$

and we see immediately that none of these polynomials are factors of (22) over the integers since their constant terms do not divide 5; furthermore there is no way to obtain a divisor of (22) by grouping two of these factors, since none of the conceivable constant terms 12×25, -12×7, $12 \times (-12)$ is congruent to ± 1 or ± 5 (modulo 71).

Incidentally, it is not trivial to obtain good bounds on the coefficients of polynomial factors, since a lot of cancellation can occur when polynomials are multiplied. For example, the innocuous-looking polynomial $x^n - 1$ has irreducible

factors whose coefficients exceed $\exp(n^{1/\lg\lg n})$ for infinitely many n. [See R. C. Vaughan, *Michigan Math. J.* **21** (1974), 289–295.] The factorization of $x^n - 1$ is discussed in exercise 32.

Instead of using a large prime p, which might have to be truly enormous if $u(x)$ has large degree or large coefficients, we can also make use of small p, provided that $u(x)$ is squarefree mod p. For in this case, an important construction introduced by K. Hensel [*Theorie der Algebraischen Zahlen* (Leipzig: Teubner, 1908), Chapter 4] can be used to extend a factorization modulo p in a unique way to a factorization modulo p^e for arbitrarily high e. Hensel's method is described in exercise 22; if we apply it to (23) with $p = 13$ and $e = 2$, we obtain the unique factorization

$$u(x) \equiv (x - 36)(x^3 - 18x^2 + 82x - 66)(x^4 + 54x^3 - 10x^2 + 69x + 84)$$

(modulo 169). Calling these factors $v_1(x)v_3(x)v_4(x)$, we see that $v_1(x)$ and $v_3(x)$ are not factors of $u(x)$ over the integers, nor is their product $v_1(x)v_3(x)$ when the coefficients have been reduced modulo 169 to the range $(-\frac{169}{2}, \frac{169}{2})$. Thus we have exhausted all possibilities, proving once again that $u(x)$ is irreducible over the integers—this time using only its factorization modulo 13.

The example we have been considering is atypical in one important respect: We have been factoring the *monic* polynomial $u(x)$ in (22), so we could assume that all its factors were monic. What should we do if $u_n > 1$? In such a case, the leading coefficients of all but one of the polynomial factors can be varied almost arbitrarily modulo p^e; we certainly don't want to try all possibilities. Perhaps the reader has already noticed this problem. Fortunately there is a simple way out: the factorization $u(x) = v(x)w(x)$ implies a factorization $u_n u(x) = v_1(x)w_1(x)$ where $\ell(v_1) = \ell(w_1) = u_n = \ell(u)$. ("Do you mind if I multiply your polynomial by its leading coefficient before factoring it?") We can proceed essentially as above, but using $p^e > 2B$ where B now bounds the maximum coefficient for factors of $u_n u(x)$ instead of $u(x)$.

Putting these observations all together results in the following procedure:

F1. Find the unique squarefree factorization

$$u(x) \equiv \ell(u)v_1(x)\ldots v_r(x) \text{ (modulo } p^e),$$

where p^e is sufficiently large as explained above, and where the $v_j(x)$ are monic. (This will be possible for all but a few primes p, see exercise 23.) Also set $d \leftarrow 1$.

F2. For every combination of factors $v(x) = v_{i_1}(x)\ldots v_{i_d}(x)$, with $i_1 = 1$ if $d = \frac{1}{2}r$, form the unique polynomial $\bar{v}(x) \equiv \ell(u)v(x)$ (modulo p^e) whose coefficients all lie in the interval $[-\frac{1}{2}p^e, \frac{1}{2}p^e)$. If $\bar{v}(x)$ divides $\ell(u)u(x)$, output the factor $\mathrm{pp}(\bar{v}(x))$, divide $u(x)$ by this factor, and remove the corresponding $v_i(x)$ from the list of factors modulo p^e; decrease r by the number of factors removed, and terminate the algorithm if $d > \frac{1}{2}r$.

F3. Increase d by 1, and return to F2 if $d > \frac{1}{2}r$. ∎

At the conclusion of this process, the current value of $u(x)$ will be the final irreducible factor of the originally given polynomial. Note that if $|u_0| < |u_n|$, it is preferable to do all of the work with the reverse polynomial $u_0x^n + \cdots + u_n$, whose factors are the reverses of the factors of $u(x)$.

The procedure as stated requires $p^e > 2B$, where B is a bound on the coefficients of *any* divisor of $u_n u(x)$, but we can use a much smaller value of B if we only guarantee it to be valid for divisors of degree $\le \frac{1}{2}\deg(u)$. In this case the divisibility test in step F2 should be applied to $w(x) = v_1(x)\ldots v_r(x)/v(x)$ instead of $v(x)$, whenever $\deg(v) > \frac{1}{2}\deg(u)$.

The above algorithm contains an obvious bottleneck: We may have to test as many as 2^{r-1} potential factors $v(x)$. The average value of 2^r in a random situation is about n, or perhaps $n^{1.5}$ (see exercise 5), but in nonrandom situations we will want to speed up this part of the routine as much as we can. One way to rule out spurious factors quickly is to compute the trailing coefficient $\bar{v}(0)$ first, continuing only if this divides $\ell(u)u(0)$; the complication explained in the preceding paragraph does not have to be considered unless this divisibility condition is satisfied, since such a test is valid even when $\deg(v) > \frac{1}{2}\deg(u)$.

Another important way to speed up the procedure is to reduce r so that it tends to reflect the true number of factors. The distinct degree factorization algorithm above can be applied for various small primes p_j, thus obtaining for each prime a set D_j of possible degrees of factors modulo p_j; see exercise 26. We can represent D_j as a string of n binary bits. Now we compute the intersection $\bigcap D_j$, namely the logical "and" of these bit strings, and we perform step F2 only for $i_1 + \cdots + i_d \in \bigcap D_j$. Furthermore p is chosen to be that p_j having the smallest value of r. This technique is due to David R. Musser, whose experience suggests trying about five primes p_j (see *JACM* **25** (1978), 271–282). Of course we would stop immediately if the current $\bigcap D_j$ shows that $u(x)$ is irreducible.

Musser has given a complete discussion of a factorization method similar to the steps above, in *JACM* **22** (1975), 291–308. The procedure above incorporates an improvement suggested in 1978 by G. E. Collins, namely to look for trial divisors by taking combinations of d factors at a time rather than combinations of total degree d. This improvement is important because of the statistical behavior of the modulo-p factors of polynomials that are irreducible over the rationals (cf. exercise 37).

Greatest common divisors. Similar techniques can be used to calculate greatest common divisors of polynomials: If $\gcd\big(u(x), v(x)\big) = d(x)$ over the integers, and if $\gcd\big(u(x), v(x)\big) = q(x)$ (modulo p) where $q(x)$ is monic, then $d(x)$ is a common divisor of $u(x)$ and $v(x)$ modulo p; hence

$$d(x) \text{ divides } q(x) \quad (\text{modulo } p). \tag{24}$$

If p does not divide the leading coefficients of both u and v, it does not divide the leading coefficient of d; in such a case $\deg(d) \le \deg(q)$. When $q(x) = 1$ for such a prime p, we must therefore have $\deg(d) = 0$, and $d(x) = \gcd\big(\mathrm{cont}(u), \mathrm{cont}(v)\big)$. This justifies the remark made in Section 4.6.1 that the simple computation