A proof of the conjecture of Cohen and Mullen on sums of primitive roots

Stephen D. Cohen
School of Mathematics and Statistics,
University of Glasgow, Scotland
Stephen.Cohen@glasgow.ac.uk

Tomás Oliveira e Silva
Departamento de Electrónica, Telecomunicações e Informática / IEETA
University of Aveiro, Portugal
tos@ua.pt

Tim Trudgian*
Mathematical Sciences Institute
The Australian National University, ACT 0200, Australia
timothy.trudgian@anu.edu.au

February 9, 2014

Abstract

We prove that for all $q > 61$, every non-zero element in the finite field $\mathbb{F}_q$ can be written as a linear combination of two primitive roots of $\mathbb{F}_q$. This resolves a conjecture posed by Cohen and Mullen.

AMS Codes: 11T30, 11Y99

1 Introduction

For q a prime power, let $\mathbb{F}_q$ denote the finite field of order q, and let $g_1, g_2, \ldots, g_{\phi(q-1)}$ denote the primitive roots of q. Various questions have been asked about whether non-zero elements of $\mathbb{F}_q$ can be written as a linear sum of two primitive roots, g_1 and g_2. To develop this idea, let a, b and c be arbitrary non-zero elements in $\mathbb{F}_q$. Is there some q_0 such that there is always one representation

$$a = bg_n + cg_m \tag{1}$$

for all $q > q_0$? Since such a representation is possible if and only if $a/b = g_n + c/b\, g_m$, we may suppose that $b = 1$ in (1). Accordingly, define $\mathcal{G}$ to be the set of prime powers q such that for all non-zero $a, c \in \mathbb{F}_q$ there exists a primitive root $g \in \mathbb{F}_q$ such that $a - cg$ is also a primitive root of $\mathbb{F}_q$.

*Supported by Australian Research Council DECRA Grant DE120100173.

It appears that Vegh [10] was the first to consider a specific form of (1), namely, that with $b = 1$ and $c = -1$. This has been referred to as Vegh's Conjecture — see [7, §F9]. Vegh verified his own conjecture for $61 < q < 2000$; Szalay [9] proved it for $q > q_0$ and claimed that one could take $q_0 = 10^{19}$. In the special case when $a = 1$, Cohen [2] proved Vegh's conjecture for all $q > 7$.

Golomb [6] proposed (1) with $b = 1$ and $c = 1$. This has applications to Costas arrays, which appear in the study of radar and sonar signals. This was proved by Sun [8] for $q > 2^{60} \approx 1.15 \times 10^{18}$.

Cohen and Mullen [5] considered (1) in its most general form, namely $b = 1$ and arbitrary non-zero c and a. Cohen [3] calls this 'Conjecture G'. Cohen and Mullen proved Conjecture G for all $q \geq 4.79 \times 10^8$; Cohen [3] proved it for all $q \geq 3.854 \times 10^7$, and states that it is true for even $q > 4$; it is false for $q = 4$. Chou, Mullen, Shiue and Sun [1] tested it for odd $q < 2130$ and found that it failed only for $q = 3$, 5, 7, 11, 13, 19, 31, 43, and 61. Thus, in effect, Conjecture G can be interpreted as claiming that all prime powers exceeding 61 lie in the set $\mathcal{G}$. What this means is that 'all' one needs to do is to check (1) for $2130 \leq q \leq 3.854 \times 10^7$.

We improve on Cohen's method, given in [3], to isolate easily the possible counterexamples to Conjecture G. We compile an initial list of values of q that may need checking. We then examine this list in more detail, sieving out some values of q. This produces a secondary list of only 777 values of q with $2131 \leq q \leq 2762761$. This list is just small enough to enable us to verify (1) for each q. The result is:

Theorem 1. *For $q > 61$ and for arbitrary non-zero elements a, b, c of $\mathbb{F}_q$, there is always one representation of the form*

$$a = bg_n + cg_m,$$

where g_n and g_m are primitive roots of $\mathbb{F}_q$.

