Computations on Normal Families of Primes

by

Erick Wong

B.Sc., Simon Fraser University, 1994

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the Department
of
Mathematics & Statistics

SIMON FRASER UNIVERSITY
August 1997

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

0-612-24272-2

APPROVAL

Name: Erick Wong

Degree: Master of Science

Title of thesis: Computations on Normal Families of Primes

Examining Committee: Dr. C. Schwarz
Chair

____
Dr. J. M. Borwein
Senior Supervisor

____
Dr. P./B. Borwein

____
Dr. M. Monagan

____
Dr. A. Gupta
Associate Professor
Computing Science Department

Date Approved: ________________________________

Abstract

We call a family of primes $\mathcal{P}$ *normal* if it contains no two primes p, q such that p divides $q - 1$. In this thesis we study two conjectures and their related variants.

Giuga's conjecture is that $\sum_{k=1}^{n-1} k^{n-1} \equiv n - 1 \pmod{n}$ implies n is prime. We study a group of eight variants of this equation and derive necessary and sufficient conditions for which they hold.

Lehmer's conjecture is that $\phi(n) \mid n - 1$ if and only if n is prime. This conjecture has been verified for up to 13 prime factors of n, and we extend this to 14 prime factors. We also examine the related condition $\phi(n) \mid n + 1$ which is known to have solutions with up to 6 prime factors and extend the search to 7 prime factors.

For both of these conjectures the set of prime factors of any counterexample n is a normal family, and we exploit this property in our computations.

Dedication

I dedicate this thesis in memory of Kaz Shinyashiki, my best friend of many years. *To Kaz: you're in the books for sure now...*

Acknowledgements

First, I thank my supervisor Jon Borwein for his patience and constructive comments throughout the production of this thesis.

I would also like to thank my friends in the Math Department at SFU, especially Kathy, Brian, Harvey and Luis, for their helpful advice and guidance over the past eight years.

I thank my family for their constant support throughout my studies, my mother for fostering my early interest in mathematics, and my father for driving me to school every day for so many years.

Many thanks go to my good friend Jen Chang for her encouragement and assistance in writing Chapter 3.

Finally, a very special thank-you goes to Janice Fung, who has been a dear friend through some hard times, and also helped to proofread the early drafts of Chapter 2. I would never have completed this work without her continuing support from beginning to end.

Contents

List of Tables

Chapter 1

Introduction

1.1 Introduction

The fields of number theory and computing science have enjoyed a fruitful partnership over the last few decades. Studies of such number-theoretic problems as primality testing, factoring, and discrete logarithms have helped make practical public-key cryptography a reality. Conversely, the computer has become an invaluable tool for dealing with large numbers, making once impossibly long calculations feasible, and once tedious work trivial.

There are many conjectures which have simple and elegant statements, but for which there seems to be little hope of a complete resolution. This is particularly true in the field of number theory (for example, the twin prime conjecture, the existence of odd perfect numbers, and until quite recently Fermat's conjecture). In the absence of a proof or disproof, one method of investigation is to derive a lower bound on the size of a counterexample. G. Giuga and D. H. Lehmer each studied such problems and obtained computational bounds for them through hand calculations and tables, well before the advent of modern computing.

As computer speeds increase, more problems become tractable, and better bounds can be obtained. In this thesis we look at both Giuga's and Lehmer's conjectures as well as some variants, and see what new results we can derive with the level of computational power available today. In Chapter 2 we study Giuga's conjecture

$$\boxed{\sum_{k=1}^{n-1} k^{n-1} \equiv n-1 \pmod{n} \quad \text{iff } n \text{ is prime,}}$$

together with several related variants. The bulk of this work is contained in (**BW**). In Chapter 3 we examine Lehmer's conjecture

$$\boxed{\phi(n) \mid n-1 \quad \text{iff } n \text{ is prime,}}$$

and obtain a new bound on the number of prime factors of a counterexample. We also partially extend Lehmer's work of determining which n satisfy $\phi(n) \mid n+1$. We will find that these problems all share a common structure which can be exploited to make the computations somewhat easier.

1.2 Normal Families

We call a finite family of distinct primes $\mathcal{P} = \{p_1, p_2, p_3, \ldots, p_k\}$ *normal* if $p_i \not| \, p_j - 1$ for all $1 \leq i, j \leq k$. It is clear that if $k > 1$ then $2 \notin \mathcal{P}$.

We say that a prime q is *normal to* an existing normal family $\mathcal{P}$ if $q > \max \mathcal{P}$ and $\mathcal{P} \cup \{q\}$ is also normal.

Example: $\{3, 5, 23, 29\}$ is a normal family of primes. The smallest prime normal to this family is 53.

In general, we denote by $\lceil \mathcal{P} \rceil$ the first prime normal to $\mathcal{P}$, and by $\lceil \mathcal{P}, q \rceil$ the first prime greater than or equal to q which is normal to $\mathcal{P}$. Except for the trivial case $\mathcal{P} = \{2\}$, Dirichlet's theorem (**Leve**) guarantees that such a prime must exist. Similarly, $\lfloor \mathcal{P}, q \rfloor$ denotes the largest prime less than or equal to q which is normal to $\mathcal{P}$ (provided $q \geq \lceil \mathcal{P} \rceil$).

It turns out that for both Giuga's and Lehmer's conjectures as well as many of the variants we consider, a necessary condition for n to be a counterexample is that the prime factors of n form a normal family. By examining only normal families of a given cardinality, we reduce the search space required to verify these conjectures for a given number of prime factors. Incidentally, these n have an interesting group-theoretic property (thanks to Dr. Derek Holt at the University of Warwick for this observation).

Fact. Let n be a product of primes from a normal family. Then there is a unique group of order n up to isomorphism (namely $\mathbf{Z}_n$). Conversely, if n is not squarefree or if n is divisible by primes p, q where $p \mid q-1$, then there exists a group of order n which is not cyclic.

Sketch. Let G be a group of order n. If P is a Sylow p-subgroup of G, then $|\mathrm{Aut}(P)| = p-1$ does not divide $|G|$. We then apply Burnside's Transfer Theorem (see **Hall**) to show that G has a normal subgroup N of order n/p such that $G = NP$. Choosing P to be a normal Sylow p-subgroup of G, we see that $G \cong N \times P$, and an induction on the number of prime factors of n completes the argument.

To prove the converse, it is sufficient to simply find non-cyclic groups of orders p^2 and pq. The former is easily given by $\mathbf{Z}_p \times \mathbf{Z}_p$. For the latter, let b be a primitive p-th root of unity modulo q. Then the group with presentation

$$\langle x, y \mid x^p = y^q = 1, xy = y^b x \rangle$$

is non-Abelian of order pq. ☺

Before proceeding to study the two conjectures in detail, we introduce the following concepts which will be useful.

A *normal sequence* is an infinite sequence $\mathcal{P} = \{p_1 < p_2 < p_3 < \cdots\}$ of primes where $p_i \not| \, p_j - 1$ for all $i, j \geq 1$. It is easy to construct such a sequence inductively from any prescribed finite subset $\mathcal{P}'$ by repeatedly adjoining $\lceil \mathcal{P}' \rceil$. We define a *restriction* $\mathcal{P}[k]$ of $\mathcal{P}$ to be the elements of $\mathcal{P}$ which are less than or equal to k. We denote by $\mathcal{P}_k$ the simple restriction consisting of $\{p_1, p_2, \ldots, p_k\}$.

Given two families of primes $\mathcal{A} = \{a_1 < a_2 < \cdots < a_k\}$ and $\mathcal{B} = \{b_1 < b_2 < \cdots < b_l\}$ we say that $\mathcal{A}$ *dominates* $\mathcal{B}$ if $k \geq l$ and $a_i \leq b_i$ for each $1 \leq i \leq l$. This definition easily extends to an infinite sequence $\mathcal{A}$ (and a finite or infinite $\mathcal{B}$), and a natural interpretation is that the primes of $\mathcal{A}$ are more densely distributed than those of $\mathcal{B}$. We make the following observation.

Proposition. Let $k \geq 3$. Then any normal family (or normal sequence) $\mathcal{P}$ is dominated by a sequence of the form $\mathcal{Q} \cup \{a_0, a_1, a_2, \ldots\}$, where $\mathcal{Q}$ is a (possibly empty) normal family with no primes greater than k, $a_0 = \lceil \mathcal{Q}, k+1 \rceil$, and $a_{i+1} = \lceil \mathcal{Q}, a_i + 1 \rceil$.

Proof. Take $\mathcal{Q} = \mathcal{P}[k]$. ☺

By considering all possible normal families with elements less than a given k, we can universally bound any normal family. We can assume without loss of generality that k is prime.

