MATHEMATICS OF COMPUTATION
VOLUME 42, NUMBER 165
JANUARY 1984, PAGES 331 340

Divisors in Residue Classes

By H. W. Lenstra, Jr.

Abstract. In this paper the following result is proved. Let r, s and n be integers satisfying $0 \leqslant r < s < n$, $s > n^{1/3}$, $\gcd(r, s) = 1$. Then there exist at most 11 positive divisors of n that are congruent to r modulo s. Moreover, there exists an efficient algorithm for determining all these divisors. The bound 11 is obtained by means of a combinatorial model related to coding theory. It is not known whether 11 is best possible; in any case it cannot be replaced by 5. Nor is it known whether similar results are true for significantly smaller values of $\log s/\log n$. The algorithm treated in the paper has applications in computational number theory.

In this paper we prove the following theorem.

THEOREM. *Let r, s and n be integers satisfying*

$$0 \leqslant r < s < n, \quad s > n^{1/3}, \quad \gcd(r, s) = 1.$$

Then there exist at most 11 *positive divisors of n that are congruent to r modulo s, and there is a polynomial algorithm for determining all these divisors.*

The algorithm referred to in the theorem is described in Section 1. It is *polynomial* in the sense that the number of bit operations required by the algorithm is bounded by a polynomial function of the binary length of n. More precisely, we shall see that this number of bit operations is $O((\log n)^3)$. Employing fast multiplication techniques we can improve this bound to $O((\log n)^{2+\varepsilon})$ for every $\varepsilon > 0$.

We mention two applications of the algorithm. In several primality testing algorithms (see [3], [7]), the number n to be tested is subjected to a collection of "pseudo-prime" tests. If n does not pass all these tests it is composite. If n does pass all these tests, one knows that each divisor of n lies in one of a small and explicitly known set of residue classes modulo an auxiliary number s. In the latter case, all divisors of n can easily be found if s satisfies the condition $s > n^{1/2}$. Our algorithm shows that the same can be done if s satisfies the weaker condition $s > n^{1/3}$. In special cases this observation was already made in [2, Theorems 5 and 17].

The second application is to the related problem of factoring n. Choosing s to be a suitable integer exceeding $n^{1/3}$ and applying our algorithm to all residue classes $r \bmod s$, we obtain an algorithm that factors n in time $O(n^{(1/3)+\varepsilon})$ for every $\varepsilon > 0$. The same bound was achieved by Lehman [6] and, conjecturally, by Pinter [9], by methods that are similar in spirit. There exist better factoring methods, both in theory and in practice (see [7]), but this application indicates at least that it may be difficult to extend the algorithm to significantly smaller values of s.

Received April 27, 1983.
1980 *Mathematics Subject Classification*. Primary 10H20, 10A25, 94B25.
Key words and phrases. Divisors, residue classes, coding theory, computational number theory.

0025-5718/84 $1.00 + $.25 per page

For the purposes of these two applications, the restrictive condition $\gcd(r, s) = 1$ is clearly not an essential limitation. In the theorem, however, this condition cannot be omitted. To see this, we remark that for odd n the divisors of n^2 that are congruent to n modulo $2n$ are in one-to-one correspondence with the divisors of n. Their number is not bounded by 11, and not even by a polynomial function of $\log(n^2)$, by [4, Theorem 317]; so they cannot be determined by a polynomial algorithm.

In Section 2 we discuss a combinatorial problem that is related to coding theory. Using the results of Section 2 we complete the proof of the theorem in Section 3. More generally, it is proved that for every real number $\alpha > \frac{1}{4}$ there exists a number $c(\alpha)$ with the following property: if r, s, n are positive integers satisfying

$$\gcd(r, s) = 1, \qquad s > n^{\alpha},$$

then the number of positive divisors of n that are $r \bmod s$ is at most $c(\alpha)$. I do not know whether the same result holds for *any* positive α.

The value 11 in the theorem is the best that can be obtained by our method of proof, but it is not clear whether it is best possible. All we know is that it cannot be replaced by 5, as is shown in Section 3 by means of examples.

