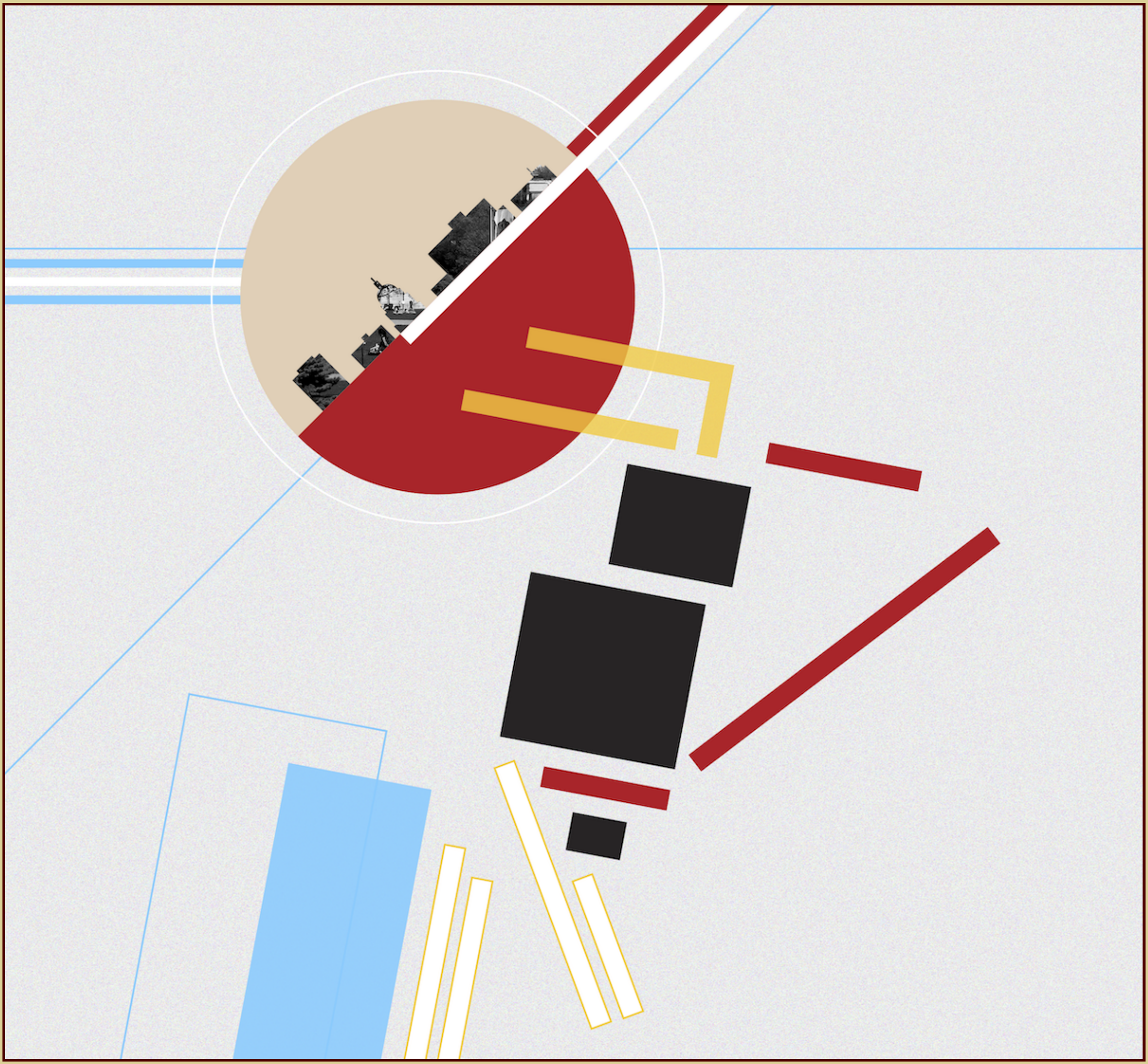
THE OPEN BOOK SERIES 2

ANTS XIII
Proceedings of the Thirteenth
Algorithmic Number Theory Symposium

Fast tabulation of challenge pseudoprimes

Andrew Shallue and Jonathan Webster



msp

THE OPEN BOOK SERIES **2** (2019)
Thirteenth Algorithmic Number Theory Symposium
dx.doi.org/10.2140/obs.2019.2.411

Fast tabulation of challenge pseudoprimes

Andrew Shallue and Jonathan Webster

We provide a new algorithm for tabulating composite numbers which are pseudoprimes to both a Fermat test and a Lucas test. Our algorithm is optimized for parameter choices that minimize the occurrence of pseudoprimes, and for pseudoprimes with a fixed number of prime factors. Using this, we have confirmed that there are no PSW-challenge pseudoprimes with two or three prime factors up to 2^{80}. In the case where one is tabulating challenge pseudoprimes with a fixed number of prime factors, we prove our algorithm gives an unconditional asymptotic improvement over previous methods.

