EVALUATING RECURRENCES OF FORM $X_{m+n} = f(X_m, X_n, X_{m-n})$ VIA LUCAS CHAINS

Peter L. Montgomery

December 13, 1983; Revised March, 1991 and January, 1992

Abstract. The Lucas function $V_n = V_n(P, 1)$ satisfies $V_{m+n} = V_m V_n - V_{m-n}$. Lucas chains resemble addition chains, with an identity $X_{m+n} = f(X_m, X_n, X_{m-n})$ replacing $x^{m+n} = x^m x^n$. We find a lower bound on the length of a Lucas chain, and present an algorithm which performs within 8% of that bound on average for prime $n < 10^6$.

1. Introduction

Let P and Q be elements of a commutative ring with identity. Define the Lucas functions $U_n(P,Q)$ and $V_n(P,Q)$ by

$$\begin{aligned} U_0(P, Q) &= 0, & U_1(P, Q) &= 1, & U_{n+2}(P, Q) &= PU_{n+1}(P, Q) - QU_n(P, Q), \\ V_0(P, Q) &= 2, & V_1(P, Q) &= P, & V_{n+2}(P, Q) &= PV_{n+1}(P, Q) - QV_n(P, Q) \end{aligned}$$

for nonnegative n. If Q^{-1} exists, then also define

$$U_{-n}(P, Q) = -Q^{-n} U_n(P, Q), \qquad V_{-n}(P, Q) = Q^{-n} V_n(P, Q)$$

for $n > 0$. If $x^2 - Px + Q = (x - \alpha)(x - \beta)$, then

$$(\alpha - \beta) U_n(P, Q) = \alpha^n - \beta^n, \qquad V_n(P, Q) = \alpha^n + \beta^n. \tag{1.1}$$

Lucas functions occur in primality testing algorithms [3][4][12][14][16], factorization algorithms [11][15], and combinatorics (see Conjecture 13).

The familiar Fibonacci and Lucas numbers are $F_n = U_n(1, -1)$ and $L_n = V_n(1, -1)$. The following are consequences of (1.1):

$$\begin{aligned} F_n &= F_{n-1} + F_{n-2}, \\ F_n &\ge 0 \qquad \text{if } n \ge -1, \\ |L_n| &= |L_{-n}| = L_{|n|}, \\ L_n &= L_{n-1} + L_{n-2} = F_{n+1} + F_{n-1} = F_{n+2} - F_{n-2}, \\ 5F_n &= 2L_n + L_{n-3}, \\ F_m F_n &= F_{m+n+1} - F_{m+1} F_{n+1} = F_{m+n-2} + F_{m-2} F_{n-2}, \\ 5F_m F_n &= L_{m+n} - (-1)^n L_{m-n}. \end{aligned}$$

1991 *Mathematics Subject Classification.* Primary 11B37; Secondary 11–04, 11B39.

Key words and phrases. Lucas chains, Lucas functions, recurrences, addition chains, continued fractions, Chebyshev polynomials.

This work was begun while the author was at System Development Corporation (now Unisys). Completed under U.S. Army fellowship DAAL03–89–G–0063 (1989–1992)

When $Q = 1$, then $V_n(P) = V_n(P, 1)$ is a monic polynomial of degree $|n|$ in P. This polynomial is related to Chebyshev polynomials of the first kind [1, pp. 776ff], since

$$V_n(2\cos x) = 2\cos nx, \qquad V_n(2\cosh x) = 2\cosh nx, \qquad V_n(x + x^{-1}) = x^n + x^{-n} \quad \text{if } x \neq 0$$

for real x. The first few such polynomials are:

$$\begin{aligned} V_0(P) &= 2, & V_2(P) &= P^2 - 2, & V_4(P) &= P^4 - 4P^2 + 2,\\ V_1(P) &= P, & V_3(P) &= P^3 - 3P, & V_5(P) &= P^5 - 5P^3 + 5P. \end{aligned}$$

These polynomials satisfy the following identities [8, 15]:

$$V_{-n}(P) = V_n(P), \tag{1.2}$$
$$V_{2n}(P) = V_n^2(P) - 2, \tag{1.3}$$
$$V_{m+n}(P) = V_m(P)V_n(P) - V_{m-n}(P), \tag{1.4}$$
$$V_{mn}(P) = V_m(V_n(P)). \tag{1.5}$$

The above are an example of the more general recurrence

$$X_{m+n}(P) = f(X_m(P), X_n(P), X_{m-n}(P)), \tag{1.6}$$
$$X_{mn}(P) = X_m(X_n(P)), \tag{1.7}$$
$$X_1(P) = P,$$
$$X_0(P) \equiv X_0 \qquad \text{(independent of } P\text{)}.$$

This more general recurrence is used for x-coordinates of multiples of a point P in one parameterization of elliptic curves [11, pp. 260–261].

Equation (1.7) follows from the others when $m \geq 0$. The proof is by induction on m. When $m = 0$, then X_0 is assumed to be a constant. When $m = 1$, both sides reduce to $X_n(P)$. For $m \geq 2$, use the induction hypothesis and (1.6) to verify that, with $P' = X_n(P)$,

$$\begin{aligned} X_{mn}(P) &= f(X_{mn-n}(P), X_n(P), X_{mn-2n}(P)) \\ &= f(X_{m-1}(P'), X_1(P'), X_{m-2}(P')) \\ &= X_m(P') = X_m(X_n(P)). \end{aligned}$$

Lucas chains resemble addition chains but represent algorithms for computing $X_n = X_n(P)$ from P for positive n via (1.6), rather than for computing x^n via $x^{m+n} = x^m x^n$. For example, the addition chain 1, 2, 3, 6, 12, 24, 25, 50, 100, 101, derived from the left-to-right binary expansion of 101, allows one to compute x^{101} from x by successively computing $x^1 = x$, x^2, x^3, x^6, x^{12}, x^{24}, x^{25}, x^{50}, x^{100}, x^{101}. We can use the first portion of this chain along with (1.6) to successively compute $X_1 = P$, X_2, X_3, X_6, X_{12}, X_{24} from P. But, although $x^{25} = x^{24}x^1$, (1.6) requires X_{23} if we try to apply it with $m = 24$ and $n = 1$. One can overcome this using the addition chain

$$1,\ 2,\ 3,\ 4,\ 6,\ 7,\ 12,\ 13,\ 25,\ 26,\ 50,\ 51,\ 101. \tag{1.8}$$

Now $X_{25} = f(X_{13}, X_{12}, X_1)$ and $X_{26} = f(X_{13}, X_{13}, X_0)$, for example.

We measure the cost of an algorithm for $X_n = X_n(P)$ in terms of how many evaluations of f it requires. For example, (1.8) requires 12 evaluations. This is a reasonable measure when the time required to evaluate f is independent of the arguments to f, such as with modular arithmetic or floating point arithmetic (but not polynomial arithmetic). Theorem 8 of Section 3 gives a lower bound on this cost. Algorithm CFRC of Section 5 performs within 2% of this bound on average for prime $n < 10^4$, but it is not suitable for implementation and its worst case performance is not $O(\log n)$. Algorithm PRAC of Section 7 overcomes these limitations and performs within 8% of this lower bound on average for prime $n < 10^6$. In contrast, the binary method costs 30% more than the lower bound predicts.

We let $\lfloor x \rfloor$ and $\lceil x \rceil$ designate the greatest integer not exceeding x and the least integer not less than x, respectively. The integer nearest to x is $\operatorname{round}(x) = \lfloor x + 0.5 \rfloor$. The truncated base 2 logarithm of the positive number x is $\lg x = \lfloor \log_2 x \rfloor$. It satisfies $\lg xy \geq \lg x + \lg y$. The greatest common divisor of two integers m and n is $\gcd(m, n)$.

The notation $(x_1, \ldots, x_n) \leftarrow (e_1, \ldots, e_n)$ designates a parallel assignment statement. The x_i must be distinct variables. To execute it, evaluate all expressions on the right. Then assign the value of each e_i to the corresponding x_i.

2. Binary Method

Let $n > 0$. As (1.8) illustrates we can evaluate $X_n = X_n(P)$ from P with $O(\log n)$ evaluations of f using the binary expansion of n and (1.6). If $n = 1$, then $X_n = X_1 = P$ is known. If $n \geq 2$ is even, then

$$X_n(P) = X_{n/2}(X_2(P)) = X_{n/2}(f(P, P, X_0)).$$

For arbitrary $n > 1$, let $m = \lceil n/2 \rceil$. The identities

$$\begin{aligned} X_{2m-2} &= f(X_{m-1}, X_{m-1}, X_0), \\ X_{2m-1} &= f(X_m, X_{m-1}, X_1), \\ X_{2m} &= f(X_m, X_m, X_0) \end{aligned} \tag{2.1}$$

express X_n and X_{n-1} in terms of X_m and X_{m-1}, since $X_1 = P$ and X_0 are known. Use (2.1) to recursively compute X_m and X_{m-1} until $m \leq 3$. The final computation of X_{n-1} can then be dropped.