2 Theory

Let $\omega(n)$ denote the number of distinct prime factors of n so that $W(n) = 2^{\omega(n)}$ is the number of square-free divisors of n. Also, let $\theta(n) = \prod_{p|n}(1 - p^{-1})$. From the working of [3] (see also [5]) one can conclude that a prime power $q \in \mathcal{G}$ if $q > W(q-1)^4$ and hence if $\omega(q-1) \geq 16$ or $q > 2^{60}$. More significantly, a sieving method was given yielding improved lower bounds for q guaranteeing membership of $\mathcal{G}$. Instead of $W(q-1)$, these depend on appropriate choices of divisors e_1, e_2 of $q-1$ and the quantities $W(e_i)$ and θ_{e_i}, $i = 1, 2$, and can be applied to successively smaller values of $\omega(q-1) \leq 15$. In particular, it was shown that, if $\omega(q-1) \geq 9$, then $q \in \mathcal{G}$. Further, for each value of $\omega(q-1) \leq 8$ an upper bound can be derived on the set of prime power values requiring further analysis.

It turns out that the lists of possible exceptions that are thereby obtained from the method of [3] are small enough for direct computer verification on contemporary computer hardware. However, we can do better, as we now proceed to show. For any integer n define its radical $\mathrm{Rad}(n)$ as the product of all distinct prime factors of n. In the appendix we prove

Theorem 2. *Let $q \geq 4$ be a prime power. Let e be a divisor of $q-1$. If $\mathrm{Rad}(e) = \mathrm{Rad}(q-1)$ set $s = 0$ and $\delta = 1$. Otherwise, let $p_1, \ldots, p_s$, $s \geq 1$, be the primes dividing $q-1$ but not e and set $\delta = 1 - 2\sum_{i=1}^{s} p_i^{-1}$. Suppose that δ is positive and that*

$$q > \left(\frac{2s-1}{\delta} + 2\right)^2 W(e)^4. \tag{2}$$

Then $q \in \mathcal{G}$.

As an example of the usefulness of this theorem, consider the case $\omega(q-1) = 8$. For $s = 5$ we have $W(e) = 8$ and $\delta \geq 1 - 2\left(\frac{1}{7} + \frac{1}{11} + \frac{1}{13} + \frac{1}{17} + \frac{1}{19}\right)$. Therefore the right hand side of (2) is at most 14647129.006, and so $q > 14647129$ guarantees membership in $\mathcal{G}$ when $\omega(q-1) = 8$. Moreover, up to 14647129 there is only one prime power with $\omega(q-1) = 8$, namely $q = 13123111 = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 19 \cdot 23 + 1$. Using again Theorem 2, still with $s = 5$ but this time with $\delta = 1 - 2\left(\frac{1}{7} + \frac{1}{11} + \frac{1}{13} + \frac{1}{19} + \frac{1}{23}\right)$ tailored to the specific value of q, allows us to conclude that $13123111 \in \mathcal{G}$.

We repeat the procedure for $1 \leq \omega(q-1) \leq 7$, and $q \geq 2130$, each time noting the upper bound of the intervals that need further analysis. These are reported in the second column in Table 1. We then enumerate all possible q and eliminate as many values as we can by checking if (2) is true for some value of s. We are left with a final list of values of q that need checking. Column 3 of Table 1 contains the largest elements of these lists; Columns 4 and 5 contain respectively the initial and final number of elements of these lists, discriminating primes (on the left of the summation sign) and prime powers (on the right of the summation sign).

Table 1: Improved bounds for q

$\omega(q-1)$	Upper bound	Largest q	Initial list size	Final list size
8	14647129	—	$1 + 0$	$0 + 0$
7	3402711	2762761	$78 + 1$	$22 + 1$
6	947062	840841	$635 + 6$	$162 + 4$
5	238715	231001	$1741 + 21$	$290 + 9$
4	34124	33601	$1024 + 24$	$259 + 10$
3	3441	4057	$84 + 5$	$16 + 4$

It is worthwhile to mention that the largest value that needs to be verified (2762761) is considerably smaller than the largest value that, according to [3], would have to be verified (25555531).

3 Computation

Let $q = p^k$, where p is a prime. When $k = 1$ the full finite field machinery is not needed to test Conjecture G. For efficiency reasons, we thus developed two programs to test it: one to deal with the case $k = 1$, and another to deal with the case $k > 1$.

3.1 Verification of Conjecture G when $q = p$

One way to verify Conjecture G for a given value of p is to call Algorithm 1 (or Algorithm 2) for $c = 1, 2, \ldots, p-1$. If it returns success in all cases then Conjecture G is true. Otherwise it is false.