1. Introduction

Pomerance, Selfridge, and Wagstaff famously offered \$620 for a composite n that satisfies

(1) $2^{n-1} \equiv 1 \pmod{n}$ so n is a base-2 Fermat pseudoprime,

(2) $(5 \mid n) = -1$ so n is not a square modulo 5, and

(3) $F_{n+1} \equiv 0 \pmod{n}$ so n is a Fibonacci pseudoprime,

or to prove that no such n exists. We call composites that satisfy these conditions PSW-challenge pseudoprimes. In [PSW80] they credit R. Baillie with the discovery that combining a Fermat test with a Lucas test (with a certain specific parameter choice) makes for an especially effective primality test [BW80]. Perhaps not as well known is Jon Grantham's offer of \$6.20 for a Frobenius pseudoprime n to the polynomial $x^2 - 5x - 5$ with $(5 \mid n) = -1$ [Gra01]. Similar to the PSW challenge, Grantham's challenge number would be a base-5 Fermat pseudoprime, a Lucas pseudoprime with polynomial $x^2 - 5x - 5$, and satisfy $(5 \mid n) = -1$. Both challenges remain open as of this writing, though at least in the first case there is good reason to believe infinitely many exist [Pom84].

The largest tabulation to date of pseudoprimes of similar type is that of Gilchrist [Gil13], who found no Baillie-PSW pseudoprimes (a stronger version of the PSW challenge) up to $B = 2^{64}$. After first tabulating 2-strong pseudoprimes [Fei13; Nic12] using an algorithm due to Pinch [Pin00], he applied the strong

Shallue was supported in part by Illinois Wesleyan University's Artistic and Scholarly Development grant and Webster was supported in part by Butler University's Holcomb Awards Committee.
MSC2010: primary 11Y11, 11Y16; secondary 11A41, 68W40.
Keywords: pseudoprimes.

Lucas test using the code of Nicely [Nic12]. Taking inspiration from tabulations of strong pseudoprimes to several bases [Jae93; Ble96; JD14; SW17], our new idea is to treat the tabulation as a two-base computation: a Fermat base and a Lucas base. In this way we exploit both tests that make up the definition.

Specifically, we improve upon [Pin00] in three ways:

- GCD computations replace factorizations of $b^n - 1$.
- Sieving searches are done with larger moduli.
- Fewer preproducts are constructed.

Other notable attempts to find a PSW-challenge number involve construction techniques that result in a computationally infeasible subset-product problem [GA99; CG03]. The first of such attempts would have also found the number requested at the end of [Wil77] which is simultaneously a Carmichael number and a (P, Q)-Lucas pseudoprime for all pairs (P, Q) with $5 = P^2 - 4Q$ and $(5 \mid n) = -1$.

The new algorithm presented constructs n by pairing primes p with admissible preproducts k. In Section 6 we provide an unconditional proof of the running time. Unfortunately, the provable running time gets worse as the number of primes dividing k increases. Specifically, we prove the following.

Theorem 1. *There exists an algorithm which tabulates all PSW-challenge pseudoprimes up to B with t prime factors, while using $\tilde{O}(B^{1-1/(3t-1)})$ bit operations and space for $O(B^{(3t-2)/(4t-2)})$ words.*

The running time improves, under a heuristic assumption that factoring plays a minimal role, to $\tilde{O}(B^{1-1/(2t-1)})$ bit operations.

No PSW-challenge pseudoprimes with two or three prime factors exist up to $B = 2^{80}$.

For the computation performed we chose 2 as the Fermat base and $(1, -1)$ as the Lucas base, but the algorithm as designed can handle arbitrary choices.