For $n > 0$, let $L^{\mathrm{b}}(n)$ be the number of uses of (1.6) required to compute $X_n(P)$ by this algorithm. Then $L^{\mathrm{b}}(1) = 0$ and $L^{\mathrm{b}}(2n) = L^{\mathrm{b}}(n) + 1$. If $n > 1$ is odd, then

$$L^{\mathrm{b}}(n) = \begin{cases} 2\lg n - 1, & \text{if } n < 3 \cdot 2^{\lg n - 1}, \\ 2\lg n, & \text{if } n \geq 3 \cdot 2^{\lg n - 1}. \end{cases}$$

Equivalently,

$$L^{\mathrm{b}}(n) = \lg n + \lg(2n/3) \qquad (n \text{ odd},\ n > 1). \tag{2.2}$$

3. Lucas Chains and Lower Bounds on their Lengths

Let $n > 0$. An *addition chain for* n is an increasing sequence of integers

$$1 = a_0 < a_1 < \cdots < a_r = n$$

with the property that for $i = 1, 2, \ldots, r$ there exist j, k such that $a_i = a_j + a_k$ and $k \le j < i$. The *length* of the above chain is r. The length of the shortest addition chain for n is denoted by $\ell(n)$. By repeatedly using the identity $x^{m+n} = x^m x^n$, one can compute x^n from x and n with $\ell(n)$ multiplications. Knuth [7, pp. 441ff.] devotes several pages to addition chains.

By (1.6), we can compute X_{m+n} from X_m, X_n, and X_{m-n} with one evaluation of f. Define a *Lucas chain for* n to be an increasing sequence of integers

$$0 = a_{-1} < 1 = a_0 < a_1 < \cdots < a_r = n \tag{3.1}$$

with the property that for $i = 1, 2, \ldots, r$ there exist j, k, m such that $a_i = a_j + a_k$ and $a_m = a_j - a_k$ with $-1 \le k, m \le j < i$. When a_{-1} is removed, a Lucas chain becomes an addition chain, so many properties of addition chains apply to Lucas chains. In particular, the *length* of the above chain is defined to be r. Let $L(n)$ be the length of the shortest Lucas chain for n.

The binary method satisfies $L(n) \le L^{\mathrm{b}}(n) \le 2 \lg n$. On the other hand, $L(n) \ge \ell(n) \ge \lg n$. Therefore the binary method is optimal to within a constant factor. Many n satisfy $L(n) < L^{\mathrm{b}}(n)$, as Table 1 illustrates:

n	$L^{\mathrm{b}}(n)$	Binary Lucas chain for n	Shorter Lucas chain(s) for n
9	5	0, 1, 2, 3, 4, 5, 9	0, 1, 2, 3, 6, 9
13	6	0, 1, 2, 3, 4, 6, 7, 13	0, 1, 2, 3, 5, 8, 13
15	6	0, 1, 2, 3, 4, 7, 8, 15	0, 1, 2, 3, 5, 10, 15
			0, 1, 2, 3, 6, 9, 15
17	7	0, 1, 2, 3, 4, 5, 8, 9, 17	0, 1, 2, 3, 4, 7, 10, 17
			0, 1, 2, 3, 5, 6, 11, 17
			0, 1, 2, 3, 5, 7, 10, 17
			0, 1, 2, 3, 5, 7, 12, 17

Table 1. Some cases where binary method is not optimal

The cases $n = 9$ and $n = 15$ are typical of odd composite numbers. If $n = jk$ has a known factorization where neither j nor k is a power of 2, then it is shorter to use the binary method once for $X_k(P)$ and again for $X_j(X_k(P))$ than to apply the binary method directly to $X_{jk}(P)$.

Theorem 1. *If j and k are positive integers and neither is a power of 2, then $L^{\mathrm{b}}(jk) > L^{\mathrm{b}}(j) + L^{\mathrm{b}}(k)$.*

Proof. It suffices to consider the case where j and k are both odd. Add the four inequalities:

$$\begin{aligned}
\lg jk &\ge \lg j + \lg k, \\
\lg jk &\ge \lg(2j/3) + \lg(2k/3) + \lg(9/4), \\
\lg(2jk/3) &\ge \lg j + \lg(2k/3), \\
\lg(2jk/3) &\ge \lg(2j/3) + \lg k,
\end{aligned}$$

and use (2.2) three times to obtain $2L^{\mathrm{b}}(jk) \geq 2L^{\mathrm{b}}(j) + 2L^{\mathrm{b}}(k) + \lg(9/4) > 2(L^{\mathrm{b}}(j) + L^{\mathrm{b}}(k))$. ∎

Theorem 2. *If j and k are positive integers, then $L(jk) \leq L(j) + L(k)$.*

Proof. Designate $r = L(j)$ and $s = L(k)$. Let $0,\ 1,\ a_1,\ a_2,\ \ldots,\ a_r = j$ and $0,\ 1,\ b_1,\ b_2,\ \ldots,\ b_s = k$ be Lucas chains for j and k, of lengths r and s respectively. Then

$$0,\ 1,\ a_1,\ a_2,\ \ldots,\ a_r,\ a_r b_1,\ a_r b_2,\ \ldots,\ a_r b_s = jk$$

is a Lucas chain for jk of length $r + s$, so $L(jk) \leq r + s$. ∎

Corollary 3. *If j and k are positive integers and neither is a power of 2, then $L(jk) < L^{\mathrm{b}}(jk)$.*

Proof. Apply Theorems 1 and 2, using $L(j) \leq L^{\mathrm{b}}(j)$. ∎

The analog of Theorem 2 holds for addition chains [7, p. 445], but the analog of Theorem 1 does not hold for the binary method of exponentiation. For example, it is shorter to compute x^{33} as $x^{32} \cdot x$ than as $\left(x^{11}\right)^3$.

The conclusion of Theorem 2 cannot be strengthened to an equality. Examples where $L(jk) < L(j) + L(k)$ are $23 \cdot 53 = 1219$, $41 \cdot 53 = 2173$, $37 \cdot 83 = 3071$, and $37 \cdot 113 = 4181$. The Lucas chains

$$\begin{aligned}
&0,\ 1,\ 2,\ 3,\ 4,\ 7,\ 11,\ 18,\ 29,\ 47,\ 76,\ 123,\ 170,\ 293,\ 463,\ 756,\ 1219,\\
&0,\ F_2,\ F_3,\ \ldots,\ F_{12} = 144,\ F_{13} = 233,\ F_{14} = 377,\ 521,\ 898,\ 1275,\ 2173,\ 3071,\\
&0,\ F_2,\ F_3,\ \ldots,\ F_{18} = 2584,\ F_{19} = 4181
\end{aligned}$$

show that $L(1219) \leq 15$, $L(2173) \leq 16$, $L(3071) \leq 17$, and $L(4181) \leq 17$. We will later show that $L(23) = 7$, $L(37) = L(41) = 8$, $L(53) = 9$, and $L(83) = L(113) = 10$; Theorem 8 provides the necessary lower bounds. Another example is

$$\begin{gathered}
L(2 \cdot 17 \cdot 53 \cdot 109) = L(196418) = L(F_{27}) \leq 25,\\
L(2) + L(17) + L(53) + L(109) = 1 + 6 + 9 + 10 = 26.
\end{gathered}$$

Such examples seem rare, so we concentrate on the case where n is prime or where no prime divisor of n is known. By (1.7), it suffices to do these cases.

In an addition or Lucas chain, step i is called a *doubling step* if $a_i = 2a_{i-1}$. Knuth shows that short addition chains consist primarily of doubling steps. Theorem 5 shows that a Lucas chain for n cannot have many doubling steps unless n is highly composite. Theorems 7 and 8 use this to obtain bounds on $L(n)$.