Algorithm 1: Verification that for every non-zero element a of $\mathbb{F}_p$ there exist two primitive roots also of $\mathbb{F}_p$, g_m and g_n, such that $a = g_n + cg_m$.

```
1  Set t_0 to 1, set t_1, t_2, ..., t_{p-1} to 0, and set r to p − 1
2  for m = 1, 2, ..., φ(p − 1) do
3      Set d to cg_m
4      for n = 1, 2, ..., φ(p − 1) do
5          Set a to g_n + d
6          if t_a is equal to 0 then set t_a to 1 and decrease r
7      if r is equal to 0 then terminate with success
8  Terminate with failure
```

If at some point during the execution of Algorithm 1 t_a is equal to 0 then no solution to $a = g_n + cg_m$ was found up to that point. Note that r counts the number of t_a's that are still equal to 0. For small values of m, Algorithm 1 is quite efficient at discarding values of a, but when r starts to become much smaller than $\phi(p-1)$ it becomes inefficient. Algorithm 2 handles small values of r better, but is less efficient than Algorithm 1 when r is large.

Algorithm 2: Verification that for every non-zero element a of $\mathbb{F}_p$ there exist two primitive roots also of $\mathbb{F}_p$, g_m and g_n, such that $a = g_n + cg_m$.

```
1  Set a_1 to 1, a_2 to 2, ..., a_{p-1} to p − 1, and set r to p − 1
2  for m = 1, 2, ..., φ(p − 1) do
3      Set i to 1 and set d to cg_m
4      while i ≤ r do
5          Set j to a_i − d
6          if j is a primitive root then
7              Set a_i to a_r and decrease r
8          else
9              Increase i
10     if r is equal to 0 then terminate with success
11 Terminate with failure
```

Note that $a_1, \ldots, a_r$ hold the r values of a for which no solution of $a = g_n + cg_m$ has yet been found. It is quite easy to switch from the first algorithm to the second and *vice versa* at the point where a new value of m is to be considered. We leave the easy details about how to do this for the reader to amuse herself/himself. In our implementation of these algorithms, we switch from the first to the second as soon as r drops below $0.25\,\phi(p-1)$. For our range of values of p, the hybrid algorithm has an execution time that is very nearly proportional to p.

The verification of Conjecture G for a given value of p can be done easily in parallel by assigning different ranges of values of c (a work unit) to each of the available processor cores. This was done for the 749 prime values of q that, according to Table 1 had to be tested. Using an Intel Core 2 Duo E8400 processor running at 3.0 GHz this took about 12.4 one-core days. In all cases it was found that $q \in \mathcal{G}$. A second run of the program, on an Intel Core i5-2400 processor running at 3.1 GHz, produced exactly the same results for each work unit, with the obvious exception of execution times, and required 13.4 one-core days. (This second run was slower due to data cache effects.) For each work unit we recorded the number of

times the hybrid algorithm terminated with a given value of m and we computed a 32-bit cyclic redundancy checksum that depended on the values of some variables at key points of the hybrid algorithm.

For our first run, it took approximately $1.3 \times 10^{-8}\, p^2$ seconds to test Conjecture G for a given value of p. For p above 10^3 our algorithm terminated with success for an average value of m that was close to $\log\bigl(\frac{1}{2p}\bigr) / \log\bigl(1 - \frac{\phi(p-1)}{p}\bigr)$.

To double-check the results of [1], we also ran our programs for all primes up to 2130. As expected, we found that Conjecture G is false only for $q = 3$, $q = 5$, $q = 7$, $q = 11$, $q = 13$, $q = 19$, $q = 31$, $q = 43$, and $q = 61$.

While the two runs of the program were underway, we found a way to share most of the work needed to test several values of c, thus giving rise to a much more efficient program. The key to this improvement is the observation that in (1) c appears multiplied by g_m. Thus, if instead of iterating on m on the outer loops of Algorithms 1 and 2 we iterate on carefully chosen values of the product cg_m, which we denote by d, then it becomes possible to exclude the same value of a simultaneously for several different c's, also carefully chosen.