The rest of the paper is organized as follows. Section 2 establishes key definitions and notation, while Section 3 provides the theoretical underpinnings of the algorithm. The algorithm is presented in Section 4 along with a proof of correctness. The running time is analyzed in Sections 5 and 6. We conclude the paper with comments on our computation with $B = 2^{80}$.

2. Definitions and notation

A *base-b Fermat pseudoprime* is a composite n with $\gcd(n, b) = 1$ that satisfies the congruence $b^{n-1} \equiv 1 \pmod{n}$.

Lucas sequences have many equivalent definitions. We state a few important ones and let the reader consult standard sources such as [Leh30] for a more thorough treatment. Let $P, Q \in \mathbb{Z}$ and $\alpha, \bar{\alpha}$ be the distinct roots of $f(x) = x^2 - Px + Q$, with $D = P^2 - 4Q$ the discriminant. Then the Lucas sequences are

$$U_n(P, Q) = \frac{\alpha^n - \bar{\alpha}^n}{\alpha - \bar{\alpha}} \quad \text{and} \quad V_n(P, Q) = \alpha^n + \bar{\alpha}^n.$$

Equivalently, we may define these as recurrence relations, where

$$U_0(P, Q) = 0, \quad U_1(P, Q) = 1, \quad \text{and} \quad U_n(P, Q) = PU_{n-1}(P, Q) - QU_{n-2}(P, Q),$$

and

$$V_0(P,Q)=2,\quad V_1(P,Q)=P,\quad \text{and}\quad V_n(P,Q)=PV_{n-1}(P,Q)-QV_{n-2}(P,Q).$$

We will use $\epsilon(n)=(D\,|\,n)$ for the Jacobi symbol and will frequently write U_n or V_n when the particular sequence is clear from context. It should be noted that the definition below guarantees that n is odd so that the Jacobi symbol is well-defined. Often U_n is referred to as the Lucas sequence with parameters P and Q, but both V_n and U_n are needed for the "double-and-add" method for computing U_n using $O(\log n)$ arithmetic operations. For a more modern take on this classic algorithm, see [JQ96].

A (P, Q)-*Lucas pseudoprime* is a composite n with $\gcd(n, 2QD)=1$ such that $U_{n-\epsilon(n)}\equiv 0 \pmod n$.

Definition 2. We call a composite n a (b, P, Q)-challenge pseudoprime if it is simultaneously a base-b Fermat pseudoprime, a (P, Q)-Lucas pseudoprime, and additionally satisfies $\epsilon(n)=-1$.

Note that $\epsilon(n)=-1$ means that D is not a square.

A PSW-challenge pseudoprime is then a $(2,1,-1)$-challenge pseudoprime in our notation. To get a Baillie-PSW pseudoprime, one replaces the Fermat test with a strong pseudoprime test and the Lucas test with a strong Lucas test. The Lucas parameters are chosen as $P=1$ and $Q=(1-D)/4$, where D is the first discriminant in the sequence $\{5,-7,9,-11,\dots\}=\{(-1)^k(2k+1)\}_{k\geq 2}$ for which $(D\,|\,n)=-1$.

Certain parameter choices should be avoided as they make the challenge much less interesting. Specifically, roots of unity create unwanted degenerate behavior. Thus we exclude $b=\pm 1$, and any (P, Q) for which the squarefree part of D is either -1 or -3, in addition to excluding D which are squares. The reason is that the only quadratic extensions of $\mathbb{Q}$ that contain roots of unity are those corresponding to the quadratic cyclotomic polynomials x^2+1 and $x^2\pm x+1$.

We use $\ell_b(n)$ when $\gcd(b,n)=1$ to denote the multiplicative order of b modulo n, i.e., the smallest positive integer such that $b^{\ell_b(n)}=1 \pmod n$. When $n=p$ is a prime, $\ell_b(p)\,|\,p-1$ by Lagrange's theorem since $p-1$ is the order of $(\mathbb{Z}/p\mathbb{Z})^\times$.

Given a prime p, there exists a least positive integer ω such that $U_\omega\equiv 0 \pmod p$. We call ω the rank of apparition of p with respect to the Lucas sequence (P, Q), and we denote it by $\omega(p)$. It is also well known that $U_{p-\epsilon(p)}\equiv 0 \pmod p$ and hence that $\omega(p)\,|\,p-\epsilon(p)$.