Lemma 4. *If the Lucas chain (3.1) has exactly d doubling steps, then $n \leq 2^{d-1} F_{r-d+3}$.*

Proof. See [7, p. 448]. ∎

Theorem 5. *If $a_{i+1} = 2a_i$ in the Lucas chain (3.1), then $a_i \mid a_j$ for all $j \geq i$.*

Proof. By induction on j. This is evident for $j = i$ and $j = i + 1$. Assume that $j \geq i + 2 > 1$. The definition of Lucas chains implies the existence of k, ℓ, m such that $-1 \leq \ell,\ m \leq k < j$ where $a_j = a_k + a_\ell$ and $a_m = a_k - a_\ell$. From

$$2a_k = a_j + a_m \geq a_j > a_{i+1} = 2a_i,$$

we conclude that $a_k > a_i$ and hence $k > i$. By induction, a_i divides a_k. From

$$a_\ell + a_m = a_k \geq a_{i+1} = a_i + a_i,$$

we conclude that $a_\ell \geq a_i$ or $a_m \geq a_i$, and hence $\ell \geq i$ or $m \geq i$. By induction, either a_i divides a_ℓ or a_i divides a_m. In either case, a_i divides $a_j = a_k + a_\ell = 2a_k - a_m$. ■

Corollary 6. *If $a_{i+1} = 2a_i$ in the Lucas chain (3.1), then $r \geq L(a_i) + L(n/a_i)$.*

Proof. Define $b_j = a_{i+j}/a_i$ for $0 \leq j \leq r - i$. By Theorem 5, each b_j is an integer. Hence $0,\ a_0,\ a_1,\ \ldots,\ a_i$ and $0,\ b_0,\ b_1,\ \ldots,\ b_{r-i}$ are Lucas chains for a_i and for $b_{r-i} = a_r/a_i$, respectively. Therefore

$$r = i + (r - i) \geq L(a_i) + L(a_r/a_i) = L(a_i) + L(n/a_i). \quad ■$$

Theorem 7. *Let n be a positive integer with s prime divisors (including multiplicities). Then the number of doubling steps in a Lucas chain for n cannot exceed s, and $n \leq 2^{s-1} F_{L(n)-s+3}$.*

Proof. Let (3.1) be a Lucas chain for n. Suppose that $a_{i+1} = 2a_i$ for $i = i_1,\ i_2,\ \ldots,\ i_k$ where $i_1 < i_2 < \cdots < i_k$. Then

$$\begin{aligned} a_{i_1} \mid a_{i_2} \mid \cdots \mid a_{i_k} \mid n, \\ 1 \leq a_{i_1} < a_{i_2} < \cdots < a_{i_k} < n, \\ n = a_{i_2} \cdot \frac{a_{i_3}}{a_{i_2}} \cdot \ \cdots \ \cdot \frac{a_{i_k}}{a_{i_{k-1}}} \cdot \frac{n}{a_{i_k}}. \end{aligned}$$

This expresses n as a product of k integers greater than 1, so $k \leq s$. By Lemma 4, $n \leq 2^{k-1} F_{r-k+3} \leq 2^{s-1} F_{r-s+3}$. The claim follows by choosing (3.1) so that $r = L(n)$. ■

Theorem 8. *Let n be an integer. Let $r = L(n)$, and suppose that $r < L(s) + L(n/s)$ whenever $1 < s < n$ and $s \mid n$. Then*

(i) $n \leq F_{r+2}$.
(ii) *If $n \neq F_{r+2}$, then $n \leq F_{r+2} - F_{r-3}$.*
(iii) *If $n > L_r$, then $|5n - 2L_{r+2}|$ is a Lucas number.*

Proof. Let (3.1) be a Lucas chain for n. By Corollary 6, we may assume that $a_i \leq a_{i-1} + a_{i-2}$ for all $i > 1$. A simple consequence is $a_i \leq F_{i+2}$ for $i = 0,\ 1,\ \ldots,\ r$. Setting $i = r$ gives (i).

Suppose that $n \neq F_{r+2}$. Then there exists $m > 2$ such that $a_m \neq a_{m-1} + a_{m-2}$, so

$$\begin{aligned} a_m &\leq \max(a_{m-1} + a_{m-3},\ 2a_{m-2}) \\ &\leq \max(F_{m+1} + F_{m-1},\ 2F_m) \\ &= F_{m+1} + F_{m-1} = L_m = F_{m+2} - F_{m-2}. \end{aligned}$$

Use this and $a_{m-1} \leq F_{m+1}$ to derive

$$a_i \leq F_{i+2} - F_{i-m+1} F_{m-2} \qquad (m - 1 \leq i \leq r).$$

Set $i = r$. Use $F_{r-m-1} \geq 0$ if $r \geq m$, and $F_{m-4} \geq 0$ since $m > 2$, to obtain

$$a_r \leq F_{r+2} - F_{r-m+1}F_{m-2} = F_{r+2} - F_{r-3} - F_{r-m-1}F_{m-4} \leq F_{r+2} - F_{r-3}.$$

This proves (ii).

To prove (iii) we find all cases where $n > L_r$, and show that $|5n - 2L_{r+2}|$ is a Lucas number in each case. If $n = F_{r+2}$, then $5n - 2L_{r+2} = L_{r-1}$. If $a_i = a_{i-1} + a_{i-2}$ for all $i > 1$ except $i = m$, and if $a_m = a_{m-1} + a_{m-3}$, then $n = a_r = F_{r+2} - F_{r-m+1}F_{m-2}$, so

$$5n - 2L_{r+2} = 5F_{r+2} - 5F_{r-m+1}F_{m-2} - 2L_{r+2} = L_{r-1} - 5F_{r-m+1}F_{m-2} = (-1)^{m-2}L_{r-2m+3}.$$

It remains to show that $n \leq L_r$ in all other cases. If $a_m \leq 2a_{m-2}$ for some m, then

$$a_i \leq F_{i+2} - F_{i-m+1}F_{m-1} = L_i + F_{i-2} - F_{i-m+1}F_{m-1} = L_i - F_{i-m-1}F_{m-3}$$

for $m - 1 \leq i \leq r$. Consequently $n = a_r \leq L_r$. This argument also applies if $a_m \leq a_{m-1} + a_{m-4} \leq F_{m+1} + F_{m-2}$ for some m. The remaining case occurs when two (or more) values of m satisfy $a_m = a_{m-1} + a_{m-3}$, say $m = j$ and $m = k$ where $r \geq k > j > 2$. We successively verify that:

$$\begin{aligned}
a_i &\leq F_{i+2} && (0 \leq i < r),\\
a_{j-3} &\leq F_{j-1} = L_{j-2} - F_{j-3}, &&\\
a_{j-1} &\leq F_{j+1} = L_{j-1} + F_{j-3}, &&\\
a_j &= a_{j-1} + a_{j-3} \leq L_j, &&\\
a_i &\leq L_i + F_{i-j}F_{j-3} && (j - 1 \leq i \leq r),\\
a_{k-3} &\leq F_{k-1} = L_{k-2} - F_{k-3}, &&\\
a_{k-1} &\leq L_{k-1} + F_{k-j-1}F_{j-3} = L_{k-1} + F_{k-3} - F_{k-j}F_{j-2}, &&\\
a_k &= a_{k-1} + a_{k-3} \leq L_{k-2} + L_{k-1} - F_{k-j}F_{j-2} = L_k - F_{k-j}F_{j-2} \leq L_k, &&\\
a_{k+1} &\leq a_{k-1} + a_k \leq (L_{k-1} + F_{k-j-1}F_{j-3}) + (L_k - F_{k-j}F_{j-2}) \leq L_{k+1} &&\\
a_i &\leq L_i && (k \leq i \leq r).
\end{aligned}$$

In the inequality for a_{k+1}, we used $0 \leq F_{k-j-1} \leq F_{k-j}$ and $0 \leq F_{j-3} \leq F_{j-2}$. In all cases, $n = a_r \leq L_r$. ■

Theorem 8 is useful when n is prime. We will soon show that this bound is very good, by exhibiting Lucas chains for several n which are as short as Theorem 8 permits.

4. Binary-Ternary Method for $X_n(P)$

The key idea behind the binary method is the ability to derive a Lucas chain containing $n - 1$ and n from a Lucas chain containing $k = \lceil n/2 \rceil$ and $k - 1$.

The Lucas chain 0, 1, 2, 3, 5, 6, 11, 17 for 17 illustrates the following. If $n = 3k - 1$, use the binary method to get a Lucas chain containing $k - 1$ and k. Append $2k - 1$ and $3k - 1$ to the end of the chain. A similar construction applies if $n = 3k - 2$. These may reduce the length of the binary Lucas chain for n by one, and never increase its length.

The intermediate steps of the binary method, where one needs a Lucas chain containing both $m-1$ and m, can be improved to append $2k-1$, $3k-2$, and $3k-1$ to a Lucas chain containing $k-1$ and k, if $m = 3k-1$.