Example: When $k = 7$, the possible normal families are $\{\}, \{3\}, \{5\}, \{7\}, \{3,5\}$ and $\{5,7\}$. Thus every normal family is dominated by one of the following sequences:

- $\{11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, \ldots\}$,
- $\{\mathbf{3}, 11, 17, 23, 29, 41, 47, 53, 59, 71, 83, 89, 101, \ldots\}$,
- $\{\mathbf{5}, 13, 17, 19, 23, 29, 37, 43, 47, 53, 59, 67, 73, \ldots\}$,
- $\{\mathbf{7}, 11, 13, 17, 19, 23, 31, 37, 41, 47, 53, 59, 61, \ldots\}$,
- $\{\mathbf{3}, \mathbf{5}, 17, 23, 29, 47, 53, 59, 83, 89, 107, 113, 137, \ldots\}$,
- $\{\mathbf{5}, \mathbf{7}, 13, 17, 19, 23, 37, 47, 53, 59, 67, 73, 79, \ldots\}$.

Chapter 2

Giuga's Family

2.1 Giuga's Conjecture

In 1950, G. Giuga conjectured that if an integer $n > 1$ satisfies

$$\sum_{k=1}^{n-1} k^{n-1} \equiv n-1 \pmod{n}, \tag{2.1}$$

then n is a prime.

Clearly the converse holds: if n is a prime, then $k^{n-1} \equiv 1 \pmod{n}$ for $1 \le k \le n-1$ by Fermat's little theorem, thus

$$\sum_{k=1}^{n-1} k^{n-1} \equiv \sum_{k=1}^{n-1} 1 \equiv n-1 \pmod{n}.$$

Giuga (**Giu**) showed that a composite n satisfies this condition if and only if $p \mid (n/p) - 1$ and $p - 1 \mid n - 1$ for each prime p dividing n. Using this criterion, he verified the conjecture computationally for all integers up to 1000 digits. E. Bedocchi (**Bed**) later improved this to 1700 digits. With greater computational resources and a significant refinement of the technique, D. Borwein, J. M. Borwein, P. B. Borwein, and R. Girgensohn (**B^{3}G**) have recently verified the conjecture for up to 13887 digits. In this chapter we study eight natural variations of condition 2.1 and derive for each of them similar characterizations in terms of divisibility.

2.2 The Eightfold Way

Recall that condition 2.1,

$$\sum_{k=1}^{n-1} k^{n-1} \equiv n-1 \pmod{n},$$

holds whenever n is a prime. If we alter this condition by changing the exponent $n-1$ to $\phi(n)$ (the Euler phi function), it will not have any effect for prime n. Similarly we can change the right-hand side $n-1$ to $\phi(n)$. Additionally, rather than summing with k between 1 and $n-1$, we can take the sum over only the $\phi(n)$ integers relatively prime to n.

By combining these alternatives, we get eight (2^3) different variations on Giuga's condition, all of which hold for prime n. The question remains: *for which composite n do each of these conditions hold?* Before we proceed to analyze these cases, let us introduce some notation.

We use $[n]$ to denote the set of integers between 1 and n (note that the inclusion of n does not affect any sum modulo n), and $[n]^*$ for the subset of $[n]$ consisting of integers relatively prime to n. Recall that $[n]^*$ forms a group under multiplication modulo n.

Example: $[15]^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$

We will make the convention that p is a prime, and $q = p^r$ is a prime power, with $r \geq 1$. Given $n \geq 2$ we will denote its prime factorization $p_1^{r_1} p_2^{r_2} \cdots p_l^{r_l}$, where $p_1 < p_2 < \cdots < p_l$, and define $q_i = p_i^{r_i}$. We will use the notation $O(n)$ to represent an arbitrary integer divisible by n, in a fashion similar to the "big-oh" notation of complexity theory. This notation is particularly useful when n is a prime power, for which we note the following.

Observations:

- if $r \leq s$, $O(p^r) + O(p^s) = O(p^r)$
- $O(p^r) \cdot O(p^s) = p^r \cdot O(p^s) = O(p^{r+s})$

- $O(p^1) \supset O(p^2) \supset O(p^3) \supset \cdots$

We will use $S(n,m)$ to denote the sum $\sum_{k\in[n]} k^m$, and $S^*(n,m)$ for the sum $\sum_{k\in[n]^*} k^m$. We call $S(n,m)$ a *sum of Type I* and $S^*(n,m)$ a *sum of Type II*. In the next section we prove a theorem that is useful for dealing with Type I and Type II sums modulo n.

2.3 Evaluating sums

We start off with a simple lemma:

Lemma. If $a \equiv 1 \pmod{p}$, then $a^{p^i} \equiv 1 \pmod{p^{i+1}}$, for all $i \geq 0$.

Proof. If $a = 1 + O(p^j)$ for some j, then the binomial expansion gives

$$a^p = 1 + p \cdot O(p^j) + O(p^{2j}) = 1 + O(p^{j+1}),$$

and the result follows by induction. ☺

The multiplicative structure of the group $[q]^*$, where $q = p^r$, is well-known (see, for example, **NZM**).

Fact.

- If p is odd, $[q]^*$ is cyclic with some generator α_q of order $\phi(q) = p^{r-1}(p-1)$.
- If $r \geq 2$, $[2^r]^*$ is isomorphic to $\mathbf{Z}_{2^{r-2}} \times \mathbf{Z}_2$, and is generated by the following elements: 5 which has order 2^{r-2} and -1 which has order 2.
- $[2]^*$ is trivial.

Let $[q]^*_m$ denote the set of mth powers (modulo q) of the elements of $[q]^*$. If m and $\phi(q)$ are relatively prime, then this set is identical to $[q]^*$. Furthermore, we note that given any $m > 0$, we can find m' dividing $\phi(q)$ such that $[q]^*_m = [q]^*_{m'}$, namely by taking $m' = \gcd(m, \phi(q))$. If $q = 2^r$ and $r \geq 2$, then we can replace $\phi(q)$ in this expression with $2^{r-2} = \frac{1}{2}\phi(q)$.

Examples:

- $[25]^*_{15} = \{1, 18, 7, 24\} = [25]^*_5$,
- $[25]^*_{12} = \{1, 11, 16, 21\} = [25]^*_4$,
- $[16]^*_4 = [16]^*_8 = \{1\}$.

With this in mind, we now prove the following theorem concerning $S^*(n, m)$ when n is a prime power.

Theorem 2.1 *If $q = p^r$ is a prime power, then $S = S^*(q, m) = \sum_{k\in[q]^*} k^m$ is congruent to either 0 or $\phi(q)$ modulo q. Furthermore, the cases where $S \equiv \phi(q)$ (mod q) are characterized as follows.*

Case 1 If $p \geq 3$, then $S^*(q, m) = \phi(q)$ iff $p - 1 \mid m$.

Case 2 If $q = 2^r$ and $r \geq 2$, then $S^*(q, m) = \phi(q)$ iff m is even.

Case 3 If $q = 2$, then $S = \phi(2) = 1$.

Proof. We consider the first two cases individually. The third case is trivial.

Case 1: We have

$$S = \sum_{k=0}^{\phi(q)-1} \left(\alpha_q^k\right)^m = \sum_{k=0}^{\phi(q)-1} \beta^k, \tag{2.2}$$

where α_q is a primitive root modulo p^r, and $\beta = \alpha_q^m$. The condition $p - 1 \not| \, m$ is equivalent to $\beta - 1$ being non-zero modulo p and hence invertible modulo q. Thus if $p - 1 \not| \, m$ then S is a geometric series which evaluates to

$$\frac{\beta^{\phi(q)} - 1}{\beta - 1} = 0.$$

Now suppose $p - 1 \mid m$. Then we may assume without loss of generality that $m = (p - 1)p^{s-1}$, where $1 \leq s \leq r$, and thus $\beta \equiv 1$ (mod p^s) by the lemma. Since the order of β is $\phi(q)/m = p^{r-s}$, we see that the sum (2.2) consists of exactly m repetitions of the sum

$$T = \sum_{k=0}^{p^{r-s}-1} \beta^k.$$

Since α_p is primitive, the summands of T are distinct modulo q and are all congruent to 1 modulo p^s. Thus they must form a permutation of the arithmetic progression $1 + n \cdot p^s$, $0 \leq n < p^{r-s}$. The sum of the terms in this progression is

$$p^{r-s} + p^s p^{r-s}(p^{r-s} - 1)/2 = p^{r-s} + O(p^r),$$

which gives us $T = p^{r-s}$, hence $S = mT = (p-1)p^{r-1} = \phi(q)$.