Acknowledgements are due to H. Cohen, P. Erdös, B. J. Lageweg, A. K. Lenstra, A. M. Odlyzko, C. Pomerance, D.B. Zagier and H. Zantema, who all contributed in one way or another to the contents of this paper.

1. The Algorithm. Let r, s and n be as in the theorem. Before we describe the algorithm referred to in the theorem we briefly sketch the underlying idea. We look for divisors of n of the form $xs + r$, so we have to solve the equation

$$(xs + r)(ys + r') = n \tag{1.1}$$

in nonnegative integers x, y; here r' is such that $rr' \equiv n \bmod s$. Viewing (1.1) modulo s^2, we obtain a congruence for $xr' + yr$ modulo s. This congruence can be used to obtain a series of congruences of the form

$$xa_i + yb_i \equiv c_i \bmod s.$$

Using that $s > n^{1/3}$, one proves that for some i the number $xa_i + yb_i$ is so small that this leaves only a few possible values for $xa_i + yb_i$. For each fixed value, x or y can be eliminated from (1.1), and the resulting quadratic equation can be solved.

(1.2) *Algorithm.* Given r, s and n as in the theorem, this algorithm determines all positive divisors of n that are congruent to r modulo s.

First apply the Euclidean algorithm to calculate an integer r^* satisfying $r^*r \equiv 1 \bmod s$, see [5, p. 325], and determine the integer r' by

$$r' \equiv r^*n \bmod s, \qquad 0 \leqslant r' < s.$$

Secondly, for $i = 0, 1, 2, \ldots$ do the following. Calculate a_i, b_i, c_i from the formulae

$$\begin{aligned} &a_0 = s, \quad b_0 = 0, \quad c_0 = 0, \\ &a_1 \equiv r'r^* \bmod s, \qquad 0 < a_1 \leqslant s, \\ &b_1 = 1, \\ &c_1 \equiv \frac{n - rr'}{s} \cdot r^* \bmod s, \end{aligned}$$

and if $i \geqslant 2$

$$a_i = a_{i-2} - q_i a_{i-1},$$
$$b_i = b_{i-2} - q_i b_{i-1},$$
$$c_i \equiv c_{i-2} - q_i c_{i-1} \bmod s,$$

where q_i is the unique integer for which

$$0 \leqslant a_i < a_{i-1} \quad \text{if } i \text{ is } \textit{even},$$
$$0 < a_i \leqslant a_{i-1} \quad \text{if } i \text{ is } \textit{odd}.$$

Next, for each integer c satisfying

$$\text{(1.3)} \qquad \begin{cases} c \equiv c_i \bmod s, \\ |c| < s \quad \text{if } i \text{ is even}, \\ 2a_i b_i \leqslant c \leqslant \dfrac{n}{s^2} + a_i b_i \quad \text{if } i \text{ is odd}, \end{cases}$$

solve the pair of equations

$$\text{(1.4)} \qquad \begin{cases} xa_i + yb_i = c, \\ (xs + r)(ys + r') = n \end{cases}$$

(see (1.5)), and if x and y are found to be nonnegative integers add $xs + r$ to the list of divisors of n that are $r \bmod s$. If $a_i = 0$, then the algorithm stops at this point; otherwise, proceed with the next value of i.

This finishes the description of Algorithm (1.2). The correctness will be proved below, see (1.7).

(1.5) It is easily seen that the system (1.4) can be reduced to a single quadratic equation in one variable. Explicitly, if we put

$$u = a_i(xs + r), \qquad v = b_i(ys + r'),$$

then

$$uv = a_i b_i n, \qquad u + v = cs + a_i r + b_i r',$$

so u, v are the zeros of the polynomial

$$T^2 - (cs + a_i r + b_i r')T + a_i b_i n.$$

We remark that the numbers a_i, b_i appearing in the algorithm are computed by means of the extended Euclidean algorithm (see [5, p. 325]) applied to s, a_1. Therefore the termination condition $a_i = 0$ is satisfied for some value of i, and, denoting this value by t, we have $t = O(\log s)$, by [5, p. 343]. Since $a_i > 0$ for odd i, the number t is *even*.