The event in the last paragraph occurs with probability 3/7 (not 1/3). For a heuristic proof, let p_i be the probability that m is congruent to i mod 6, for $i = 0, 1, 2, 3, 4, 5$. Assume that the behaviors modulo 12 and modulo 18 mirror that modulo 6, i.e.,

$$\begin{aligned}
&\Pr(m \equiv i \pmod{12}) = \Pr(m \equiv i+6 \pmod{12}) = p_i/2,\\
&\Pr(m \equiv i \pmod{18}) = \Pr(m \equiv i+6 \pmod{18}) = \Pr(m \equiv i+12 \pmod{18}) = p_i/3
\end{aligned}$$

for each i. If the probabilities exist, then they must satisfy

$$\begin{aligned}
p_0 &= p_0/2 + p_5/3, & p_3 &= p_0/2 + p_2/3,\\
p_1 &= p_1/2 + p_2/3, & p_4 &= p_1/2 + p_5/3,\\
p_2 &= p_3/2 + p_4/2 + p_5/3, & p_5 &= p_2/3 + p_3/2 + p_4/2,\\
1 &= p_0 + p_1 + p_2 + p_3 + p_4 + p_5.
\end{aligned}$$

The solution is $p_0 = p_1 = p_3 = p_4 = 1/7$ and $p_2 = p_5 = 3/14$. The average reduction of $\log k$ per term in the Lucas chain is

$$\frac{(3/7)\log 3 + (4/7)\log 2}{(3/7)3 + (4/7)2} = \frac{\log 432}{17}.$$

Let $L^{\mathrm{t}}(n)$ be the length of the Lucas chain for n generated by this method, called the binary-ternary method. For example, $L^{\mathrm{t}}(101) = 11$ since the chain is 0, 1, 2, 3, 5, 6, 11, 16, 17, 33, 34, 67, 101. For large n, $L^{\mathrm{t}}(n)$ is approximately $17\log_{432} n \sim 1.94\log_2 n$. One easily verifies that $L^{\mathrm{t}}(n) \le L^{\mathrm{b}}(n)$ for all odd n, using (2.2). There are infinitely many n (e.g. $n = 9\cdot 2^k + 1$) for which $L^{\mathrm{t}}(n) = L^{\mathrm{b}}(n)$.

5. Continued Fraction Method for $X_n(P)$

The binary-ternary method performs 3% better than the binary method on average. Theorem 8 suggests that much more improvement is possible.

The Lucas chain 0, 1, 2, 3, 4, 7, 10, 17 for 17 was found by trial and error. Any Lucas chain for 17 must include two numbers whose total is 17 and whose difference is in the chain. The binary method selects $17 = 9+8$ and the binary-ternary method selects $17 = 11+6$. Let's try $17 = 10+7$. Then we must include $10-7 = 3$ in the chain. Also, the chain must include two numbers totaling 10 whose difference is in the chain. Since $10 = 7+3$, we choose to include $7-3 = 4$ in the chain (another strategy observes that $10 = 5+5$, and includes 5 in the chain). At each stage, subtract the smallest number so far from the second smallest number so far and include the difference in the chain. Terminate when a zero difference is obtained. The resulting sequence is 17, 10, 7, 3, 4, 1, 2, 1, 0. Eliminate the duplicate 1 and rearrange to get 0, 1, 2, 3, 4, 7, 10, 17.

This is the same sequence obtained when computing gcd(17, 10) using only subtraction. It always generates a Lucas chain if the two starting numbers are coprime. The partition $17 = 12+5$ leads to the fourth Lucas chain for 17 in Table 1.

Let $/\!\!/ x_0, x_1, x_2, \ldots, x_k /\!\!/$ designate the continued fraction

$$x_0 + \cfrac{1}{x_1 + \cfrac{1}{x_2 + \cdots + \cfrac{1}{x_k}}}.$$

If $(n-r)/r = /\!\!/ x_0, \ldots, x_k /\!\!/$ where $\gcd(n, r) = 1$ and $0 < r < n$, then $L(n) \le x_0 + \cdots + x_k$. More precisely, there exists a Lucas chain for n of length $x_0 + \cdots + x_k$ containing both $n - r$ and r. This is easily proved by induction on $\max(n - r, r)$. For example, $62/39 = /\!\!/ 1, 1, 1, 2, 3, 2 /\!\!/$, and X_{101} can be computed by the Lucas chain 0, 1, 2, 3, 5, 7, 9, 16, 23, 39, 62, 101. This has length 10, whereas $L^{\mathrm{b}}(101) = 12$ and $L^{\mathrm{t}}(101) = 11$.

It remains to select $r = r(n)$ so as to minimize the sum of the partial quotients of $(n-r)/r$ – see next section.

A straightforward implementation of this algorithm requires one pass to compute and store the partial quotients of $(n - r)/r$, and a second pass in reverse order applying (1.6). The second pass can be eliminated, in view of:

Theorem 9. *Suppose $jk \equiv \pm 1 \pmod{n}$, where $0 < j, k \le n/2$. If $n/j = /\!\!/ x_0, \ldots, x_k /\!\!/$, where each x_i is positive, then $n/k = /\!\!/ x_k, \ldots, x_0 /\!\!/$.*

Proof. See [7, exercise 4.5.3–26]. ■

Corollary 10. *If $jk \equiv \pm 1 \pmod{n}$, and $0 < j, k < n$, then the sums of the partial quotients of $\frac{n-j}{j}$, $\frac{j}{n-j}$, $\frac{n-k}{k}$, and $\frac{k}{n-k}$ are all equal.*

For example, $39 \cdot 44 \equiv -1 \pmod{101}$. The regular continued fraction expansions of $101/39$ and $101/44$ are $/\!\!/ 2, 1, 1, 2, 3, 2 /\!\!/$ and $/\!\!/ 2, 3, 2, 1, 1, 2 /\!\!/$, respectively. These closely resemble those of $62/39$ and $57/44$, namely $/\!\!/ 1, 1, 1, 2, 3, 2 /\!\!/$ and $/\!\!/ 1, 3, 2, 1, 1, 2 /\!\!/$. We can compute the partial quotients of $62/39$ while generating the Lucas chain 0, 1, 2, 3, 5, 8, 13, 18, 31, 44, 57, 101, derived from $57/44$.

Algorithm CFRC is based upon these ideas. It computes $X_n(P)$ given n, P, and an integer r satisfying $0 < r \le n$ and $\gcd(n, r) = 1$. It finds the continued fraction expansion of $(n - r)/r$, with e/d representing the part not yet expanded. At the same time, it builds another fraction in a/b. When $d > e$, set $q = \lceil d/e \rceil - 1$, so $q + 1 \ge d/e > q$. The algorithm will take the $d > e$ branch q successive times, replacing d/e by $d/e - q$ and b/a by $b/a + q$. When $e \ge d$, set $q = \lfloor e/d \rfloor$. The algorithm will take the $e \ge d$ branch q successive times, replacing e/d by $e/d - q$ and a/b by $a/b + q$. In effect, it transfers partial quotients from d/e to b/a and from e/d to a/b. At all times, the algorithm maintains $A = X_a(P)$, $B = X_b(P)$, and $C = X_{a-b}(P)$.

Algorithm CFRC$(n,\, r,\, P)$
$(a,\, b,\, d,\, e,\, A,\, B,\, C) \leftarrow (1,\, 1,\, r,\, n-r,\, P,\, P,\, X_0)$
while $e \neq 0$ **do**
  if $d > e$ **then**
    $(b,\, d,\, B,\, C) \leftarrow (a+b,\, d-e,\, f(A,\, B,\, C),\, X_{-1}(B))$
  else
    $(a,\, e,\, A,\, C) \leftarrow (a+b,\, e-d,\, f(A,\, B,\, C),\, A)$
  end if
end while
return A
end CFRC

To obtain a Lucas chain for n, output the new values of a and b whenever they change. Otherwise variables a and b need not be explicitly manipulated.

The computation of $X_{-1}(B)$ in CFRC is free when $X_n(P) = V_n(P)$ is the Lucas sequence, since this sequence satisfies $X_{-1}(B) = B$. The cost can be avoided in other applications by remembering whether $C = X_{a-b}(P)$ or $C = X_{b-a}(P)$, using $f(A,\, B,\, C)$ or $f(B,\, A,\, C)$ on the next iteration.

Although Algorithm CFRC was discovered using continued fractions, it is easily verified without them [10]. The following invariants hold at the start of the **while** loop in CFRC:

$$\begin{aligned} &ad + be = n, \\ &d > 0, \qquad e \geq 0, \\ &A = X_a(P), \qquad B = X_b(P), \qquad C = X_{a-b}(P), \\ &\gcd(d,\, e) \mid \gcd(n,\, r). \end{aligned} \tag{5.1}$$

The last line reduces to $\gcd(d,\, e) = 1$ but has been generalized in anticipation of Algorithm PRAC of Section 7. When $e = 0$, then $d = \gcd(d,\, e) = 1$. The top equation simplifies to $a = n$, so $A = X_a(P) = X_n(P)$ upon termination.

6. Analysis of Algorithm CFRC

Algorithm CFRC is really a family of algorithms indexed by r. Which value of r should one use?

Let $n > 1$. For $0 < r < n$, if $\gcd(n,\, r) = 1$, let $L^{\mathrm{c}}(n,\, r)$ designate the sum of the partial quotients of $(n-r)/r$. Then the Lucas chain built by Algorithm CFRC has length $L^{\mathrm{c}}(n,\, r)$. Let $r(n)$ be a value of r minimizing $L^{\mathrm{c}}(n,\, r)$ and let $L^{\mathrm{c}}(n) = L^{\mathrm{c}}(n,\, r(n))$ be this minimum. What is the asymptotic value of $L^{\mathrm{c}}(n)$? How is $r(n)$ determined?