Case 2: For odd m, this is simple: the terms of the sum $\sum_{k\in[2^r]^*} k^m$ cancel out in pairs, thus $S(q,m) = 0$. We may therefore assume $m = 2^s$ with $1 \leq s \leq r-2$. The sum S factors into

$$S = (1 + (-1)^m) \sum_{k=0}^{(q/4)-1} \beta^k,$$

where $\beta = 5^m$. Since m is even, the first factor is 2. The second factor consists of exactly m repetitions of the sum

$$T = \sum_{k=0}^{(q/4m)-1} \beta^k.$$

The summands of T are distinct modulo q, and congruent to 1 modulo $4m$, thus they form a permutation of the arithmetic progression $1 + 4mn$, $0 \leq n < q/4m$, which sums to

$$q/4m + 4m(q/4m)(q/4m - 1)/2 = q/4m + O(q/2).$$

Multiplication by $2m$ yields $S = 2mT = q/2 = \phi(q)$. ☺

Note in particular that $S^*(p^r, m)$ is always divisible by p^{r-1}, regardless of m. This observation allows us to prove that the results of Theorem 2.1 also hold for most sums of the form $\sum_{k\in[p^r]} k^m$. More specifically, we have the following theorem:

Theorem 2.2 *If $m > 1$, then $S(p^r, m) \equiv S^*(p^r, m)$ (mod p^r). When p is odd, the congruence holds for $m \geq 1$.*

Proof. This is trivial for $m > r$ (and hence $r = 1$), which is really enough for the following sections, but the full result has a nice proof by induction: we have, for $r > 1$, $S(p^r, m) = S^*(p^r, m) + p^m S(p^{r-1}, m)$. The last term is divisible by $p^m(p^{r-2})$ and thus 0 modulo p^r.

When $m = 1$, a straightforward calculation gives

$$S(p^r, 1) - S^*(p^r, 1) = \sum_{k=1}^{p^{r-1}} pk = \frac{1}{2}p^r p^{r-1} + 1,$$

which is $O(p^r)$ if p is odd. ☺

2.4 Classification

In this section we consider the general case of $S(n, m)$ and $S^*(n, m)$ when n is any integer greater than 1. We first consider the four sums of Type I, including Giuga's original sum.

2.4.1 Sums of Type I

Recall that n has a prime factorization given by $\prod_{i=1}^{l} p_i^{r_i} = \prod_{i=1}^{l} q_i$. For a given i $(1 \leq i \leq l)$, the sum $\sum_{k \in [n]} k^m$ modulo q_i simply consists of n/q_i repetitions of $\sum_{k \in [q_i]} k^m$. Applying Theorem 2.1 to this latter sum gives us

$$S(n, m) \equiv 0 \text{ or } (n/q_i)\phi(q_i) \pmod{q_i}.$$

We first consider the case of the right-hand side $n - 1$. Now, $S(n, m)$ equals $n - 1$ modulo n iff it equals -1 modulo each of the q_i. This gives us the following theorem.

Theorem 2.3 $\sum_{k \in [n]} k^m \equiv n - 1$ *(mod n) if and only if n is squarefree, and for all odd primes p dividing n,* $p - 1 \mid m$ *and* $p \mid (n/p - 1)$.

Proof. First we show that n must be squarefree. Suppose $r_i \geq 2$ for some i, so that $p_i^2 \mid n$. Then $p_i \mid \phi(q_i)$, and hence $S(n, m) \equiv 0 \not\equiv -1 \pmod{q_i}$.

By similar logic, we need $\sum_{k\in[p]} k^m$ to be non-zero modulo p, hence the condition that $p-1 \mid m$ for odd p, from Theorem 2.1. If $p=2$, we have the third case of Theorem 2.1, and the congruence holds trivially.

We now must satisfy $S(n,m) \equiv (n/p)\phi(p) \equiv n-1 \pmod p$. Dividing through by $\phi(p) \equiv -1 \pmod p$ we see this is equivalent to $n/p \equiv 1 \pmod p$, giving us the third condition.

It is clear, from the proof, that these conditions are sufficient. ☺

The next two corollaries follow immediately from Theorem 2.3.

Corollary 2.1 *$S(n,\phi(n)) \equiv n-1$ (mod n) iff n is squarefree, and for each prime p dividing n, $p \mid (n/p-1)$.*

Corollary 2.2 (Giuga) *$S(n,n-1) \equiv n-1$ (mod n) iff n is squarefree, and for each prime p dividing n, $p \mid (n/p-1)$ and $p-1 \mid n-1$.*

A natural number n which satisfies the conditions of Corollary 2.1 is called a *Giuga number*. These numbers are studied in (**B^{3}G**) as well as in (**BrJ**), and we look at them briefly in the next section.

We now consider the case $S(n,m) \equiv \phi(n) \pmod n$. This will hold if and only if for each i, $(n/q_i)S(q_i,m) \equiv \phi(n) \pmod{q_i}$ By examining the two possibilities for the left-hand side of this congruence, we arrive at the following theorem.

Theorem 2.4 *$\sum_{k\in[n]} k^m \equiv \phi(n)$ (mod n) if and only if, for each i, $1 \le i \le l$, one of the following conditions holds:*

- *$S(q_i,m) \equiv 0$ (mod q_i) and $p_i \mid p'-1$ for some prime p' dividing n; or*
- *$S(q_i,m) \equiv \phi(q_i)$ (mod q_i) and $n/q_i \equiv \phi(n/q_i)$ (mod p_i).*

Proof. For the first case, we need $\phi(q_i)\phi(n/q_i) \equiv 0 \pmod{q_i}$. This is equivalent to $p_i \mid \phi(n/q_i)$, giving us the first condition.

For the second case, we must have $(n/q_i)\phi(q_i) \equiv \phi(n) \equiv \phi(q_i)\phi(n/q_i) \pmod{q_i}$. Since $\phi(q_i)$ is divisible by $p_i^{r_i-1}$ but not $p_i^{r_i}$, we can divide through by $\phi(q_i)$ to get $(n/q_i) \equiv \phi(n/q_i) \pmod{p_i}$. ☺

Corollary 2.3 *$S(n,\phi(n)) \equiv \phi(n)$ (mod n) iff for all primes p dividing n, $n/q \equiv \phi(n/q)$ (mod p), where $q = p^r$ is the highest power of p which divides n.*

We call n a *co-Giuga number* if it satisfies the conditions of Corollary 2.3. Note that co-Giuga numbers need not be squarefree, while Giuga numbers are. Also, the prime factors of a co-Giuga number n must form a normal family; otherwise $\phi(n/q) \equiv 0 \not\equiv n/q \pmod{p}$ for some $p \mid n$.

Corollary 2.4 *$S(n,n-1) \equiv \phi(n)$ (mod n) iff $n = 2$ or n is odd, co-Giuga, and $p - 1 \mid n - 1$ for all primes p dividing n.*

Proof. We will use a descent argument to rule out the first condition in Theorem 2.4. Suppose that for some p_i, $p_i \not| \, n-1$ so $S(q_i, n-1) \equiv 0 \pmod{q_i}$. We can choose such a p_i to be maximal. Then there must exist p' dividing n such that $p_i \mid p' - 1$. But clearly $p_i \not| \, n - 1$ and thus $p' - 1 \not| \, n - 1$. Since $p' > p_i$, this is a contradiction.

Now if $n > 2$ is even, then it must be a power of two, for no odd prime p can satisfy $p-1 \mid n-1$. Then we have $S(n,n-1) \equiv 0 \pmod{n}$ by Case 2 of Theorem 2.1, and the condition does not hold. ☺

The condition $p - 1 \mid n - 1$ stated above is identical to the well-known characterization of Carmichael numbers (**AGP**), with the exception that n in our case is not required to be squarefree. Relaxing this requirement results in a class of numbers which we call *pseudo-Carmichael numbers.* We will return to these numbers as well as the co-Giuga numbers in the next section.

2.4.2 Sums of Type II

We now consider the sums $S^*(n,m) = \sum_{k\in[n]^*} k^m$ of Type II. In order to do so, we need some information about the structure of the group $[n]^*$: it is isomorphic to the

Cartesian product $[q_1]^* \times [q_2]^* \cdots \times [q_l]^*$. Furthermore, we have a natural representation as follows.

For a given n, define $[n]^*_i$ to be the subset of $[n]^*$ consisting of all elements congruent to 1 modulo n/q_i. Then $[n]^*_i$ has cardinality $\phi(q_i)$, and in fact contains a unique representative modulo q_i for each element of $[q_i]^*$. Applying the Chinese remainder theorem we see that we can factor $[n]^*$ in the following sense.

Observation. Every $x \in [n]^*$ can be written uniquely as a product $x \equiv x_1 x_2 \cdots x_l$ modulo n, with each $x_i \in [n]^*_i$.

Proof. $\prod_{i=1}^{l} x_i \equiv x_i \pmod{q_i}$, so x_i must be the unique element of $[n]^*_i$ which is congruent to x modulo q_i.

This representation allows us to factor $S = S^*(n,m)$ into $S_1 S_2 \cdots S_l$, where

$$S_i = \sum_{k \in [n]^*_i} k^m.$$

Now, S_i is clearly congruent to $\phi(q_j)$ modulo q_j for $j \neq i$; thus we are only interested in S_i, which is congruent to $\sum_{k \in [q_i]^*} k^m$ modulo q_i. By Theorem 2.1 we know that $S_i \equiv 0$ or $\phi(q_i) \pmod{q_i}$. Therefore, $S = \prod S_i$ is congruent to either 0, or to

$$\phi(q_i) \prod_{i \neq j} \phi(q_j) = \phi(n) \pmod{q_i}.$$

We are now in a position to characterize sums of Type II in the manner of the previous subsection.

Theorem 2.5 $S = \sum_{k \in [n]^*} k^m \equiv n-1$ *(mod n) if and only if* n *is prime and* $n-1 \mid m$.