The following properties of a_i, b_i are easily verified by induction:

$$(a_i, b_i) \in \mathbf{Z}_{>0} \times \mathbf{Z}_{>0} \quad \text{for } i \textit{ odd}, 0 < i < t,$$
$$(a_i, b_i) \in (\mathbf{Z}_{\geqslant 0} \times \mathbf{Z}_{\leqslant 0}) - \{(0,0)\} \quad \text{for } i \textit{ even}, 0 \leqslant i \leqslant t,$$
$$b_{i+1}a_i - a_{i+1}b_i = (-1)^i s \quad \text{for } 0 \leqslant i < t.$$

Before we prove the correctness of the algorithm we treat a lemma.

(1.6) LEMMA. *Let a_i, b_i, t be as above, and let $x, y \in \mathbf{R}_{\geqslant 0}$, $\gamma \in \mathbf{R}_{>0}$. Then there exists $i \in \{0, 1, \ldots, t\}$ such that*

$$-\gamma s < xa_i + yb_i < \gamma s \quad \text{if } i \text{ is even},$$
$$2\gamma a_i b_i \leqslant xa_i + yb_i \leqslant \gamma^{-1}xy + \gamma a_i b_i \quad \text{if } i \text{ is odd}.$$

Proof. First we consider the numbers $xa_i + yb_i$ for *even* values of i. From $b_0 = 0$, $a_t = 0$, it follows that

$$xa_0 + yb_0 \geqslant 0, \qquad xa_t + yb_t \leqslant 0.$$

Therefore there is an even index i such that

$$xa_i + yb_i \geqslant 0, \qquad xa_{i+2} + yb_{i+2} \leqslant 0.$$

If one of these numbers is less than γs in absolute value we are done. Assume therefore that the first is $\geqslant \gamma s$ and that the second is $\leqslant -\gamma s$. Then

$$(xa_i + yb_i)/\gamma \geqslant s = b_{i+1}a_i - a_{i+1}b_i \geqslant b_{i+1}a_i,$$

so $x \geqslant \gamma b_{i+1}$, and

$$(xa_{i+2} + yb_{i+2})/\gamma \leqslant -s = b_{i+2}a_{i+1} - a_{i+2}b_{i+1} \leqslant b_{i+2}a_{i+1},$$

so $y \geqslant \gamma a_{i+1}$. Therefore we have

$$xa_{i+1} + yb_{i+1} \geqslant 2\gamma a_{i+1}b_{i+1},$$

and from $(x - \gamma b_{i+1})(y - \gamma a_{i+1}) \geqslant 0$ it follows that

$$xa_{i+1} + yb_{i+1} \leqslant \gamma^{-1}xy + \gamma a_{i+1}b_{i+1}.$$

Since $i + 1$ is odd this concludes the proof of the lemma.

(1.7) PROPOSITION. *Given r, s and n as in the theorem, Algorithm* (1.2) *correctly determines all positive divisors of n that are congruent to r modulo s. The number of bit operations required by the algorithm is $O((\log n)^3)$, and $O((\log n)^{2+\varepsilon})$ for any $\varepsilon > 0$ if fast multiplication techniques are used.*

Proof. First we prove the correctness of the algorithm. Let $xs + r$ be a positive divisor of n that is r modulo s. Then $x \in \mathbf{Z}_{\geqslant 0}$, and $(xs + r)d = n$ for some $d \in \mathbf{Z}_{>0}$. Multiplying by r^*, we see that $d \equiv r^*n \equiv r' \bmod s$, so we can write $d = ys + r'$ with $y \in \mathbf{Z}_{\geqslant 0}$. Viewing $(xs + r)(ys + r') = n$ modulo s^2, we obtain $xr' + yr \equiv (n - rr')/s \bmod s$; notice that the right-hand side is an integer. Multiplying by r^*, we find that

$$xr'r^* + y \equiv \frac{n - rr'}{s} \cdot r^* \bmod s.$$

This is exactly the case $i = 1$ of the series of congruences

$$xa_i + yb_i \equiv c_i \bmod s \qquad (0 \leqslant i \leqslant t). \tag{1.8}$$

For $i = 0$ this congruence is trivially satisfied, and for $i \geqslant 2$ it follows by a straightforward inductive argument from the definition of a_i, b_i, c_i.