These seem to be open problems (still? TBD). Leo Moser [6, p. 144] conjectures that $L^{\mathrm{c}}(n) = O(\log n)$. We require that $L^{\mathrm{c}}(n) \leq L^{\mathrm{b}}(n)$ if the continued fraction method is to be competitive with the binary method. Table 2 lists $\ell(p)$, $L^{\mathrm{b}}(p)$, $L^{\mathrm{t}}(p)$, $L^{\mathrm{c}}(p)$, and $L^{\mathrm{p}}(p)$ (the

p	$\ell(p)$	$L^{\mathrm{b}}(n)$	$L^{\mathrm{t}}(p)$	$L^{\mathrm{c}}(p)$	$L^{\mathrm{p}}(p)$	Optimal values for $r(p)$
2	1	1	1	1	1	1
3	2	2	2	2	2	1
5	3	3	3	3	3	2
7	4	4	4	4	4	2, 3
11	5	5	5	5	5	3, 4
13	5	6	6	5	5	5
17	5	7	6	6	6	5, 7
19	6	7	7	6	6	7, 8
23	6	7	7	7	7	5, 7, 9, 10
29	7	8	8	7	7	8, 11, 12
31	7	8	8	7	7	12, 13
37	7	9	9	8	8	8, 10, 11, 14
41	7	9	9	8	8	11, 12, 15, 16, 17, 18
43	7	9	9	8	8	12, 18
47	8	9	9	8	8	13, 18
53	8	10	10	9	9	12, 14, 19, 22, 23
59	8	10	10	9	9	18, 23, 25, 26
61	8	10	10	9	9	17, 18, 22, 25
67	8	11	10	9	9	18, 26
71	9	11	10	9	9	21, 26, 27, 30
73	8	11	11	9	9	27
79	9	11	11	9	9	29, 30
83	8	11	11	10	10	18, 19, 22, 23, 30, 34, 35, 36
89	9	11	11	9	9	34
97	8	12	11	10	10	21, 26, 35, 36, 37, 41
101	9	12	11	10	10	30, 37, 39, 44
103	9	12	11	10	10	37, 39
107	9	12	12	10	10	41, 47
109	9	12	12	10	11	30, 40, 45, 46
113	9	12	12	11	10	21, 24, 30, 31, 33, 35, 40, 42, 43, 48, 49, 51
127	10	12	12	11	12	27, 29, 34, 35, 45, 47, 48, 49, 56, 57
131	9	13	12	10	10	50, 55
137	9	13	12	11	11	29, 31, 37, 52, 53
139	10	13	12	11	11	30, 39, 41, 51, 57, 61
149	9	13	12	11	11	34, 40, 41, 44, 55, 57, 65
151	10	13	13	11	11	56, 59, 62, 64
157	10	13	13	11	11	34, 46, 58, 60, 66, 69
163	9	13	13	11	11	44, 62, 63, 71
167	10	13	13	11	11	46, 60, 64, 69
173	10	13	13	11	11	64, 66, 73, 76
179	10	13	13	11	11	50, 68, 74, 75
181	10	13	13	11	12	50, 70, 75, 76
191	11	13	13	11	11	74, 80
193	9	14	13	11	11	81
197	10	14	13	12	12	43, 52, 55, 70, 71, 72, 76, 77, 86, 87
199	10	14	13	11	11	55, 76
Totals	364	472	459	404	406	

TABLE 2. Some numbers related to $L(p)$ for $p < 200$

last number refers to Algorithm PRAC of Section 7) for all prime $p < 200$. Table 2 also lists the optimal values of $r(p)$ below $p/2$ for Algorithm CFRC.

By Theorem 8 and Table 2, $L(p) = L^{\mathrm{c}}(p)$ for all prime $p < 200$ except possibly 113 and

197. Conclusion (iii) of Theorem 8 is needed only for $p = 127$. But $L(113) = 10 < 11 = L^c(113)$, as demonstrated by the Lucas chains

$$0,\ 1,\ 2,\ 3,\ 5,\ 6\ (\text{or } 8),\ 11,\ 16,\ 27,\ 43,\ 70,\ 113 \qquad \text{and}$$
$$0,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 16\ (\text{or } 21),\ 29,\ 42,\ 71,\ 113.$$

The exact value of $L(197)$ remains open.

If r is selected randomly, then $\mathrm{E}(L^c(p,r)) = O((\log p)^2)$ [17 TBD]. So we must be careful in our selection of r. A table was made of all p for which $L^c(p) \le 20$, by recursively enumerating all continued fractions whose sum of partial quotients is 20 or less. It included all primes below 10000. If $p < 10000$, then $L^c(p) < 1.5\log_2 p + 1$ unless p is 3847 or 5903 (Algorithm PRAC of Section 7 shows that $L(p) < 1.5\log_2 p + 1$ even when p is 3847 or 5903). The function $L^c(p)$ seems very smooth; for example, $L^c(p) = 19$ or 20 if $6053 < p < 10000$. In all cases an optimal r existed for which the largest partial quotient of $(p-r)/p$ was 3 or less. Table 3 summarizes the values of $L^c(p)$. The column titled a_{k+1} refers to Conjecture 13. TBD - extend it.

k	a_{k+1}	Minimum p with $L^c(p) = k$	Maximum p with $L^c(p) = k$	L_k	$F_{k+2} - F_{k-3}$	Number of p with $L^c(p) = k$
1	2	2	2	1	3	1
2	3	3	3	3	2	1
3	4	5	5	4	5	1
4	5	7	7	7	7	1
5	9	11	13	11	12	2
6	13	17	19	18	19	2
7	17	23	31	29	31	3
8	24	37	47	47	50	4
9	40	53	89	76	81	8
10	56	83	131	123	131	7
11	81	113	233	199	212	16
12	115	197	337	322	343	21
13	185	331	547	521	555	26
14	267	421	883	843	898	42
15	386	739	1597	1364	1453	74
16	551	1087	2351	2207	2351	96
17	882	1663	3739	3571	3804	137
18	1273	2671	6053	5778	6155	209
19	1849	3847	9791	9349	9959	320
20	2640	5903	≥ 9973	15127	16114	≥ 258

TABLE 3. Statistics on $L^c(p)$ for prime $p < 10000$

The following conjectures give an upper bound on $L^c(n)$.

Conjecture 11. *For all sufficiently large n there exists $m = m(n)$ such that $1 \le m < n$ and $\gcd(m, n) = 1$ and all the partial quotients of n/m are 1, 2, or 3.*

Partial justification. See [2][5][7, exercise 3.3.4–31]. ■

Lemma 12. *If $n/m = /\!\!/x_1, x_2, \ldots, x_k/\!\!/$ where $1 \le m \le n$ and $\gcd(m, n) = 1$, then $L^c(n, m) = x_1 + \cdots + x_k - 1$.*

Proof. Observe that $(n - m)/m = /\!\!/x_1 - 1, x_2, \ldots, x_k/\!\!/$. ■

Conjecture 13. *Suppose that*

$$m/n = /\!\!/ 0,\, x_1,\, x_2,\, \ldots,\, x_k /\!\!/ \tag{6.1}$$

where $\gcd(m, n) = 1$ *and* $0 \le m \le n$. *Designate* $s = x_1 + \cdots + x_k$. *If each* x_i *is* 1, 2, *or* 3, *then* $n \ge a_s$ *where*

$$\begin{aligned}
a_{4j} &= 8U_{j-1}(5,1) + (37V_{j-1}(5,1) + 7 - 3(-1)^j)/21,\\
a_{4j+1} &= U_{j+1}(5,1),\\
a_{4j+2} &= 18U_{j-1}(5,1) + (85V_{j-1}(5,1) + 70 + 30(-1)^j)/21,\\
a_{4j+3} &= U_{j+1}(5,1) + (2V_{j+1}(5,1) + 7 - 3(-1)^j)/7,
\end{aligned}$$

except that $a_2 = 2$, $a_6 = 9$, *and* $a_{10} = 40$.

Justification. Given s, choose n as small as possible subject to $1 \le x_i \le 3$ for $1 \le i \le k$ and $k > 0$; we want to show that this minimal n is a_s. Given s and n, choose k as small as possible. Given s, n, and k, choose $x_1, \ldots, x_k$ with as few 2's as possible.