Proof. Since $n-1$ is relatively prime to n, we cannot have $S_i \equiv 0 \pmod{q_i}$ for any i. Thus $S_i \equiv \phi(q_i) \pmod{q_i}$ for all i, so that $S = \phi(n)$. But $\phi(n) < n-1$ when n is not prime. The result now follows from Theorem 2.1. ☺

The next two corollaries are immediate, but we include them for completion.

Corollary 2.5 $S^*(n, \phi(n)) \equiv n-1$ *(mod n) iff* n *is prime.*

Corollary 2.6 *$S^*(n, n-1) \equiv n - 1$ (mod n) iff n is prime.*

Theorem 2.6 *$S = \sum_{k\in[n]^*} k^m \equiv \phi(n)$ (mod n) if and only if, for each i, $1 \leq i \leq l$, either $S^*(q_i, m) \equiv \phi(q_i)$ (mod q_i), or $p_i \mid p' - 1$ for some prime p' dividing n.*

Proof. As in the proof of Theorem 2.4, if $S^*(q_i, m) \equiv 0 \pmod{q_i}$, then we need $\phi(n) \equiv 0 \pmod{q_i}$, and this is equivalent to the p' condition. If $S^*(q_i, m) \equiv \phi(q_i) \pmod{q_i}$, then we have $S^*(n, m) \equiv \phi(n) \pmod{q}$ by the arguments preceding Theorem 2.5. ☺

These final two corollaries complete the Eightfold way. Corollary 2.7 follows immediately from Euler's theorem that $a^{\phi(n)} \equiv 1 \pmod{n}$ for $a \in [n]^*$.

Corollary 2.7 *$S^*(n, \phi(n)) \equiv \phi(n)$ (mod n) for all natural numbers n.*

Corollary 2.8 *$S^*(n, n-1) \equiv \phi(n)$ (mod n) iff $n = 2$ or n is odd and $p - 1 \mid n - 1$ for each prime $p \mid n$.*

Proof. The descent argument of Corollary 2.4 applies here to prove that n must be a pseudo-Carmichael number. Again, as in Corollary 2.4, the condition does not hold when n is a power of two, so n cannot be even. Finally, Theorem 2.6 shows that the condition is sufficient. ☺

2.5 Giuga numbers et al

2.5.1 Giuga numbers

Giuga proved a nice alternative characterization of Giuga numbers in terms of the sum of reciprocals of the prime factors of the integer n. Suppose, for instance, that $n = abc$ is a Giuga number with 3 prime factors. Then we have $a \mid bc - 1, b \mid ac - 1, c \mid ab - 1$. Multiplying these together gives

$$(bc-1)(ac-1)(ab-1) \equiv bc + ac + ab - 1 + O(abc) \equiv 0 \pmod{abc},$$

which is equivalent to $1/a + 1/b + 1/c - 1/abc \in \mathbf{Z}$. The converse also holds: if $abc \mid bc + ac + ab - 1$ then $a \mid bc - 1$ by reduction modulo a, and similarly for b and c.

This argument easily generalizes to the following theorem.

Theorem 2.7 (Giuga) *n is a Giuga number if and only if n is squarefree and*

$$\sum_{p|n} 1/p - \prod_{p|n} 1/p \in \mathbf{Z}.$$

Corollary 2.9 *n is a Giuga number if and only if*

$$\sum_{p|n} 1/p - 1/n \in \mathbf{Z}.$$

Proof. The denominator of the summation is squarefree, so if the condition holds then n must be squarefree. ☺

Example: $30 = 2 \cdot 3 \cdot 5$ is a Giuga number, as $\frac{1}{2} + \frac{1}{3} + \frac{1}{5} - \frac{1}{30} = 1$. Other Giuga numbers include $858 = 2 \cdot 3 \cdot 11 \cdot 13$ and $1722 = 2 \cdot 3 \cdot 7 \cdot 41$.

2.5.2 Co-Giuga numbers

It turns out that co-Giuga numbers have a similar characterization to the one above for Giuga numbers. Since the co-Giuga condition is independent of the exponents r_i in the prime factorization of n, we can restrict our attention to the case where n is squarefree. Let us again consider the 3-prime case and take $n = abc$ to be a co-Giuga number. Then we have

- $(b-1)(c-1) \equiv bc \pmod{a}$,
- $(a-1)(c-1) \equiv ac \pmod{b}$,
- $(a-1)(b-1) \equiv ab \pmod{c}$.

Multiplying the first equation by $a-1$, the second by $b-1$ and the third by $c-1$, and then applying the Chinese remainder theorem, we obtain

$$(a-1)(b-1)(c-1) \equiv -bc - ac - ab \pmod{abc}, \tag{2.3}$$

which is equivalent to $1/a + 1/b + 1/c + (1-1/a)(1-1/b)(1-1/c) \in \mathbf{Z}$.

We can write this more symmetrically as

$$\left(1-\frac{1}{a}\right)\left(1-\frac{1}{b}\right)\left(1-\frac{1}{c}\right) - \left(1-\frac{1}{a}\right) - \left(1-\frac{1}{b}\right) - \left(1-\frac{1}{c}\right) \in \mathbf{Z}.$$

Conversely, examining equation (2.3) modulo a we see that it implies $-(b-1)(c-1) \equiv -bc \pmod{a}$ (similarly with respect to b and c), so the above condition is also sufficient.

This argument also generalizes to the following theorem.

Theorem 2.8 *n is a co-Giuga number if and only if*

$$\prod_{p|n}\left(1-\frac{1}{p}\right) - \sum_{p|n}\left(1-\frac{1}{p}\right) \in \mathbf{Z}.$$

We note that every prime power is a co-Giuga number, but we know of no other examples. Indeed, we can use the above characterization to obtain a lower bound on the number of prime factors of a non-trivial (not a prime power) co-Giuga number.

Theorem 2.9 *There are no non-trivial co-Giuga numbers with fewer than 7695 prime factors.*

Proof. Let $\mathcal{P} = \{p_1 < p_1 < p_2 < \cdots\}$ be a normal sequence of primes. We consider the expression

$$d(\mathcal{P}, l) = \prod_{i=1}^{l}\left(1-\frac{1}{p_i}\right) + \sum_{i=1}^{l}\frac{1}{p_i} - 2, \tag{2.4}$$

where $l \geq 1$. We see immediately from Theorem 2.8 that $\prod_{i=1}^{l} p_i$ is a co-Giuga number iff $d(\mathcal{P}, l) \in \mathbf{Z}$. This quantity is equal to -1 when $l = 1$, and strictly increases as

$l \to \infty$. Furthermore, it is also monotonic in the variables p_i. Thus if $\mathcal{Q}$ dominates $\mathcal{P}$, and $m \leq l$, then $d(\mathcal{Q}, l) \geq d(\mathcal{P}, m)$.

Recalling the proposition of Chapter 1 with $k = 3$, $\mathcal{P}$ is dominated by one of the two sequences:

- $\mathcal{A} = \{5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, \ldots\}$,
- $\mathcal{B} = \{\mathbf{3}, 5, 11, 17, 23, 29, 41, 47, 53, 59, 71, 83, 89, 101, \ldots\}$.

But, $d(\mathcal{A}, 7694) = -0.0000944$ and $d(\mathcal{B}, 7694) = -0.4071613\ldots$, thus $d(\mathcal{P}, m)$ is still strictly negative for $m < 7695$. At the same time it is strictly greater than -1, and thus cannot be an integer. ☺

2.5.3 Pseudo-Carmichael numbers

The pseudo-Carmichael numbers include all prime powers and Carmichael numbers, as well as many other numbers. The first five of these "others" are 45, 225, 325, 405, and 637.

There are infinitely many such examples, for instance $3^{2^r}5^s$ where $r, s \geq 1$. More generally, given any two primes $p < q$ with $p \nmid q - 1$, $p^{r\phi(q-1)}q^{s\phi(p-1)}$ is a pseudo-Carmichael number. A similar pattern exists for any normal family of primes.

Proposition. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_l\}$ be a normal family of primes, and define $r_i = \mathrm{lcm}_{j \neq i}\phi(p_j)$. Then any number of the form $p_1^{k_1r_1}p_2^{k_2r_2}\cdots p_l^{k_lr_l}$, where $k_i \geq 1$, is a pseudo-Carmichael number. Conversely, if n is a pseudo-Carmichael number then the prime factors of n form a normal family.