To explain how this is done, let g be one primitive root of $\mathbb{F}_p$, let u be the largest non-repeated prime factor of $p-1$, and let $v = (p-1)/u$. The set $G = \{\, g^{1+iv} \,\}_{i=0}^{u-1}$ contains exactly $u-1$ primitive roots and exactly one non-primitive root, which will be denoted by z. Since g is a primitive root the sets $C_o = \{\, g^{o+iv} \,\}_{i=0}^{u-1}$, $0 \le o \le v-1$, are pairwise disjoint and their union is the set of the non-zero elements of $\mathbb{F}_p$. Moreover, the set formed by the products of one member of G and one member of C_o is C_{o+1} (note that $C_p = C_0$). Let C'_o be a non empty proper subset of C_o, and let C''_{o+1} be the corresponding subset of C_{o+1} whose members are obtained by multiplying the members of C'_o by the non-primitive root z. To test simultaneously the values of c belonging to C'_o, we use as values of d the complement of C''_{o+1}, i.e., the set $C_{o+1} - C''_{o+1}$. This ensures, for every $c \in C'_o$, that d is the product of c and a primitive root of G. These observations give rise to Algorithm 3.

Algorithm 3: Efficient verification that for $c \in C'_o$ and for every non-zero element a of $\mathbb{F}_p$ there exist two primitive roots also of $\mathbb{F}_p$, g_m and g_n, such that $a = g_n + cg_m$.

```
1  Set t_0 to 1, set t_1, t_2, ..., t_{p-1} to 0, and set r to p - 1
2  for d belonging to the complement of C''_{o+1} do
3  |  for n = 1, 2, ..., φ(p - 1) do
4  |  |  Set a to g_n + d
5  |  |  if t_a is equal to 0 then set t_a to 1 and decrease r
6  |  if r is equal to 0 then terminate with success
7  for c ∈ C'_o do
8  |  Run Algorithm 1 with a copy of the t_a and r variables, beginning it at line 2
9  |  Terminate with failure if Algorithm 1 failed
10 Terminate with success
```

To test Conjecture G for $c \in C_o$ it is obviously necessary to run Algorithm 3 twice: once for C'_o and once more with C'_o replaced by its complement. For that reason, to make the entire testing effort more efficient, C'_o and its complement should have approximately the same number of members (as u is usually odd, one should have one more member than the other). As before, the verification of Conjecture G for a given value of p can easily be done in parallel, now by assigning different ranges of values of o to each of the available processor cores. This was done for the 749 primes values of q that had to be tested. Taking only 2.7

one-core days plus 1.9 days for double-checking, this computation confirmed the results of our first two runs.

3.2 Verification of Conjecture G when $q = p^k$

We have chosen to represent a generic element a of $\mathbb{F}_q$ by the polynomial $\sum_{i=0}^{k-1} a_k x^k$ of formal degree $k-1$ with coefficients in $\mathbb{F}_p$, and henceforth to do multiplications in $\mathbb{F}_q$ using polynomial arithmetic modulo a monic irreducible polynomial of degree k. Since all finite fields with q elements are isomorphic to each other, any irreducible polynomial will do. Since we also need a primitive root, instead of finding first an irreducible polynomial and then finding a primitive root for that particular model of the finite field, we fix the primitive root (for convenience we have chosen $g = x$), and then find a monic polynomial of degree k for which $g^{q-1} = 1$ and for which $g^{(q-1)/f} \neq 1$ for each prime factor f of $q-1$. This ensures that the polynomial is indeed primitive and that g is one of its primitive roots.

Although the arithmetic operations are different when $q = p^k$, the main ideas of the three algorithms presented above remain valid. In all three algorithms it is necessary to replace p by q. In addition, in Algorithm 1 it is necessary to replace in line 6 t_a by $t_{a'}$, where a' is the value of the polynomial that represents a for $x = p$, because a was being used there as an index. Likewise for Algorithm 3 in line 5. In Algorithm 2 it is necessary to replace the way the a_j variables are initialised, since these are now polynomials with coefficients a_{ji}.

Using Algorithm 2, it took 7.0 days on a single core of a 2.8 GHz processor, plus 11.9 days to double-check the results on a slower processor, to check the conjecture for the 28 prime powers that had to be tested. In all cases it was found that $q \in \mathcal{G}$.

Finally, to double check the results of [1], we also ran our programs for all prime powers up to 2130. As expected, we found that Conjecture G is false only for $q = 4$.