Applying Lemma (1.6) with $\gamma = 1$, we find that there exists $i \in \{0, 1, \ldots, t\}$ such that

$$|xa_i + yb_i| < s \quad \text{if } i \text{ is even},$$
$$2a_i b_i \leqslant xa_i + yb_i \leqslant xy + a_i b_i \quad \text{if } i \text{ is odd}.$$

Fix such a value of i, and put $c = xa_i + yb_i$. From (1.8), the inequalities just stated and

$$xy \leqslant (xs + r)(ys + r')/s^2 = n/s^2$$

it then follows that c satisfies (1.3). Since x, y satisfy (1.4), this implies that the divisor $xs + r$ is indeed discovered by the algorithm. This proves the correctness.

Next we estimate the number of bit operations. The determination of r^* can be done in $O((\log n)^2)$ bit operations, see [5, Exercise 4.5.2.30]. From $n/s^2 < s$ and $a_i b_i > 0$, for odd i, it follows that for each $i \in \{0, 1, \ldots, t\}$ there are at most two values of c that satisfy (1.3). Hence for each i the algorithm requires only a bounded number of additions, subtractions, multiplications, divisions and square root extractions. These operations are performed on integers whose binary length is $O(\log n)$, so each of them can be done in $O((\log n)^2)$ bit operations, or $O((\log n)^{1+\varepsilon})$ with fast multiplication techniques; see [1]. Since the number of values for i is $t + 1 = O(\log n)$, this proves the proposition.

(1.9) *Remarks.* (a) The proof shows that the algorithm is also polynomial if $s/n^{1/3}$ is bounded from below.

(b) We applied Lemma (1.6) only with $\gamma = 1$. It may be that another choice of γ gives rise to a faster algorithm in practice.

(c) If s is much larger than $n^{1/3}$, then the number of quadratic equations to be solved can be greatly reduced. For example, if $s > n^{1/2}$, then $xy \leqslant n/s^2 < 1$, so we need only consider the cases $x = 0$ and $y = 0$. If $s > n^{2/5}$, one may use the fact that $a_i^2 + b_i^2 \leqslant (4/3)^{1/2}s$ for some i (see [5, Exercises 3.3.4.5 and 9]); for that value of i, the number $xa_i + yb_i$ is in an interval of length at most a constant multiple of s, unless $xy = 0$, so only a bounded number of quadratic equations need be solved. More generally, if $s > n^{\alpha}$, with $\alpha > \frac{1}{3}$, then the algorithm can be modified in such a way that the number of quadratic equations to be solved is bounded by a constant only depending on α. This observation is due to H. Zantema.

2. A Combinatorial Model. We denote by $-$ and $\triangle$ set-theoretic difference and symmetric difference, respectively: $X \triangle Y = (X - Y) \cup (Y - X)$. The cardinality of a set X is denoted by $\#X$.

A *weight function* on a finite set V is a function w that assigns a nonnegative real number to every subset of V, in such a way that $w(X \cup Y) = w(X) + w(Y)$ for any two disjoint subsets X, Y of V.

(2.1) PROPOSITION. *Let V be a finite set, w a weight function on V with $w(V) > 0$, and $\alpha \in \mathbf{R}$, $\alpha > \frac{1}{4}$. Let further $\mathfrak{D}$ be a system of subsets of V such that*

$$\max\{w(D - D'), w(D' - D)\} \geqslant \alpha \cdot w(V)$$

for all D, $D' \in \mathfrak{D}$ with $D \neq D'$. Then $\#\mathfrak{D} \leqslant c(\alpha)$, where $c(\alpha)$ is a constant that only depends on α.