Proof. Since $\phi(p_j) \mid r_i$ for $j \neq i$, we have, by Euler's theorem, $p_i^{k_ir_i} \equiv 1 \pmod{p_j - 1}$. This congruence also holds trivially when $j = i$, thus $n = \prod_{i=1}^{l} p_i^{r_ik_i} \equiv 1 \pmod{p_j - 1}$ for each j. The converse is trivial: if primes p, q divide n and $p \mid q - 1$ then $p \mid n - 1$, a contradiction since $p \mid n$. ☺

An alternative characterization of pseudo-Carmichael numbers comes from considering $n' = \prod_{p|n} p$, the squarefree portion of n. It is easy to see that $p - 1 \mid n - 1$

Table 2.1: Conditions for $\sum_{k \in I} k^m \equiv r \pmod{n}$

I	r	m	Conditions on n	Trivial cases	Non-trivial examples
$[n]$	$n-1$	$\phi(n)$	Giuga	Primes	$30, 858, 1722, \ldots$
$[n]$	$n-1$	$n-1$	Giuga & Carmichael	Primes	$>$ 13800 digits
$[n]$	$\phi(n)$	$\phi(n)$	Co-Giuga	Prime powers	$>$ 7694 prime factors
$[n]$	$\phi(n)$	$n-1$	Odd, Co-Giuga & Pseudo-Carmichael	Odd prime powers	$>$ 7694 prime factors
$[n]^*$	$n-1$	$\phi(n)$	Prime	Primes	None
$[n]^*$	$n-1$	$n-1$	Prime	Primes	None
$[n]^*$	$\phi(n)$	$\phi(n)$	All n	All n	None
$[n]^*$	$\phi(n)$	$n-1$	Odd & Pseudo-Carmichael	Odd prime powers	Carmichaels, $45, 225, 325, \ldots$

if and only if $a^n \equiv a \pmod{p}$ for all a, and thus the pseudo-Carmichael condition is equivalent to $a^n \equiv a \pmod{n'}$ (compare this with the classical Carmichael condition $a^n \equiv a \pmod{n}$).

2.6 Conclusion

The above table summarizes the characterizations and known examples for each of the conditions in the Eightfold Way. We ignore for the purposes of this table the trivial case $n = 2$.

Finally, it is worth noting a related set of problems that we have not discussed. Given the summation/product characterizations of Giuga and co-Giuga numbers in Theorems 2.7 and 2.8, it is natural to ask what happens if we relax the condition that the indices be prime. In other words, which sequences of integers $\{n_1, n_2, n_3, \ldots, n_m\}$ satisfy

$$\sum_{i=1}^{m} \frac{1}{n_i} - \prod_{i=1}^{m} \frac{1}{n_i} \in \mathbf{Z},$$

and similarly, which sequences satisfy

$$\sum_{i=1}^{m} \left(1 - \frac{1}{n_i}\right) - \prod_{i=1}^{m} \left(1 - \frac{1}{n_i}\right) \in \mathbf{Z}?$$

Borwein et al (**B**^{3}**G**) call the former a *Giuga sequence* and have found all such sequences up to $m = 7$. Brenton (**BrB**, **BrD**, **BrJ**) and his students (**BJM**) at Wayne State University have studied this problem extensively, and especially the dual problem

$$\sum_{i=1}^{m} \frac{1}{n_i} + \prod_{i=1}^{m} \frac{1}{n_i} = 1.$$

They also showed a correspondence between solutions of this equation and complex surface singularities which are homologically trivial.

Recently, Connie Mangilin at Wayne State has found (personal communication) a new Giuga number with factorization

$$2 \cdot 3 \cdot 7 \cdot 43 \cdot 1831 \cdot 138683 \cdot 2861051 \cdot 1456230512169437.$$

A systematic study of co-Giuga sequences appears to be less feasible as there does not seem to be as much structure (or known examples!). Note how we were able to obtain very large bounds on co-Giuga numbers in Theorem 2.9 with the small value of $k = 3$ and relatively little effort, compared to the extensive computations in (**B**^{3}**G**). Unlike the other two problems, however, it is not clear that the n_i must be relatively prime. If we do require the n_i to be relatively prime, then there are no solutions with $m \leq 35$ by a straightforward calculation. Even if we require only that the n_i be distinct, a similar calculation shows that no solutions with $m \leq 8$ exist.

Chapter 3

Lehmer's Conjecture

3.1 Introduction

D. H. Lehmer conjectured in 1932 (**Lehm**) that $\phi(n) \mid n - 1$ if and only if n is prime. He showed that there were no counterexamples where n has less than 7 prime factors, and also found examples of composite n such that $\phi(n) \mid n + 1$.

In either case ($n \pm 1$), the number n must be relatively prime to $\phi(n)$. From this we see immediately that n must be squarefree, and the prime factors of n must form a normal family (in fact, in the $n - 1$ case, n must be a Carmichael number).

Subsequent work on Lehmer's conjecture has primarily concentrated on larger bounds for $\omega(n)$, the number of distinct prime factors of n. Lehmer himself wrote that a proof of the conjecture "seems about as remote as the proof of the nonexistence of odd perfect numbers". Pomerance (**Pom**) considered the analytic question of how many composite $n \leq x$ can satisfy $\phi(n) \mid n - a$ for a given a, and obtained an upper bound of $O(x^{1/2}(\ln x)^{3/4}(\ln\ln x)^{-1/2})$. Shan Zun (**Shan**) reduced the exponent of $\ln x$ from 3/4 to 1/2.

Let $n = p_1p_2\cdots p_l$ where $p_1 < p_2 < \cdots < p_l$. Then $k\phi(n) = n \pm 1$ is equivalent to

$$k\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_l}\right) = 1 \pm \frac{1}{n}. \tag{3.1}$$

We focus our efforts on the value $k = 2$. For larger k, say, $k = 3$, we need at least 32 prime factors (**Lehm**) before the product term $\prod_{i=1}^{l}\left(1 - \frac{1}{p_i}\right)$ is sufficiently close to $\frac{1}{3}$.

We note the following useful fact:

Proposition. If $3 \mid n$ and $k\phi(n) = n - 1$, then $k \geq 4$.

Proof. Suppose $k = 2$ and let $n = 3m$. Then $\phi(n) = 2\phi(m)$ (because n is squarefree) and we have $4\phi(m) = 3m - 1$. Now, m can have no prime factor congruent to 1 modulo 3, for then $\phi(m)$ is divisible by 3. Therefore every prime factor of m is congruent to 2 modulo 3, so $4\phi(m) \equiv 1 \not\equiv 3m - 1 \pmod 3$. Thus $k > 2$, and $k \neq 3$ since 3 cannot divide $n - 1$. ☺

This greatly reduces the search space required for finding bounds on $\omega(n)$ where $\phi(n) \mid n - 1$, as it means that any "small" counterexample must have 5 as its smallest prime factor, which in turn means that more small prime factors are required to bring the product term $\prod_{i=1}^{l}\left(1 - \frac{1}{p_i}\right)$ down towards $\frac{1}{2}$.

Lieuwens (**Lieu**) showed that if $3 \mid n$, then $\omega(n) \geq 212$ and in the general case $\omega(n) \geq 11$. Kishore (**Kish**) improved this to $\omega(n) \geq 13$, and Cohen & Hagis (**CH**) have raised this to $\omega(n) \geq 14$. Hagis (**Hag**) improved the $3 \mid n$ result to $\omega(n) \geq 298848$. In this section we improve Cogen & Hagis' general case result to $\omega(n) \geq 15$.

For the case of $\phi(n) \mid n + 1$, little is known beyond Lehmer's original work. Part of the reason for this is that any small example requires that $3 \mid n$, and the resulting search space for even $\omega(n) = 8$ is (as we shall see) enormous!

3.2 Computational Technique

The method we use to analyze the problems $\phi(n) \mid n \pm 1$ is fairly straightforward. We fix $\omega(n)$ to a constant, l, and compute "prefix strings" $[p_1, p_2, \ldots, p_r]$ which are candidates for the smallest r prime factors of n. We derive the set of prefixes of length r by iterating over prefixes of length $r - 1$, deducing upper and lower bounds for p_r, and adjoining all the primes in this range which are normal to the previous $r-1$.

In this manner we generate a complete list of prefix strings of length $l-2$. We then apply one of two techniques to find any examples of length l beginning with each prefix. One method is to use the above technique again to obtain a set of prefixes of length $l-1$, and solving the resulting linear equation for p_l. The other technique involves solving a quadratic Diophantine equation for p_{l-1} and p_l. We go into each method in detail and describe how we choose between them in the following subsections.

The Maple source codes for each stage of the process are included in the appendix.

3.2.1 Obtaining bounds for p_r

Suppose n satisfies $2\phi(n) = n + \epsilon$ ($\epsilon = \pm 1$) and let $\mathcal{P} = \{p_1 < p_2 < \cdots < p_l\}$ be the set of prime factors of n. We will determine bounds for p_r based on the values of $p_1, p_2, \ldots, p_{r-1}$.

We rewrite equation (3.1) with $k = 2$ as:

$$\underbrace{\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right)\cdots\left(1-\frac{1}{p_{r-1}}\right)}_{=Q_{r-1}}\left(1-\frac{1}{p_r}\right)\underbrace{\left(1-\frac{1}{p_{r+1}}\right)\cdots\left(1-\frac{1}{p_l}\right)}_{\geq\tau} = \frac{1}{2}+\frac{\epsilon}{2n}.$$

Let Q_{r-1} denote the initial product $\prod_{i=1}^{r-1}\left(1-\frac{1}{p_i}\right)$, and let τ be a lower bound for the terminal product $\prod_{i=r+1}^{l}\left(1-\frac{1}{p_i}\right)$. In addition let n' be a lower bound for n. For the purposes of our computation, Q_{r-1} is known, and τ and n' can be estimated initially by taking the product over the first $(l-r)$ primes greater than p_r which are normal to $\mathcal{P}_{r-1}$. As we obtain better lower bounds for p_r our estimates for τ and n' can be refined.