A Proof of Theorem 2

Sieving methods for problems involving primitive roots have been refined since those described in [5] and [3] were formulated. A recent illustrative example occurs in [4] and is a model for the line of argument pursued here.

Throughout, suppose that a and b are arbitrary given non-zero members of $\mathbb{F}_q$. For any $g \in \mathbb{F}_q$ set $a - cg = g^*$.

Let e be a divisor of $q-1$. Call $g \in \mathbb{F}_q$ *e-free* if $g \neq 0$ and $g = h^d$, where $h \in \mathbb{F}_q$ and $d|e$, implies $d = 1$. The notion of e-free depends (among divisors of $q-1$) only on $\mathrm{Rad}(e)$. Moreover, in this terminology a primitive root of $\mathbb{F}_q$ is a $(q-1)$-free element. Next, given divisors e_1, e_2 of $q-1$, define $N(e_1, e_2)$ to be the number of $g \in \mathbb{F}_q$ such that g is e_1-free and g^* is e_2-free. In order to show that a prime power $q \in \mathcal{G}$ we have to show that $N(q-1, q-1)$ is positive (for every choice of a and c). The value of $N(e_1, e_2)$ can be expressed explicitly in terms of Jacobi sums over $\mathbb{F}_q$ as follows. We have

$$N(e_1, e_2) = \theta(e_1)\theta(e_2) \int_{d_1|e_1} \int_{d_2|e_2} \chi_{d_1}(1/c)(\chi_{d_1}\chi_{d_2})(a) J(\chi_{d_1}, \chi_{d_2}). \tag{3}$$

Here, for a divisor e of $q-1$,

$$\int_{d|e} = \sum_{d|e} \frac{\mu(d)}{\phi(d)} \sum_{\chi_d},$$

where the sum over χ_d is the sum over all $\phi(d)$ multiplicative characters χ_d of $\mathbb{F}_q$ of exact order d, and $J(\chi_{d_1}, \chi_{d_2})$ is the Jacobi sum $\sum_{g\in\mathbb{F}_q} \chi_{d_1}(g)\chi_{d_2}(1-g)$. (All multiplicative characters on $\mathbb{F}_q$ by convention take the value 0 at 0.)

Next, we present a combinatorial sieve. Let e be a divisor of $q-1$. In practice, this *kernel* e will be chosen such that $\mathrm{Rad}(e)$ is the product of the smallest primes in $q-1$. Use the notation of Theorem 2. In particular, if $\mathrm{Rad(e)} < \mathrm{Rad}(q-1)$ let $p_1, \dots, p_s$, $s \geq 1$, be the primes dividing $q-1$ but not e and set $\delta = 1 - \sum_{i=1}^{s} 2p_i^{-1}$. In practice, it is essential to choose e so that $\delta > 0$.

Lemma 1. *Suppose e is a divisor of $q-1$. Then, in the above notation,*

$$N(q-1, q-1) \geq \sum_{i=1}^{s} N(p_i e, e) + \sum_{i=1}^{s} N(e, p_i e) - (2s-1)N(e,e). \tag{4}$$

Hence

$$N(q-1,q-1) \geq \sum_{i=1}^{s} \{[N(p_i e, e) - \theta(p_i)N(e,e)] + [N(e, p_i e) - \theta(p_i)N(e,e)]\} + \delta N(e,e). \tag{5}$$

Proof. The various N terms on the right side of (4) can be regarded as counting functions on the set of $g \in \mathbb{F}_q$ for which both g and $g*$ are e-free. In particular, $N(e,e)$ counts all such elements, whereas, for example, $N(p_i e, e)$, $i \leq s$, counts only those for which additionally g is p_i-free. Since $N(q-1, q-1)$ is the number of e-free elements g for which g and g^* are both p_i-free for every $i \leq s$, we see that, for a given e-free $g \in \mathbb{F}_q$, the right side of (4) clocks up 1 if g and g^* are both primitive and otherwise contributes a non-positive (integral) quantity. This establishes (4). Since $\theta(p_i) = 1 - 1/p_i$, the bound (5) is deduced simply by rearranging the right side of (4). □