(2.2) *Remark.* The conclusion of (2.1) does not hold for $\alpha \leqslant \frac{1}{4}$. To see this, let V be a vector space over the two element field $\mathbf{F}_2$, and let $\mathfrak{D}$ be the collection of hyperplanes in V. Put $w(X) = \#X$, for $X \subset V$. Then $w(D - D') = \frac{1}{4}\#V \geqslant \alpha \cdot w(V)$ for any two D, $D' \in \mathfrak{D}$ with $D \neq D'$, but $\#\mathfrak{D} = \#V$ tends to infinity with the dimension of the vector space.

Proof of (2.1). Choose ε fixed with $0 < \varepsilon < 2\alpha - \frac{1}{2}$, and let $\eta = 4\alpha - 1 - 2\varepsilon$; so $\eta > 0$. We write

$$\mathfrak{D}_\beta = \{D \in \mathfrak{D}: \beta \cdot w(V) \leqslant w(D) < (\beta + \varepsilon) \cdot w(V)\}$$

for $\beta \in \mathbf{R}$, $0 \leqslant \beta \leqslant 1$. Below we shall prove that

$$\#\mathfrak{D}_\beta \leqslant 1 + \eta^{-1} \tag{2.3}$$

for all β. Since we have

$$\mathfrak{D} = \bigcup_{i=0}^{[1/\varepsilon]} \mathfrak{D}_{i\varepsilon},$$

this implies that

$$\#\mathfrak{D} \leqslant ([1/\varepsilon] + 1)(1 + \eta^{-1}),$$

as required.

We prove (2.3). Let $D, D' \in \mathfrak{D}_\beta$, $D \neq D'$. Then $w(D)$ and $w(D')$ differ by less than $\varepsilon \cdot w(V)$ in absolute value. Subtracting $w(D \cap D')$, we see that also $w(D - D')$ and $w(D' - D)$ differ by less than $\varepsilon \cdot w(V)$. Moreover, the largest of $w(D - D')$, $w(D' - D)$ is at least $\alpha \cdot w(V)$, by hypothesis. Hence the smallest is at least $(\alpha - \varepsilon) \cdot w(V)$, and

$$\begin{aligned} w(D \triangle D') &= w(D - D') + w(D' - D) \\ &\geqslant (2\alpha - \varepsilon) \cdot w(V) = \tfrac{1}{2}(1 + \eta) \cdot w(V). \end{aligned}$$

Write $\mathfrak{D}_\beta = \{D_1, D_2, \ldots, D_m\}$ with $m = \#\mathfrak{D}_\beta$. The inequality just proved implies that

$$\sum_{1 \leqslant i < j \leqslant m} w(D_i \triangle D_j) \geqslant \frac{1}{2}\binom{m}{2}(1 + \eta)w(V).$$

On the other hand, we have

$$\begin{aligned} &\sum_{1 \leqslant i < j \leqslant m} w(D_i \triangle D_j) \\ &\quad = \sum_{x \in V} \#\{(i, j): 1 \leqslant i < j \leqslant m \text{ and } x \in D_i \triangle D_j\} \cdot w(\{x\}) \\ &\quad = \sum_{x \in V} \#\{(i, j): 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant m, x \in D_i, x \notin D_j\} \cdot w(\{x\}) \\ &\quad = \sum_{x \in V} m_x \cdot (m - m_x) \cdot w(\{x\}), \end{aligned}$$

where $m_x = \#\{i: 1 \leqslant i \leqslant m, x \in D_i\}$. From $m_x \cdot (m - m_x) \leqslant \frac{1}{4}m^2$ we now see that

$$\sum_{1 \leqslant i < j \leqslant m} w(D_i \triangle D_j) \leqslant \frac{1}{4}m^2 \sum_{x \in V} w(\{x\}) = \frac{1}{4}m^2 w(V).$$

Combined with the earlier inequality this gives

$$\begin{aligned} &\frac{1}{2}\binom{m}{2}(1 + \eta)w(V) \leqslant \frac{1}{4}m^2 w(V), \\ &(m - 1)(1 + \eta) \leqslant m, \\ &m \leqslant 1 + \eta^{-1}, \end{aligned}$$

as required. This proves (2.3) and (2.1).