We now have the inequalities:

1. $Q_{r-1}\left(1-\frac{1}{p_r}\right) \geq \frac{1}{2}+\frac{\epsilon}{2n}$, and

2. $Q_{r-1}\left(1-\frac{1}{p_r}\right)\tau \leq \frac{1}{2}+\frac{\epsilon}{2n}$.

At this point we separate the two cases $\epsilon = +1$ and $\epsilon = -1$.

Case 1: $\epsilon = +1$. Inequality (1) gives us $1-\frac{1}{p_r} > \frac{1}{2}Q_{r-1}^{-1}$, and this results in a lower bound m for p_r. We can "round up" m to $\lceil \mathcal{P}_{r-1}, m \rceil$, as p_r must be normal to $\mathcal{P}$. This affords us a higher estimate for n'. Taking τ to be $\left(1-\frac{1}{p_r}\right)^{l-r}$, we use Inequality (2) to obtain $\left(1-\frac{1}{p_r}\right)^{l-r+1} < \left(\frac{1}{2}+\frac{1}{2n'}\right)(\tau Q_{r-1})^{-1}$, giving us an upper bound M for p_r, which we "round down" to $\lfloor \mathcal{P}_{r-1}, M \rfloor$.

Case 2: $\epsilon = -1$. Inequality (1) gives us $Q_{r-1}\left(1-\frac{1}{p_r}\right) \geq \frac{1}{2}-\frac{1}{2n'}$, resulting in a lower bound m for p_r. This lower bound allows us to obtain a better estimate for n', which in turn may give a new lower bound m. In practice, we iterate this cycle only a few times; we expect it to converge fairly rapidly. Using the same estimate for τ as in the Case 1, we arrive at $\left(1-\frac{1}{p_r}\right)^{l-r+1} < \frac{1}{2}(\tau Q_{r-1})^{-1}$, giving us an upper bound M for p_r.

3.2.2 Resolving prefixes of length $l-2$

Once $p_1, p_2, \ldots, p_{l-2}$ are fixed, it remains to determine p_{l-1} and p_l. One simple method is to use the techniques of the previous subsection to obtain bounds for p_{l-1}, and iterate through all feasible values of p_{l-1}. The equation $2\phi(n) = n+\epsilon$ reduces to a linear equation in p_l, namely $Ap_l = B$ where $A = 2\prod_{i=1}^{l-1}(p_i-1) - \prod_{i=1}^{l-1}p_i$ and $B = 2\prod_{i=1}^{l-1}(p_i-1)+\epsilon$, and we simply check that B/A is a prime integer.

This is essentially (minus some modular refinements) the method Cohen & Hagis used to eliminate $\omega(n) = 13$ for $\epsilon = -1$. However, for $l = 14$ the search space for p_{l-1} on certain prefixes becomes unwieldy for even today's computers. For these prefixes we use a second technique, essentially the one used by Lehmer.

The equation $2\phi(n) = n+\epsilon$ can be treated as a quadratic Diophantine equation in p_{l-1} and p_l, namely

$$Ap_{l-1}p_l - B(p_{l-1}+p_l) + C = 0,$$

where

- $A = 2\prod_{i=1}^{l-2}(p_i - 1) - \prod_{i=1}^{l-2} p_i$,
- $B = 2\prod_{i=1}^{l-2}(p_i - 1)$, and
- $C = 2\prod_{i=1}^{l-2}(p_i - 1) - \epsilon$.

Multiplying both sides by A and completing the square gives

$$(Ap_{l-1} - B)(Ap_l - B) = B^2 - CA.$$

Thus, if we have a prime factorization of $B^2 - CA$, we can look for divisors (positive or negative) of $B^2 - CA$ which are congruent to $-B$ modulo A. For each of these divisors we solve for p_{l-1} and p_l and check that they are positive and prime.

Modern factoring methods have allowed us to apply this method for fairly large values of A, B, C. However, the time it takes to factor a number in the range we are dealing with can range between seconds and hours. A typical prefix for $l = 14$ and $\epsilon = -1$ is $\{5, 7, 13, 17, 19, 23, 37, 59, 67, 73, 719, 1213\}$. This results in the Diophantine equation

$$354463903271283331pq - 31511686153110552576(p+q) + 31511686153110552577 = 0,$$

which, in turn, requires the factorization of the 39-digit number

$$991869388684890213907624869575058476789.$$

This takes nearly 15 minutes of *Maple* computation time on our SGI R4000 server, so clearly there is a trade-off between the factoring method and the exhaustive method. In practice we used the following procedure.

We first go through the prefixes of length $l - 2$ and apply the factoring method using the *Maple* procedure `ifactor(easy)`, or apply the exhaustive method if the search space is very small. This eliminates the trivial factorizations (small primes times a prime or prime power), and resolves a fair number of cases in a relatively short time. The remaining cases are then sorted by the size of the estimated range of p_{l-1}. We then perform two simultaneous runs: the factoring method on the large ranges of p_{l-1}, and the exhaustive method on the small ranges. At the point these runs meet, we are done. As a final check, we go through all the factorizations to verify that the factors are indeed primes rather than pseudoprimes. We use François Morain's *Elliptic Curves and Primality Proving* (ECPP) package (**AM**) for this last step.

3.3 Results

3.3.1 The $n - 1$ case

Verification of $\omega(n) > 13$:

We ran the *Maple* script **gen.mat** (see Appendix A) for 13 prime factors starting with $\{5\}$. This generated 730 11-prime prefixes ranging from $\{5, 7, 13, 17, 19, 23, 37, 59, 67, 73, 317\}$ to $\{5, 7, 13, 17, 19, 37, 47, 67, 73, 83, 89\}$. The prefix with the largest search space for p_{l-1} was $\{5, 7, 13, 17, 19, 23, 37, 59, 67, 89, 173\}$ with $25855 < p_{l-1} < 51710$. Since this is still fairly small, we tested every prefix with the exhaustive method rather than factoring. The entire process took a total of less than 10 minutes of computation on our SGI R4000 server. We then ran the same script for 12 and fewer factors, with similarly less computation time.

Proving $\omega(n) > 14$:

To rule out the case $\omega(n) = 14$, we ran **gen.mat** on our (significantly faster) SGI R10000 Challenge server. It took less than 15 minutes to generate 29631 feasible 12-prime prefixes ranging from $\{5, 7, 13, 17, 19, 23, 37, 59, 67, 73, 317, 5483\}$ to $\{5, 7, 13, 17, 19, 47, 59, 67, 73, 83, 89, 97\}$. This first prefix has the large bounds $14242698 < p_{l-1} < 28485396$, which suggests that the search space required to eliminate the next case, $\omega(n) = 15$, would be quite large indeed! (We estimate the number of 13-prime prefixes for this problem in the next section.)

26229 of these prefixes yielded to the first pass of eliminating small search spaces and trivial factorizations. We then sorted the remaining 3402 prefixes and ran them through the factoring and exhaustive methods as described above. We ran the exhaustive search on our UltraSPARC 1, while the factorizations were done on a team of 3 PowerMac 7600/132's. Two of the factorizations that *Maple* could not perform were done using *PARI/GP* and then manually converted to *Maple* code. The two runs were complete in less than 3 days, and no solutions were found. The final step was to verify the resulting 3492 primes with *ECPP*, which took only a few minutes.

3.3.2 The $n + 1$ case

Lehmer found the following eight solutions to $\phi(n) \mid n + 1$ and showed that they are the only ones with less than 7 prime factors.

- $2, \quad 3, \quad 3 \cdot 5, \quad 3 \cdot 5 \cdot 17,$
- $3 \cdot 5 \cdot 17 \cdot 257, \quad 3 \cdot 5 \cdot 17 \cdot 257 \cdot 65537,$
- $4919055 = 3 \cdot 5 \cdot 353 \cdot 929,$
- $6992962672132095 = 3 \cdot 5 \cdot 17 \cdot 353 \cdot 929 \cdot 83623937.$

Lehmer noted that if n satisfies $\phi(n) \mid n+1$ and $n+2$ is prime, then $n \cdot n+2$ is also a solution. This explains the solutions consisting of consecutive Fermat primes, and if $F_5 = 4294967297$ were prime, we would have one more. This fact did not escape Lehmer's attention, as he also noted that another solution would be generated from the last one if 6992962672132097 were prime, but did not have the means to verify that this number is in fact composite.