Any permutation of $x_1, \ldots, x_k$ will leave k, s, and the number of 2's unchanged. Motzkin and Straus [13][7, exercise 4.5.3–37] show that the minimum n will occur when

$$x_1 \le x_k \le x_3 \le x_{k-2} \le x_5 \le \cdots \le x_6 \le x_{k-3} \le x_4 \le x_{k-1} \le x_2.$$

For $x = 1, 2, 3$, define a matrix $M_x = \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}$. Since m/n is in lowest terms, equation (13) is equivalent to the matrix equation [7, exercise 4.5.3–2]

$$\begin{pmatrix} m \\ n \end{pmatrix} = M_{x_0} M_{x_1} \cdots M_{x_k} \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \tag{6.2}$$

If M and N are 2×2 matrices with nonnegative coefficients, define $M \ge N$ if each entry in M is at least as large as the corresponding entry in N; define $M > N$ if all entries are larger. Straightforward calculations show that

$$\begin{aligned}
M_1 M_1 &= \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \ge \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix} = M_2,\\
M_1 M_2 &= \begin{pmatrix} 3 & 1 \\ 2 & 1 \end{pmatrix} \ge \begin{pmatrix} 3 & 1 \\ 1 & 0 \end{pmatrix} = M_3,\\
M_2 M_1 &= \begin{pmatrix} 3 & 2 \\ 1 & 1 \end{pmatrix} \ge \begin{pmatrix} 3 & 1 \\ 1 & 0 \end{pmatrix} = M_3,\\
M_3 M_3 M_3 M_3 M_3 &= \begin{pmatrix} 360 & 109 \\ 109 & 33 \end{pmatrix} > \begin{pmatrix} 345 & 91 \\ 91 & 24 \end{pmatrix} = M_3 M_1 M_3 M_1 M_3 M_1 M_3.
\end{aligned}$$

If two adjacent x_i's are both 1, then the inequality $M_1 M_1 > M_2$ shows we can replace both 1's by a 2 without increasing n in (6.2); this contradicts the minimality of k. Likewise there cannot be a 1 adjacent to a 2. There cannot be five adjacent 3's, since we could replace them by four 3's and three 1's while preserving $x_1 + \cdots + x_k$ and reducing n. The inequality

$$2M_2 M M_2 - M_1 M M_3 - M_3 M M_1 = 2\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} M \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \ge 0$$

for any $M \geq 0$ likewise shows that we have at most one 2 in our minimal solution; otherwise we could replace them by a 1 and a 3 in some order while preserving k and without increasing n.

TBD. See file conj.13

Next let k vary, while also varying the number of x_i equal to 1, the number of x_i equal to 2, and the number of x_i equal to 3. A program did this for $s \leq 45$. For each s, the minimum denominator equaled a_s. Except for $s = 2, 6, 10$, the minimum occurred when $\lfloor (s+2)/4 \rfloor$ of the x_i were equal to three and the rest were equal to one. The other minima were $1/2 = /\!\!/0,\ 2/\!\!/$, $7/9 = /\!\!/0,\ 1,\ 3,\ 2/\!\!/$, and $31/40 = /\!\!/0,\ 1,\ 3,\ 2,\ 3,\ 1/\!\!/$. ∎

Corollary 14. *Assume Conjectures 11 and 13. Then* $L^{c}(n) \leq 4\log_{\phi} n + O(1)$ *where* $\phi = (5+\sqrt{21})/2$. *(Note that* $4\log_{\phi} n \sim 1.77\log_2 n$.*)*

Proof. By adjusting the constant $O(1)$, if necessary, we may assume that n is large enough to satisfy the requirements of Conjecture 11. Select m satisfying the conclusion of Conjecture 11. Let $s = x_1 + \cdots + x_k$. where $m/n = /\!\!/0,\ x_1,\ \ldots,\ x_k/\!\!/$

A calculation gives $\phi + \phi^{-1} = 5$. By (1.1) and Conjecture 13, there exists $c > 0$ such that $a_i \geq c\phi^{i}/4$ for all $i \geq 1$. From $n \geq a_s \geq c\phi^{s/4}$ follows $s \leq 4\log_{\phi}(n/c)$. But $L^{c}(n) \leq L^{c}(n, m) = s - 1$. ∎

7. A Practical Algorithm

The results in the last section may be of theoretical interest. But we have not specified how to select $r(n)$. We do not know whether a good choice exists, although we believe that it does.

Let $\alpha = (1+\sqrt{5})/2$. The choice $r = \text{round}(n/\alpha)$ ensures that the first several partial quotients of $r/(n-r)$ will be 1. If the other partial quotients are also small, then this choice will be near optimal, but if they are high then this choice of r can be poor. For example, if $n = 151$, then $r = \text{round}(93.323\ldots)$ leads to $r/(n-r) = 93/58 = /\!\!/1,\ 1,\ 1,\ 1,\ 1,\ 11/\!\!/$. Therefore $L^{c}(151,\ 93) = 16$ whereas $L^{b}(151) = 13$. (The bad chain is 0, 1, 2, 3, 5, 8, 13, 21, 34, 47, 60, 73, 86, 99, 112, 125, 138, 151, in which most successive terms differ by 13.) Similarly $4476/2767 = /\!\!/1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 1,\ 2,\ 81/\!\!/$, so $L^{c}(7243,\ 4476) = 90$ whereas $L^{b}(7243) = 24$.

We can do a partial or exhaustive search for r, selecting the best value found. This is reasonable if one must compute $X_n(P)$ for several different P while n remains constant (e.g. [4][11]). But we have no assurance of success, and the search time can become expensive.

Instead we modify CFRC to avoid bad behavior. The trouble occurs only if the partial quotients are large, so we introduce additional steps which preserve (5.1) and which can be used when $d > 4e$ or $e > 4d$ (the constant 4 is subject to experimentation). For example, if $d > 4e$,

If $d \equiv e \pmod 2$, replace $(a,\ b,\ d,\ A,\ B) \leftarrow (2a,\ a+b,\ (d-e)/2,\ X_2(A),\ f(A,\ B,\ C))$.
If $d \equiv 0 \pmod 2$, replace $(a,\ d,\ A,\ C) \leftarrow (2a,\ d/2,\ X_2(A),\ f(A,\ C,\ B))$.
If $e \equiv 0 \pmod 2$, replace $(b,\ e,\ B,\ C) \leftarrow (2b,\ e/2,\ X_2(B),\ f(C,\ X_{-1}(B),\ A))$.

If these are applied with $n = 151$ and $r = 93$, then the Lucas chain becomes 0, 1, 2, 3, 5, 8, 13, 21, 26, 47, 52, 99, 151, a chain of length 11 rather than 16. By Theorem 8, the new chain is optimal. The transformations lead to a chain of length 21 for 7243 using $r = 4476$. This is not optimal since $L^c(7243) = L^c(7243,\, 2776) = 20$, but is nearly so since Theorem 8 shows that $L(7243) \geq 19$.

Transformations based on $d \pmod 3$ and $e \pmod 3$ can also be used. They are incorporated into the following algorithm for $X_n(P)$. Algorithm PRAC permits composite n, using (1.7) as necessary.

Algorithm PRAC(n, P)
Cmt. Return $X_n(P)$, assuming that $n \geq 0$.
if $n = 0$ **then return** X_0
$(A,\, d) \leftarrow (P,\, n)$
while $d \neq 1$ **do**
  Cmt. Know that $A = X_{n/d}(P)$; desire $X_n(P) = X_d(A)$.
  Let p be a prime factor of d; if none known or if d is prime then $p \leftarrow d$.
  $r \leftarrow (d/p)\text{round}(p/\alpha)$ where $\alpha = (1+\sqrt{5})/2$
  Cmt. If a better value for $r(p)$ is known, use it instead.
  $(d,\, e,\, B,\, C) \leftarrow (r,\, d-r,\, A,\, X_0)$
  while $d \neq e$ **do**
    Cmt. Invariant (5.1) holds here for some a and b.
    if $d < e$ **then** $(d,\, e,\, A,\, B,\, C) \leftarrow (e,\, d,\, B,\, A,\, X_{-1}(C))$
    Do the first line of Table 4 whose condition qualifies.
  end while
  $A \leftarrow f(A,\, B,\, C)$
end while
return A
end PRAC

If $\gcd(n,\, r) = 1$ and none of the partial quotients of $r/(n-r)$ exceed 3, then Algorithms CFRC and PRAC are equivalent. Most transformations in Table 4 were selected because they preserve (5.1) and decrease $d + e$ quickly if big partial quotients do occur. (Variables a and b do not appear in PRAC, but it is easy to restore them.) Transformation 9 is a catchall for use when $d > 4e$ and $e \equiv 0 \pmod 6$; an improved transformation 9 would be welcome. Transformations 1 and 2 are look aheads, intended to save a evaluation over applying transformation 3 followed by 8 or 9, respectively. Column "Cost" gives the number of evaluations required by each transformation (equivalently, the number of new terms to be inserted into the associated Lucas chain). Column "Drop" shows the minimum factor by which each transformation reduces $d + e$. Column "Usage" shows how many times each transformation was used when building Tables 5 and 6 of Section 8.