Lemma 2. *Suppose that $q \geq 4$ is a prime power and e is a divisor of $q-1$. Then*

$$N(e,e) \geq \theta(e)^2 (q - W(e)^2 \sqrt{q}). \tag{6}$$

Moreover, for any prime l dividing $q-1$ but not e, we have

$$|N(le, e) - \theta(l)N(e,e)| \leq (1 - 1/l)\theta(e)^2 W(e)^2 \sqrt{q}. \tag{7}$$

and

$$|N(e, le) - \theta(l)N(e,e)| \leq (1 - 1/l)\theta(e)^2 W(e)^2 \sqrt{q}. \tag{8}$$

Proof. Starting with the identity (3) we use the fact that when $d_1 = d_2 = 1$ (so that $\chi_{d_1} = \chi_{d_2}$ is the principal character of $\mathbb{F}_q$), then $\chi_{d_1}(1/c)(\chi_{d_1}\chi_{d_2})(a)J(\chi_{d_1}, \chi_{d_2}) = q-2$. For all other character pairs (χ_{d_1}, χ_{d_2}), as is well-known, this quantity has absolute value $\sqrt{q}$ (at most). Because there are, for example, $\phi(d_1)$ characters χ_{d_1} of order d_1, when we take into account the implicit denominators $\phi(d_1)$ and the Möbius function within the integral notation, we obtain as an aggregate contribution to the right side of (3) a quantity of absolute value at most $\sqrt{q}$ from each pair of *square-free* divisors d_1, d_2 of e, except the pair (1,1). Hence $N(e,e) \geq \theta(e)^2\{q - 2 - (W(e)^2 - 1)\sqrt{q}\}$, which yields (6).

Further, from (3), since $\theta(le) = \theta(l)\theta(e)$,

$$N(le, e) - \theta(l)N(e,e) = \theta(l)\theta(e)^2 \int_{d_1|e} \int_{d_2|e} \chi_{ld_1}(1/c)(\chi_{ld_1}\chi_{d_2})(a)J(\chi_{ld_1}, \chi_{d_2}).$$

Hence,

$$|N(le,e) - \theta(l)N(e,e)| \le \theta(l)\theta(e)^2 W(e)(W(le) - W(e))\sqrt{q},$$

which yields (7), since $W(le) = 2W(e)$. Similarly, (8) holds. □

We now complete the proof of Theorem 2.

Proof. Assume $\delta > 0$. From (5) and Lemma 2

$$\begin{aligned} N(q-1,q-1) &\ge \theta(e)^2 \left\{ \delta(q - W(e)^2\sqrt{q}) - \sum_{i=1}^{s} 2\left(1 - \frac{1}{p_i}\right) W(e)^2\sqrt{q} \right\} \\ &= \delta\theta(e)^2\sqrt{q}\left\{ \sqrt{q} - W(e)^2 - \left(\frac{2s-1}{\delta} + 1\right) W(e)^2 \right\}. \end{aligned}$$

The conclusion follows. □

References

[1] W.-S. Chou, G. L. Mullen, J.-S. Shiue, and Q. Sun, *Pairs of primitive elements modulo p^l*, Journal of Sichuan University Natural Science Edition **26** (1991), 189–195.

[2] S. D. Cohen, *Pairs of primitive roots*, Mathematika **32** (1985), 276–285.

[3] _____, *Primitive elements and polynomials: existence results*, Finite fields, coding theory and advances in communications and computing (New York), Lecture Notes in Pure and Appl. Math. 141, Dekker, 1993.

[4] S. D. Cohen and S. Huczynska, *The strong primitive normal basis theorem*, Acta Arith. **143** (2010), 299–332.

[5] S. D. Cohen and G. L. Mullen, *Primitive elements in finite fields and Costas arrays*, AAECC **2** (1991), 45–53.

[6] S. W. Golomb, *Algebraic constructions for Costas arrays*, J. Combin. Th. Ser. A. **37** (1984), 13–21.

[7] R. K. Guy, *Unsolved problems in number theory*, 3rd ed., Problem Books in Mathematics, Springer, 2004.

[8] Q. Sun, *On primitive roots in a finite field*, Sichuan Daxue Xuebao **25** (1988), no. 2, 133–139.

[9] M. Szalay, *On the distribution of the primitive roots of a prime*, J. Number Theory **7** (1975), 184–188.

[10] E. Vegh, *A note on the distribution of the primitive roots of a prime*, J. Number Theory **3** (1971), 13–18.