The above eight examples are still the only known, but we extend the bound for any further examples to at least 8 prime factors. There are 6826 feasible 5-prime prefixes ranging from $\{3, 5, 17, 257, 65537\}$ to $\{3, 5, 53, 83, 89\}$. The first prefix has the very large bounds $4294967296 < p_{l-1} < 8589934592$. Since the prefixes contain relatively small numbers there is little reason not to use the factoring method. The large bounds on p_{l-1} are, however, a good indicator of the search space required to rule out $\omega(n) = 8$, and in the next section we estimate the number of cases required to do so.

Performing this method on all 6826 prefixes (except for those with small search spaces) resulted in no new solutions with 8 prime factors. The process took about 10 hours and then a few more hours to verify primality of the resulting 10096 primes with *ECPP*.

3.4 Conclusion

The last improvement to the lower bound for $\omega(n)$ in Lehmer's conjecture was made fifteen years ago. It is interesting therefore to consider how much computational power would be needed to obtain $\omega(n) > 15$. To estimate the magnitude of this problem, we compute all 12-prime prefixes $\mathcal{P} = \{p_1, p_2, \ldots, p_{12}\}$, and obtain the usual upper and lower bounds $m \leq p_{13} \leq M$. We then approximate the number of primes in the range that are normal to $\mathcal{P}$ by

$$\prod_{i=1}^{12} \left(1 - \frac{1}{p_i - 1}\right) \left(\frac{M}{\ln M} - \frac{m}{\ln m}\right).$$

Summing this quantity over all 12-prime prefixes we get a reasonably accurate estimate of the number of 13-prime prefixes required to apply our current methods to the problem of $\omega(n) = 15$: about 31 million. Furthermore, many of the numbers we are required to factor will likely be in the 50 to 60 digit range. Even assuming that the average case can be solved in the same amount of time, it would still take several decades with the techniques and machinery we use here!

The extension of $n + 1$ to $\omega(n) > 8$ exhibits even more explosive growth: the estimated number of 6-prime prefixes is over 216 million! But while these figures are several orders of magnitude larger than the ones we are dealing with, it does not seem unreasonable that within a few years it will be quite feasible to tackle these much larger problems. The speed of factoring could certainly be improved over the current *Maple* implementation, and the speed of the brute force search would likely be increased by rewriting it in *C*. Indeed, it should even be possible today, through a distributed effort involving many computers, as the problem is highly parallelizable. For example, the *Great Internet Mersenne Prime Search* (`http://www.mersenne.org/prime.htm`), recently discovered the 35th known Mersenne prime $2^{1398269}-1$ (also the largest known prime), and the *DESCHALL* effort (`http://www.frii.com/~rcv/desinfo.htm`), just succeeded in breaking a single 56-bit DES key.

While most of the attention to Lehmer's problems has been focused on the $n-1$ case, the $n+1$ case is perhaps more interesting from a computational perspective as it is known to have solutions. In fact, if we relax the problem (as we did in Section 2.6) so that the p_i do not have to be prime, we find that there is a similarity between the $n+1$ problem

$$n_i - 1 \left| \left(\prod_{i=1}^{m} n_i \right) + 1, \qquad \forall 1 \le i \le m, \right. \tag{3.2}$$

and the Egyptian fraction problems

$$\sum_{i=1}^{m} \frac{1}{n_i} \pm \prod_{i=1}^{m} \frac{1}{n_i} = 1 \tag{3.3}$$

studied by Brenton. Namely, if $\mathcal{N} = \{n_1, n_2, \dots n_m\}$ is a solution to (3.2), then so is $\mathcal{N} \cup \{\prod_{i=1}^{m} n_i + 2\}$. Likewise, if $\mathcal{N}$ is a solution to either form of equation (3.3), then so is $\mathcal{N} \cup \{\prod_{i=1}^{m} n_i \pm 1\}$, respectively. It may therefore be possible to apply some of the techniques used by Brenton (**BrB**, **BrJ**) to Lehmer's problem to obtain new classes of solutions to (3.2).

Appendix: Source Code Listings

These files are available online at `http://www.cecm.sfu.ca/~erick/lehmercode/`.

Prefix Generation

```
# gen.mat: A Maple file for generating all length (M-2) prefixes for
#          candidates to Lehmer's conjectures with M primes factors
#          with a prescribed starting vector.  We use the global
#          variable A to store the current prefix string.  tol is
#          a measure of the maximum fan-out for the recursive search.
#
with(numtheory):

# feasible(p,A,n) returns true if p is normal to the first n primes of A
# and false otherwise.
#
feasible := proc(p,A,n) local i;
  for i from 1 to n do
    if p mod A[i] = 1 then RETURN(false) fi;
  od;
  RETURN(true);
end:

# nextf(p,A,n) returns the smallest prime greater than p which is normal
# to the first n primes of A, i.e. $\lceil A_n, p+1 \rceil$.
#
nextf := proc(p,A,n) local q;
  q := nextprime(p);
  while not feasible(q,A,n) do q := nextprime(q); od;
  RETURN(q);
end:

# doit(n,M,L,P,eps) is a recursive function.  n represents the length
# of the current partial prefix, so it reflects the recursive depth.
# M is the number of prime factors in the final candidate.  As soon as
# n reaches M-2 a prefix is printed out.  L is the partial product of
# the first n primes of A, and P is phi(L).  eps is +1 or -1 depending
# on which version of Lehmer's conjecture we are looking at.
#
doit := proc(n,M,L,P,eps)
global A, tol;
local lo, hi, i, minN, maxN, foo, bar, p;

# First, compute lower and upper bounds for the (n+1)-th prime.
#
if (eps = 1) then      # The case phi(n) | n+1
  lo := ceil(1/(1 - L/(2*P)));
  if (lo <= A[n]) then lo := A[n]+1; fi;

  minN := L*(lo^(M-n));
  foo := (1+1/minN)*L/(2*P);
  bar := evalf( foo ^ (1/(M-n)) );

  hi := ceil(1/(1-bar));
elif (eps = -1) then   # The case phi(n) | n-1
  lo := A[n]+1;
  # We do two iterations to approximate the lower bound
  minN := L*(lo^(M-n));
  foo := (1-1/minN)*L/(2*P);
  lo := ceil(1/(1-foo));

  minN := L*(lo^(M-n));
  foo := (1-1/minN)*L/(2*P);
  lo := ceil(1/(1-foo));

  if (lo <= A[n]) then lo := A[n]+1; fi;

  foo := 1*L/(2*P);
  bar := evalf( foo ^ (1/(M-n)) );
  hi := ceil(1/(1-bar));
else
  ERROR(`eps must be +1 or -1`);
fi;

# If we reach length M-2, then output the prefix as well as estimates
# for prime number M-1.
#
if (n = M-2) then
  lprint(`Feasible`, [seq(A[i],i=1..n)], lo..hi);
  RETURN(1);
fi;

# Recursively call doit() for every feasible value of the (n+1)-th
# prime.  If the fan-out is above tolerance we give up.
#
if (hi < lo) then
  lprint(`Impossible`, [seq(A[i],i=1..n)], lo..hi);
elif (hi > lo + tol) then
  lprint(`Pretty big`, [seq(A[i],i=1..n)], lo..hi);
```

```
else
  lprint(‘Expanding‘, [seq(A[i],i=1..n)], lo..hi);
  p := nextf(lo-1,A,n);
  while (p < hi) do
    A[n+1] := p;
    doit(n+1, M, L*p, P*(p-1), eps);
    p := nextf(p,A,n);
  od;
fi;

end:

# boot(B,M,eps) is used as a bootstrap to call doit() with the correct
# parameters.  B is the starting array (typically [3,5] for eps=+1 or
# [5] for eps=-1), and M is the target number of prime factors.
#
boot := proc(B,M,eps)
local iL, iP, C;
global A;

for i from 1 to nops(B) do
  A[i] := B[i];
od;

iL := convert(B,‘*‘);
C := map(x->x-1,B);
iP := convert(C,‘*‘);

doit(nops(B),M,iL,iP,eps);
end:

Digits := 100:
tol := 2000:
```

```
interface(screenwidth=1000):

lprint(‘%%Run begins here%%‘);
```

Brute-Force Solver

```
# qsolve.mat: Returns a list of pairs [p,q] satisfying n*p*q+eps
#             = ph(p-1)(q-1) by linear search on p.  The range for
#             p is passed by the global variable currange.  Every
#             integer solution is returned (even non-primes).
#
qsolve := proc(n,ph,eps)
local p,q,r,start,finish,a,b,L;
global curA, currange;

# Round start and finish to odd numbers
#
start:=lhs(currange); start:=start+1-(start mod 2);
finish:=rhs(currange); finish:=finish-1+(finish mod 2);

L := NULL;

for p from start to finish by 2 do
  a := 2*ph*(p-1) - n*p;
  b := eps + 2*ph*(p-1);
  if (a <> 0) then
    q:=iquo(b,a,’r’); if (q>0 and r=0) then L := L,[p,q,‘YES‘]; fi;
  fi;
od;

# Add a special marker if we found nothing
```

```
#
if (nops([L])>0) then
  RETURN([L]);
else
  RETURN([[currange,'empty','NO']]);
fi;

end:
```