We claim that the remaining cost of Algorithm PRAC never exceeds $4\log_2(d+e)$ at the start of the inner while loop, and that it never exceeds $4\log_2 d$ at the start of the outer while

No.	Condition	Action(s)	Cost	Drop	Usage
1	$d \le 1.25e$ and $d \equiv -e \pmod 3$	$(d,\, e) \leftarrow ((2d-e)/3,\, (2e-d)/3)$ $T \leftarrow f(A,\, B,\, C)$ $(A,\, B) \leftarrow (f(T,\, A,\, B),\, f(T,\, B,\, A))$	3	3	41289
2	$d \le 1.25e$ and $d \equiv e \pmod 6$	$d \leftarrow (d-e)/2$ $(A,\, B) \leftarrow (X_2(A),\, f(A,\, B,\, C))$	2	2	5517
3	$d \le 4e$	$d \leftarrow d-e$ $(B,\, C) \leftarrow (f(A,\, B,\, C),\, X_{-1}(B))$	1	5/4	1654399
4	$d \equiv e \pmod 2$	$d \leftarrow (d-e)/2$ $(A,\, B) \leftarrow (X_2(A),\, f(A,\, B,\, C))$	2	2	72970
5	$d \equiv 0 \pmod 2$	$d \leftarrow d/2$ $(A,\, C) \leftarrow (X_2(A),\, f(A,\, C,\, B))$	2	5/3	69301
6	$d \equiv 0 \pmod 3$	$d \leftarrow d/3 - e$ $(T_1,\, T_2) \leftarrow (X_2(A),\, f(A,\, B,\, C))$ $A \leftarrow f(T_1,\, A,\, A)$ $(B,\, C) \leftarrow (f(T_1,\, T_2,\, C),\, X_{-1}(B))$	4	3	14385
7	$d \equiv -e \pmod 3$	$d \leftarrow (d-2e)/3$ $T_1 \leftarrow f(A,\, B,\, C)$ $(A,\, B) \leftarrow (X_3(A),\, f(T_1,\, A,\, B))$	4	3	13180
8	$d \equiv e \pmod 3$	$d \leftarrow (d-e)/3$ $(T_1,\, T_2) \leftarrow (f(A,\, B,\, C),\, f(A,\, C,\, B))$ $(A,\, B,\, C) \leftarrow (X_3(A),\, T_1,\, T_2)$	4	5/2	2377
9	$e \equiv 0 \pmod 2$	$e \leftarrow e/2$ $(B,\, C) \leftarrow (X_2(B),\, f(C,\, X_{-1}(B),\, A))$	2	1	3161

TABLE 4. Transformations used by Algorithm PRAC

loop. These are evident from the Cost and Drop columns of Table 4 if transformation 9 is never required; indeed its cost won't exceed $\log_{5/4}(d+e) \sim 3.1 \log_2(d+e)$. Suppose that transformation 9 is used $m > 0$ successive times. The starting values $(d_1,\, e_1)$ and the ending values $(d_2,\, e_2)$ of $(d,\, e)$ will satisfy $d_1 = d_2$, $e_1 = e_2 \cdot 2^m$, $d_1 > 4e_1$, and $e_2 \equiv 3 \pmod 6$. Now only transformations 4 and 5 can qualify. Apply them k times until $d \le 4e$ but at most m times. Let $(d_3,\, e_3)$ be the final values of $(d,\, e)$. Then $d_3 \le d_2/2^k$ and $e_3 = e_2$. If $d_3 \le 4e_3$, then

$$\frac{d_1 + e_1}{d_3 + e_3} \ge \frac{5e_1}{5e_3} = 2^m.$$

If instead $k = m$, then

$$\frac{d_1 + e_1}{d_3 + e_3} \ge \frac{d_2 + e_2 \cdot 2^m}{d_2/2^k + e_2} = 2^m.$$

In both cases we have reduced the value of $d+e$ by a factor at least 2^m at a cost of $2(m+k) \le 4m$. Consequently the total cost of Algorithm PRAC will not exceed $4 \log_2 n$ if $n > 0$. An improved estimate is possible, since our selection of r ensures that transformation 3 will be used several times initially.

Let $L^{\mathrm{p}}(n)$ be the length of the Lucas chain associated with Algorithm PRAC, using $p = d$ and $r = \mathrm{round}(d/\alpha)$ in the outer while loop. The values of $L^{\mathrm{p}}(p)$ for prime p below 200 appear in Table 2.

While debugging Algorithm PRAC, one can use invariant (5.1) to insert a check. Replace the assignment $A \leftarrow f(A, B, C)$ by:

$T \leftarrow f(A, B, C)$
if $X_2(A) \neq f(T, C, X_2(B))$ **then** signal an error condition
$A \leftarrow T$.

The test is valid because if $A = X_a$, $B = X_b$, and $C = X_{a-b}$ for some integers a and b, then $T = X_{a+b}$ and

$$X_2(A) = X_{2a} = f(X_{a+b}, X_{a-b}, X_{2b}) = f(T, C, X_2(B)).$$

For particular functions f, it may possible to optimize this check. For example, when $X_n(P) = V_n(P)$, the check can be shortened to $(C-2)(T-2) \neq (A-B)^2$. The author concedes that inclusion of this check revealed subtle errors in an early version of his multiple precision routines.

If n is a multiple precision integer, then the tests on $d \pmod 3$ and $e \pmod 3$ within PRAC may be too expensive. A simplified version of PRAC uses only transformations 3, 4, 5, and 9. The worst cost estimate of $4 \log_2 n$ still applies.

8. Comparative Performance of Algorithms

Table 5 summarizes the total costs of each algorithm for prime p below 10^4 and below 10^6, except that CFRC was run only for $p \leq 10000$. The results are compared to the lower bound predicted by Theorem 8.

Algorithm	Total cost for $p < 10^4$	Total cost for $p < 10^6$	Excess over Theorem 8	Least squares fit
Theorem 8	21141	2114698		$1.446 \log_2 p + 0.409$
L^{b}	26636	2755571	30.3%	$1.992 \log_2 p - 1.455$
L^{t}	25874	2679141	26.7%	$1.944 \log_2 p - 1.541$
L^{c}	21558		2.0%	$1.536 \log_2 p - 0.304$
L^{p}	22204	2278430	7.7%	$1.628 \log_2 p - 0.856$

Table 5. Comparative performance for prime $p < 10^6$

The results of Algorithm PRAC are not as good as those of CFRC, but they are nearly as good. Both PRAC and CFRC do considerably better than the binary and binary-ternary methods. PRAC can be programmed as is, whereas CFRC does not specify how to choose $r(n)$. PRAC beats CFRC for $n = 113$, 439, 479, 809, 2029, 2039, 2707, 2819, 3023, 3469, 3847, 4493, 4561, 4567, 4637, 4703, 4909, 4967, 5333, 5903, 6737, 6779, 7459, 7643, 7927, 7993, and 8629.

A simple variation of Algorithm PRAC which tried eight values of r, corresponding to

$$r = (d/p)\mathrm{round}(p \cdot /\!\!/0, 1, 1, 1, \ldots /\!\!/),$$
$$r = (d/p)\mathrm{round}(p \cdot /\!\!/0, 1, 2, 1, \ldots /\!\!/),$$
$$r = (d/p)\mathrm{round}(p \cdot /\!\!/0, 1, 1, 2, \ldots /\!\!/),$$
$$\vdots$$

$L^{\mathrm{b}}(p) - L^{\mathrm{p}}(p)$	-4	-3	-2	-1	0	1	2
Number of p	17	40	54	111	145	328	691
First p	103529	63997	52813	6151	2	13	67
$L^{\mathrm{b}}(p) - L^{\mathrm{p}}(p)$	3	4	5	6	7	8	9
Number of p	1801	4456	12855	25880	24255	7514	351
First p	131	521	1597	8219	25463	67103	263513

TABLE 6. Comparison of Algorithm PRAC and binary method for prime $p < 10^6$

had a total cost of 21541 for $p < 10^4$, slightly better than CFRC. TBD - how many is best?

The worst case constant of proportionality derived for PRAC is not as good as that of the binary method, and indeed PRAC occasionally does worse. Table 6 tallies the difference $L^{\mathrm{p}}(p) - L^{\mathrm{b}}(p)$ for the 78498 prime p below 10^6.

If n is odd, and we set $r \leftarrow d - 1$ in PRAC (so that $e = 1$ throughout the inner loop and only transformations 3, 4, 5 are used), we discover a right-to-left binary method. For 101, the corresponding Lucas chain is 0, 1, 2, 3, 4, 5, 8, 11, 16, 27, 32, 37, 64, 101. The cost of this algorithm is $2 \lg n$, which is slightly worse than the binary method of Section 2. When $n \equiv \pm 3 \pmod 8$, as in this case, we can remove 4 from the chain. The cost of this method drops by half if the values of $X_{2^k}(P)$ are available in a table for $k \leq \lg n$.

9. Open Problems

Here are some open problems in this field:

Problem 1. Find an easily computed function $r(n)$ and a constant $c < 2$ such that Algorithm CFRC or PRAC (or a variation thereof) requires at most $c \lg n$ evaluations of f to compute $X_n(P)$, for all sufficiently large n. Possibly most of the transformations in Table 4 can be replaced by a generalized transformation based upon the least prime not dividing e.