Remark. Notice the resemblance of the above proposition to Plotkin's bound in coding theory, see [8, Chapter 2, Section 2].

(2.4) PROPOSITION. *Let* $V, w, \mathfrak{D}, \alpha$ *satisfy all hypotheses of Proposition* (2.1), *and suppose moreover that* $\alpha > 1/3$. *Then* $\#\mathfrak{D} \leqslant 11$.

(2.5) *Remark.* This proposition is best possible in the sense that for $\alpha \leqslant 1/3$ we may have $\#\mathfrak{D} \geqslant 12$. To see this, let $\#V = 6$ and let $\mathfrak{D}$ be the system of subsets whose characteristic functions are given by the columns of the following matrix:

$$\begin{array}{cccccccccccc}
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1.
\end{array}$$

In this example, we take $w(X) = \#X$ for all $X \subset V$.

Before we prove (2.4) we treat two lemmas.

(2.6) LEMMA. *Let* $V_1, V_2, \ldots, V_l \subset V$ *and* $t \in \mathbf{Z}$. *Then*

$$\frac{1}{2}t(t+1) \cdot w\left(\bigcup_{i=1}^{l} V_i\right) + \sum_{1 \leqslant i < j \leqslant l} w(V_i \cap V_j) \geqslant t \cdot \sum_{i=1}^{l} w(V_i).$$

Proof. For every $y \in \mathbf{Z}$ we clearly have $\frac{1}{2}(y-t)(y-(t+1)) \geqslant 0$, which is the same as

$$\tfrac{1}{2}t(t+1) + \tfrac{1}{2}y(y-1) \geqslant ty.$$

We apply this to

$$y = n_x = \#\{i\colon 1 \leqslant i \leqslant l, x \in V_i\}$$

for $x \in \bigcup_{i=1}^{l} V_i$. Multiplying the resulting inequality by $w(\{x\})$ and summing over $x \in \bigcup_{i=1}^{l} V_i$, we obtain precisely the inequality stated in the lemma. This proves (2.6).

(2.7) LEMMA. *Let the hypotheses be as in* (2.1), *and let* $V_1, V_2, \ldots, V_l \in \mathfrak{D}$ *satisfy*

$$w(V_1) \leqslant w(V_2) \leqslant \cdots \leqslant w(V_l), \qquad V_i \neq V_j \quad (1 \leqslant i < j \leqslant l).$$

Then the numbers $y_i = w(V_i)/w(V)$ *satisfy for every* $t \in \mathbf{Z}$ *the inequality*

$$-ty_1 + (-t+1)y_2 + \cdots + (-t+l-1)y_l + \tfrac{1}{2}t(t+1) \geqslant \tfrac{1}{2}l(l-1)\alpha.$$

Proof. This follows in a straightforward way from the previous lemma, if we use that

$$w\left(\bigcup_{i=1}^{l} V_i\right) \leqslant w(V),$$

$$w(V_i \cap V_j) \leqslant w(V_j) - \alpha \cdot w(V) \quad \text{for } 1 \leqslant i < j \leqslant l,$$

the last inequality coming from the hypothesis on $\mathfrak{D}$ in (2.1). This proves (2.7).

Proof of (2.4). Suppose that $\#\mathfrak{D} \geqslant 12$, and choose $D_1, D_2, \ldots, D_{12} \in \mathfrak{D}$ such that

$$w(D_1) \leqslant w(D_2) \leqslant \cdots \leqslant w(D_{12}), \qquad D_i \neq D_j \quad (1 \leqslant i < j \leqslant 12).$$