Factoring Solver

```
# pqsolve.mat: Solves the quadratic equation a*pq - b*(p+q) + c = 0
#              using factoring techniques.  If the global variable
#              "easyonly" is true then we only use ifactor(easy),
#              and if this fails we return an error.
#
with(numtheory):

pqsolve := proc(a1,b1,c1)
local disc,S,i,bp,nbp,sp,L,p,q,g,
      a,b,c;
global curA, currange, easyonly;

# First, extract the GCD of a and b and make sure c is divisible by it.
# Otherwise there are no solutions.
#
g:=igcd(a1,b1);
if (igcd(g)>1) and (irem(c1,g) <> 0) then RETURN([[g,'gcd','NO']]); fi;

# Divide through by the GCD and compute the discriminant.
#
a:=a1/g; b:=-b1/g; c:=c1/g;
```

```
disc := b^2 - a*c;

# These cases should not happen in real life.  They must be errors.
#
if (a <= 0) then
  appendto(‘misfits‘);
    printf(‘%a\n‘,curA,currange);
  appendto(terminal);
  RETURN([[disc,‘negative a‘,‘ERROR‘]]);
fi;

if (disc = 0) then
  appendto(‘misfits‘);
    printf(‘%a\n‘,curA,currange);
  appendto(terminal);
  RETURN([[disc,‘zero disc‘,‘ERROR‘]]);
fi;

# Try to factor it.  If factorization is incomplete give up (assuming
# easyonly is true) and write it out to a file.
#
if (easyonly = true) then
  if (not type(ifactor(disc,easy),facint)) then
    appendto(‘tuff‘);
      printf(‘%a %a,%a\n‘,rhs(currange)-lhs(currange),curA,currange);
    appendto(terminal);
    RETURN([[disc,‘can’t factor‘,‘TOOHARD‘]]);
  fi;
fi;

# bp and nbp are the residues of the factors we’re looking for.
#
bp := b mod a; nbp = -b mod a;
```

```
# Write factorization to a file for later verification.
#
S := divisors(disc);
appendto(`primes`);
  printf(`%a\n`,ifactors(disc));
appendto(terminal);

L := NULL;

# Match both negative and positive factors congruent to -b mod a.
#
for i from 1 to nops(S) do
  if (S[i]*S[i] <= abs(disc)) then
    sp := S[i] mod a;
    if (sp = nbp) then
      p := -(S[i] + b)/a; q := -(disc/S[i] + b)/a; L := L,[p,q,`YES`];
    fi;
    if (sp = bp) then
      p := (S[i] - b)/a; q := (disc/S[i] - b)/a; L := L,[p,q,`YES`];
    fi;
  fi;
od;

# Add a special marker if we found nothing
#
if (nops([L])>0) then
  RETURN([L]);
else
  RETURN([[disc,`empty`,`NO`]]);
fi;
end:
```

Initial Scan

```
# easy.mat: Solves only the easy cases, where the linear search space
#           is less than 5000, or the factoring is trivial.  The code
#           for solving the other cases is just a simple modification.
#
read `qsolve.mat`:
read `pqsolve.mat`:

doit := proc(A,range,eps)
local Ap, n, ph;
global curA, currange;

#print(`Solving`,A);
Ap := map(x->x-1,A);

n := convert(A,`*`);
ph := convert(Ap,`*`);

curA:=A; currange:=range;
if (rhs(range)-lhs(range) <= 5000) then
printf(`Result %a,%a\n`,A,qsolve(n,ph,eps));
else
printf(`Result %a,%a\n`,A,pqsolve(2*ph-n,2*ph,2*ph-eps));
fi;

end:

interface(screenwidth=1000):
easyonly:=true:
lprint(`%%Run begins here%%`);
read `easy.in`;
```

Translators

```
#!/usr/local/bin/perl
# tr12: This perl script converts the output of gen.mat into a sequence
#       of doit() commands suitable to be read into easy.mat.  Requires
#       a parameter eps equal to -1 or +1.
#
if ($#ARGV!=0) {die("Usage: $0 eps\n");} else {$foo=$ARGV[0];shift;}
while(<>)
{chop;if(/Feasible\s*/){$str=$';$str=~/\]/;$str="doit($`$&,$',$foo);";
$str=~tr/ //d;$str=~tr/,/,/s;print "$str\n";}}
```

```
#!/usr/local/bin/perl
# tr23: This perl script converts the ‘tuff’ file output by easy.mat
#       into a sequence of doit() commands.  To sort them from easiest
#       to hardest use ‘sort -n tuff | tr23 [+1/-1]’.  For the reverse
#       order use ‘sort -nr tuff | tr23 [+1/-1]’.
#
if ($#ARGV!=0) {die("Usage: $0 eps\n");} else {$foo=$ARGV[0];shift;}
while(<>){chop;if(/\[/){$str=$&.$';$str=~tr/ //d;
          print "doit($str,$foo);\n";}}
```

```
#!/bin/sh
# getprimes: This shell script takes the factorizations dumped into
#            the ‘primes’ file and extracts the primes, then sorts them
#            into ‘primes.filt’ and removes duplicates.  Currently the
#            exponents are also included in the mix but they won't be
#            as large as the primes we want to verify.
#
perl -p -e '{tr/\s//d;tr/0-9/\012/cs;}' < primes > primes.filt
sort -un primes.filt -o primes.filt
```

Bibliography

[AGP] W. R. Alford, A. Granville, and C. Pomerance. "There are infinitely many Carmichael Numbers," *Annals of Mathematics*, 140:1–20, 1994.

[AM] A. O. L. Atkin and F. Morain. "Elliptic curves and primality proving," *Mathematics of Computation*, 203:29–68, 1993.

[Bed] E. Bedocchi. "Nota ad una congettura sui numeri primi," *Riv. Mat. Univ. Parma*, 11:229–236, 1985.

[B^{3}G] D. Borwein, J. M. Borwein, P. B. Borwein, and R. Girgensohn. "Giuga's conjecture on primality," *American Mathematics Monthly*, 103:40–50, 1996.

[BW] J. M. Borwein and E. Wong. "A survey of results relating to Giuga's conjecture on primality," *Proceedings of the 25th Anniversary Conference of the Centre de Récherches Mathématiques* (to appear), *CECM Preprint Series*, 95-035:1–23, 1995.

[BrB] L. Brenton and R. R. Bruner. "On recursive solutions of a unit fraction equation," *J. Austral. Math. Soc. Ser. A*, 57:341–356, 1994.

[BrD] L. Brenton and D. Drucker. "On the number of solutions of $\sum_{j=1}^{s}(1/x_j) + 1/(x_1 \cdots x_s) = 1$," *Journal of Number Theory*, 44:25-29, 1993.

[BrJ] L. Brenton and M. Joo. "On the system of congruences $\prod_{j \neq i} n_j \equiv 1 \bmod n_i$" *Fibonacci Quarterly*, 33:258–267, 1995.

[BJM] W. Butske, L. M. Jaje and D. R. Mayernik. "On the equation $\sum_{p|N} \frac{1}{p} + \frac{1}{N} = 1$, quasi-perfect numbers and perfectly weighted graphs," *(preprint)*.

[CH] G. L. Cohen and P. Hagis. "On the number of prime factors of n if $\phi(n) \mid (n-1)$," *Nieuw. Arch. Wisk.*, 28:177-185, 1980.

[Giu] G. Giuga. "Su una presumibile proprietà caratteristica dei numeri primi," *Ist. Lombardo Sci. Lett. Rend. A*, 83:511–528, 1950.

[Hag] P. Hagis. "On the equation $M\phi(n) = n-1$," *Nieuw. Arch. Wisk.*, 6:255-261, 1988.

[Hall] M. Hall. *The Theory of Groups.* Macmillan, New York, 1959.

[HW] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers.* Oxford University Press, Oxford, 1979.

[Kish] M. Kishore. *The Number of Distinct Prime Factors for Which $\sigma(N) = 2N$, $\sigma(N) = 2N \pm 1$ and $\phi(N) \mid N-1$.* Ph.D. Thesis, University of Toledo, Ohio, 39 pages, 1977.

[Lehm] D. H. Lehmer. "On Euler's totient function," *Bull. Amer. Math. Soc.*, 38:745–757, 1932.

[Leve] W. J. LeVeque. *Fundamentals of Number Theory.* Addison-Wesley, 1977.

[Lieu] E. Lieuwens. "Do there exist composite numbers for which $k\phi(M) = M-1$ holds?" *Nieuw. Arch. Wisk.*, 18:49–60, 1970.

[NZM] I. Niven, H. Zuckermann, and H. L. Montgomery. *An Introduction to the Theory of Numbers.* J. Wiley, New York, 1991.

[Pom] C. Pomerance. "On composite n for which $\phi \mid n-1$," *Pacific J. Math.*, 69:177–186, 1977.

[Rib] P. Ribenboim. *The New Book of Prime Number Records.* Springer-Verlag, New York, 1996.

[Shan] Z. Shan. "On composite n for which $\phi \mid n-1$," *J. China Univ. Sci. Tech.*, 15:109–112, 1985.