Problem 2. Does $\lim_{p \to \infty} L(p)/\ln p$ exist, if p is restricted to prime values? Are this and $\limsup L(n)/\ln n$ equal to $1/\ln((1+\sqrt{5})/2)$?

Problem 3. The sequence 0, 1, 2, 3, 4, 7, 10, 11, 9 is not a Lucas chain for 9, since it is not ascending. It cannot be rearranged to form a Lucas chain for 9 (although $9 = 7 + 2$, the difference $7 - 2$ is missing). It does represent a way to compute $X_9 = f(X_{10}, X_{-1}(X_1), X_{11})$. Does there exist a positive integer n such that X_n can be computed using (1.6) fewer than $L(n)$ times?

Problem 4. Strengthen Theorem 8. (Neither CFRC nor PRAC does as well as Theorem 8 allows for $p =$ 197, 421, 461, 491, 509, 739, 751, 757, 761, 769, 797, 811, 821, 823, 827, 829, 839. The Lucas chains

$$0, 1, 2, 3, 5, 7, 12, 19, 24, 43, 67, 110, 177, 244, 421 \qquad \text{and}$$
$$0, 1, 2, 3, 5, 7, 12, 19, 24, 43, 67, 110, 177, 287, 464, 751$$

show that CFRC and PRAC are suboptimal for $p = 421$ and $p = 751$.)

Problem 5. One can evaluate $V_n(P)$ without Lucas chains. For example, $V_5(P) = (P - 2)(P^2 + P - 1)^2 + 2$ uses $3 = L(5)$ multiplications. Do all polynomial chains for $V_n(P)$ require at least $L(n)$ multiplications (see [7, p. 475ff.] for definitions)?

Problem 6. Estimate the density of composite n for which there exist j, $k > 1$ with $n = jk$ and $L(n) < L(j) + L(k)$.

10. Using Lucas Chains to Evaluate other Lucas Functions

Sometimes one needs both $U_n = U_n(P, 1)$ and $V_n = V_n(P, 1)$. If n is negative, then $V_n = V_{-n}$ and $U_n = -U_{-n}$. If n is even, then $V_n = V_{n/2}^2 - 2$ and $U_n = U_{n/2}V_{n/2}$. So we may assume that n is odd and positive.

If $\Delta = P^2 - 4$ is invertible, use the binary or binary-ternary method to compute both V_n and V_{n-1}. Then $U_n = (V_1V_n - 2V_{n-1})/\Delta$. Otherwise the identities

$$\begin{aligned} U_{i+j} &= U_iV_j - U_{i-j} = U_jV_i + U_{i-j}, \\ V_{i+j} &= V_iV_j - V_{i-j} \end{aligned}$$

express U_{i+j} and V_{i+j} in terms of U_i, V_i, U_j, V_j, U_{i-j}, and V_{i-j}. If (3.1) is a Lucas chain for n, then we can successively compute U_{a_i} and V_{a_i} for $i = 0, 1, \ldots, r$. At most $2L(n)$ multiplications are needed to get U_n and V_n. Actually, we need calculate U_{a_i} only for some of the a_i, so the true cost is somewhere between $L(n)$ and $2L(n)$ multiplications.

If Q is arbitrary, then

$$\begin{aligned} V_{i+j}(P, Q) &= V_i(P, Q)V_j(P, Q) - Q^jV_{i-j}(P, Q) \\ &= V_i(P, Q)V_j(P, Q) - Q^iV_{j-i}(P, Q). \end{aligned}$$

If $n \geq 0$, then $3L(n)$ multiplications suffice to compute both $V_n(P, Q)$ and Q^n for arbitrary Q. If $U_n(P, Q)$ is also required, then one can use

$$\begin{aligned} U_{i+j}(P, Q) &= U_i(P, Q)V_j(P, Q) - Q^jU_{i-j}(P, Q) \\ &= V_i(P, Q)U_j(P, Q) - Q^iU_{j-i}(P, Q). \end{aligned}$$

In this case the binary method is convenient and efficient, since all differences $i - j$ will be -1, 0, or $+1$, causing $Q^jU_{i-j}(P, Q)$ and $Q^iU_{j-i}(P, Q)$ to be $\pm Q^j$, $\pm Q^i$, or 0.

11. Summary

A Lucas chain for n is a sequence of subscripts used to compute $V_n(P)$ or other sequence satisfying (1.6). The shortest Lucas chain for n has length $L(n)$. The binary method shows that $L(n) \leq L^{\mathrm{b}}(n) \leq 2\lg n$. Theorem 8 shows that $L(p) > 1.44\log_2 p$ for all sufficiently large prime p. For large random n, Algorithm PRAC seems to generate a Lucas chain of length about $1.6\log_2 n$, but its length occasionally exceeds $2\log_2 n$. Algorithm CFRC seems to do better, but it is incomplete and can do much worse in its worst case. It appears that $L(n) < 1.5\log_2 n + 1$ for all n, but it seems hard even to prove that $\limsup L(n)/\lg n < 2$. TBD check to 10^6

References

1. Milton Abramowitz and Irene A. Stegun (eds.), *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publications, Inc., New York, 1965.
2. I. Borosh, *Rational continued fractions with small partial quotients*, Abstract 731–10–29., Notices Amer. Math. Soc. **23** (1976), A–52.

3. John Brillhart, D. H. Lehmer, and J. L. Selfridge, *New primality criteria and factorizations of* $2^m \pm 1$, Math. Comp. **29** (1975), 620–647.
4. J. P. Buhler, R. E. Crandall, and M. A. Penk, *Primes of the form* $n! \pm 1$ *and* $2 \cdot 3 \cdot 5 \dots p \pm 1.$, Math. Comp. **38** (1982), 639–643.
5. T W. Cusick, *Continuants with bounded digits*, Mathematika **24** (1977), 166–172.
6. Richard K. Guy, *Unsolved Problems in Intuitive Mathematics,* Unsolved Problems in Number Theory, Vol. I (P. R. Halmos, ed.), Springer-Verlag, New York, 1981.
7. Donald E. Knuth, *Seminumerical Algorithms* The Art of Computer Programming, Vol. II, 2nd ed., Addison-Wesley, Reading, Mass., 1981.
8. D. H. Lehmer, *An extended theory of Lucas' functions*, Ann. of Math. **31** (1930), 419–448.
9. D. H. Lehmer, *Computer technology applied to the theory of numbers*, Studies in Number Theory (Math. Assoc. Amer. Studies in Math. Vol. 6) (W. J. LeVeque, ed., ed.), 1969, pp. 117–151.
10. Peter. L. Montgomery, *Problem 1202*, Math. Mag. **58** (1985), 300–301.
11. Peter L. Montgomery, *Speeding the Pollard and elliptic curve methods of factorization*, Math. Comp. **48** (1987), 243–264.
12. Michael A. Morrison., *A note on primality testing using Lucas sequences*, Math. Comp. **29** (1975), 181–182.
13. T. S. Motzkin and E. G. Straus, *Some combinatorial extremum problems*, Proc. Amer. Math. Soc. **7** (1956), 1014–1021.
14. Carl Pomerance, J. L. Selfridge, snf Samuel S. Wagstaff, Jr., *The pseudoprimes to* $25 \cdot 10^9$, Math, Comp. **35** (1980), 1003–1026.
15. H. C. Williams, *A* $p+1$ *method of factoring*, Math. Comp. **39** (1982), 225–234.
16. M. C. Wunderlich, *A performance analysis of a simple prime-testing algorithm*, Math. Comp. **40** (1983), 709–714.
17. Andrew C. Yao and Donald E. Knuth, *Analysis of the subtractive algorithm for greatest common divisors*, Proc. Nat. Acad. Sci. USA **72** (1975), 4720–4722.

Notes to Printer

An $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX version of this manuscript is available.

The "$/\!\!/ \dots /\!\!/$" notation is intended to match that in [7, pp. 339ff.]. The macros I used to generate these symbols are crude.

Table 2 can be split into two parts if it does not fit on one page.

Notes to Referee

I did not try very hard to prove Conjecture 13. Possibly one can first prove a corresponding result when the largest partial quotient is 2. Problem 3 of Section 9 may also have an easy solution.

I coined the term "Lucas chain". Do you approve of this name?

I did not try to get a comprehensive list of low values of j and k for which $L(jk) < L(j) + L(k)$. The examples following Corollary 3 were found by hand.

The solution to [7, exercise 3.3.4–31] references a paper by I. Borosh and H. Niederreiter related to Conjecture 11. I wrote Borosh regarding this in November, 1982, but got no response. TBD - it is in BIT 1983

This paper seems long. I could present PRAC early, using (5.1) as a justification, but the article would seem too dry. Are there some portions which you recommend I remove?

I welcome your verifying some numbers in the tables.

Is the double usage of $L(n)$ and L_n confusing? If so, please suggest an alternate notation.

Department of Mathematics, University of California, Los Angeles, CA 90024–1555
E-mail address: pmontgom@math.ucla.edu