Write $x_i = w(D_i)/w(V)$. Applying (2.7) to $\{V_1, V_2\} = \{D_1, D_2\}$, $t = 0$ and to $\{V_1, V_2\} = \{D_{11}, D_{12}\}$, $t = 1$ we find that

$$x_2 \geqslant \alpha, \qquad x_{11} \leqslant 1 - \alpha.$$

With $\{V_1, V_2, \ldots, V_t\} = \{D_2, D_3, D_4, D_5, D_6, D_7\}$, $t = 2$ we obtain

$$-2x_2 - x_3 + x_5 + 2x_6 + 3x_7 + 3 \geqslant 15\alpha,$$

and $\{V_1, V_2, \ldots, V_t\} = \{D_6, D_7, D_8, D_9, D_{10}, D_{11}\}$, $t = 3$ leads to

$$-3x_6 - 2x_7 - x_8 + x_{10} + 2x_{11} + 6 \geqslant 15\alpha.$$

Adding the last two inequalities and using that $x_3 \geqslant x_2 \geqslant \alpha$, $x_{10} \leqslant x_{11} \leqslant 1 - \alpha$, we find that

$$-3\alpha + x_5 - x_6 + x_7 - x_8 + 3(1 - \alpha) + 9 \geqslant 30\alpha.$$

Since $x_5 \leqslant x_6$ and $x_7 \leqslant x_8$, this yields

$$12 \geqslant 36\alpha,$$

a contradiction. This proves (2.4).

(2.8) For an integer $k \geqslant 2$, let $\alpha(k)$ be the largest value of α for which the hypotheses of (2.1) can be satisfied with $\#\mathfrak{V} = k$. It is not difficult to see that $\alpha(k)$ exists and that, for given k, it can be computed by solving a linear programming problem with $2^k + 1$ variables.

From (2.4) and (2.5) we see that $\alpha(12) = \frac{1}{3}$. Table 1 shows the values of $\alpha(k)$ for $2 \leqslant k \leqslant 12$. The table was obtained as follows. The fact that the tabulated values are upper bounds for $\alpha(k)$ was shown with linear programming techniques; the help of B. J. Lageweg is gratefully acknowledged. In all cases except $k = 9$ the inequalities from (2.7) were sufficient to obtain these upper bounds. The fact that the tabulated values are lower bounds for $\alpha(k)$ was next shown by H. Zantema, who exhibited examples as in (2.5).

If $\alpha > \alpha(k)$, then in (2.1) we can take $c(\alpha) = k - 1$. From (2.1) and (2.2) it follows that $\alpha(k)$ tends to $\frac{1}{4}$ for k tending to infinity. The proof of (2.1) shows that we can take $c(\alpha) = O((\alpha - \frac{1}{4})^{-2})$ for $\frac{1}{4} < \alpha \leqslant 1$, so $\alpha(k) = \frac{1}{4} + O(k^{-1/2})$, but I do not know whether this is the correct rate of convergence.

TABLE 1

k	$\alpha(k)$	
2	1	= 1.000000
3	1/2	= 0.500000
4	1/2	= 0.500000
5	2/5	= 0.400000
6	2/5	= 0.400000
7	3/8	= 0.375000
8	4/11	= 0.363636
9	13/37	= 0.351351
10	9/26	= 0.346154
11	31/92	= 0.336957
12	1/3	= 0.333333

3. Proof of the Theorem. For a positive integer k we put

$$V(k) = \{ p^t : p \text{ prime}, t \in \mathbf{Z}, t \geqslant 1, p^t \text{ divides } k\},$$

e.g. $V(12) = \{2, 4, 3\}$. We define a weight w on each set $V(k)$ by putting $w(\{p^t\}) = \log p$. An easy calculation shows that $w(V(k)) = \log k$.

Proof of the Theorem. Since the last assertion of the theorem was proved in Section 1, see (1.7), it suffices to prove the first assertion.

We apply (2.4) to $V = V(n)$, with w as above. We have $w(V) > 0$ if $n > 1$, which may clearly be assumed. We take $\mathfrak{D} = \{V(d) : d \text{ divides } n, d > 0, d \equiv r \bmod s\}$.

Let d, d' be two distinct positive divisors of n that are $r \bmod s$. Since s divides $d - d'$ and is coprime to d, the greatest common divisor of d and d' divides $(d - d')/s$. Therefore we have

$$\gcd(d, d') \leqslant \frac{|d - d'|}{s} < \frac{\max\{d, d'\}}{n^{1/3}},$$

so

$$\tfrac{1}{3} \log n < \max\{\log(d/\gcd(d, d')), \log(d'/\gcd(d, d'))\}.$$

Since $V(\gcd(d, d')) = V(d) \cap V(d')$, this leads to

$$\tfrac{1}{3} w(V) < \max\{w(V(d) - V(d')), w(V(d') - V(d))\}.$$

Hence we can choose $\alpha > \frac{1}{3}$ such that the right-hand side is $\geqslant \alpha \cdot w(V)$ for all pairs d, d'. Then all hypotheses of (2.4) are satisfied, and therefore $\#\mathfrak{D} \leqslant 11$.

This completes the proof of the theorem.

(3.1) PROPOSITION. *For every $\alpha \in \mathbf{R}$ with $\alpha > \frac{1}{4}$ there exists a constant $c(\alpha)$ with the following property. If r, s, n are integers satisfying*

$$n > 0, \quad s > n^{\alpha}, \quad \gcd(r, s) = 1,$$

then the number of positive divisors of n that are congruent to r modulo s is at most $c(\alpha)$.

Proof. The proof is similar to the proof just given, with (2.4) replaced by (2.1). This proves (3.1).

If $\alpha \geqslant \alpha(k)$, with $\alpha(k)$ as in (2.8), then we can take $c(\alpha) = k - 1$ in Proposition (3.1). I do not know whether the condition $\alpha > \frac{1}{4}$ in (3.1) can be replaced by $\alpha > 0$.

In the theorem, the value 11 cannot be replaced by 5. In fact, H. Cohen proved that there exist infinitely many positive integers n that have at least six positive divisors in the same coprime residue class modulo a number $s > n^{1/3}$. The first ten values of n are listed in Table 2, together with the residue classes $r \bmod s$ that contain six divisors of n. The table was computed by A. K. Lenstra with the help of the VAX 11-780 computer at the Mathematical Centre in Amsterdam. No further examples with $n \leqslant 3 \cdot 10^6$ exist, and no example with *seven* divisors in the same residue class was found.

TABLE 2

n	s	r	n	s	r
245784	65	1, 19	1755600	131	2, 100
288288	71	1, 28	1796760	137	3, 93
320320	69	1, 22	2066400	143	2, 25
480480	83	5, 65	2511600	149	7, 8
911064	115	1, 34	2841696	175	2, 23

Mathematisch Instituut
Universiteit van Amsterdam
Roetersstraat 15
1018 WB Amsterdam, The Netherlands

1. H. ALT, "Square rooting is as difficult as multiplication," *Computing*, v. 21, 1979, pp. 221–232.

2. J. BRILLHART, D. H. LEHMER & J. L. SELFRIDGE, "New primality criteria and factorizations of $2^m \pm 1$," *Math. Comp.*, v. 29, 1975, pp. 620–647; pp. 112–139 in: *Selected Papers of D. H. Lehmer*, Vol. I, Charles Babbage Research Centre, St. Pierre, Manitoba, 1981.

3. H. COHEN & H. W. LENSTRA, JR., "Primality testing and Jacobi sums," *Math. Comp.*, v. 42, 1984, pp. 297–330.

4. G. H. HARDY & E. M. WRIGHT, *An Introduction to the Theory of Numbers*, 5th ed., Oxford Univ. Press, Oxford, 1979.

5. D. E. KNUTH, *The Art of Computer Programming*, Vol. 2, *Seminumerical Algorithms*, 2nd ed., Addison-Wesley, Reading, Mass., 1981.

6. R. S. LEHMAN, "Factoring large integers," *Math. Comp.*, v. 28, 1974, pp. 637–646.

7. H. W. LENSTRA, JR. & R. TIJDEMAN (editors), *Computational Methods in Number Theory*, Mathematical Centre Tracts 154/155, Amsterdam, 1982.

8. F. J. MACWILLIAMS & N. J. A. SLOANE, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1978.

9. R. Y. PINTER, *Using Hyperbolic Tangents in Integer Factoring*, thesis, M.I.T., Cambridge, Mass., 1980.