Throughout, we will use log to represent the natural logarithm.

The function $P(n)$ returns the largest prime factor of n, and for asymptotic analysis we often use $\widetilde{O}$, where $f=\widetilde{O}(g)$ means there are positive constants N, c such that $f(n)\leq g(n)(\log(4+g(n)))^c$ for nonnegative functions $f(n)$ and $g(n)$ and for all $n\geq N$ [vzGG03, Definition 25.8].

3. Algorithmic theory

The main idea of the tabulation comes from [Jae93; Ble96; JD14; SW17], but instead of tabulating pseudoprimes to many bases, we have just a Fermat base and a Lucas base. For the Fermat case we state known results for completeness, while for the Lucas case we state and prove the required results. We follow the notation in [SW17] when possible.

To find all (b, P, Q)-challenge pseudoprimes $n < B$, we construct n in factored form $n = p_1 p_2 \cdots p_{t-1} p_t$, where t is the number of prime divisors of n and $p_i \leq p_{i+1}$. We call $k = p_1 p_2 \cdots p_i$ for $i < t$ a preproduct. Section 3.1 states theorems limiting the number of preproducts that need to be considered. Section 3.2 shows that p_t may be found via a gcd computation when k is small and by a sieving search when k is large.

3.1. *Conditions on $n = wk$.*

We will frequently make use of the fact that if $\epsilon(n) = -1$ and $n = wk$ then $\epsilon(w) = -\epsilon(k)$ by the multiplicative property of the Jacobi symbol.

Proposition 3 [Ble96, Theorem 3.20]. *Let $k \geq 1$ be an integer and p a prime. If $n = kp^2$ is a Fermat pseudoprime to the base b then the following two conditions must be satisfied*:

(1) $b^{p-1} \equiv 1 \pmod{p^2}$.

(2) $b^{k-1} \equiv 1 \pmod{p^2}$.

Proposition 4. *Let $k \geq 1$ be an integer and p a prime. If $n = kp^2$ is a (P, Q)-Lucas pseudoprime with $\epsilon(n) = -1$ then the following two conditions must be satisfied*:

(1) $U_{p-\epsilon(p)} \equiv 0 \pmod{p^2}$.

(2) $U_{k-\epsilon(k)} \equiv 0 \pmod{p^2}$.

Proof. We start by noting that $\omega(p^2) \mid p\omega(p)$ and hence $\omega(p^2)$ divides $p(p - \epsilon(p))$ by the law of repetition [Leh30, Theorem 1.6]. In addition, $U_{n+1} \equiv 0 \pmod{n}$ by assumption so that $U_{n+1} \equiv 0 \pmod{p^2}$ and hence $\omega(p^2) \mid n + 1$. With p relatively prime to $n + 1$, it follows that $\omega(p^2)$ divides $\gcd(n + 1, p - \epsilon(p))$, and we conclude that $\omega(p^2)$ divides $p - \epsilon(p)$, which proves the first congruence.

For the second congruence, if $k = 1$ then $U_{k-\epsilon(k)} = U_0$ and the congruence is satisfied. In the case $k > 1$, we have $\omega(p^2)$ divides $n + 1 = kp^2 + 1 = kp^2 - \epsilon(k)$ and $p - \epsilon(p)$. Thus $\omega(p^2)$ divides

$$kp^2 - \epsilon(k) - k(p-1)(p+1) = kp^2 - \epsilon(k) - k(p^2 - 1) = k - \epsilon(k).$$

It follows that $U_{k-\epsilon(k)} \equiv 0 \pmod{p^2}$. □

In the case $b = 2$, these primes are known as Wieferich primes and in the $(1, -1)$ case they are known as Wall–Sun–Sun primes. The paper [CDP97] suggests the following heuristic argument to understand the rarity of these primes. Consider either $b^{p-1} - 1$ or $U_{p-\epsilon(p)}$ in a base-p representation. The constant coefficient is zero by Fermat's little theorem and its analogue. The coefficient on p needs to be 0 to satisfy the above congruence and we expect this to happen with probability $1/p$. Summing over the reciprocals of primes gives an expected count of such primes up to x as being on the order of $\log\log x$. For challenge pseudoprimes, both congruences would have to be met simultaneously. The corresponding count from the expected values is now a sum of $1/p^2$ and the infinite sum converges. So we expect the count to be finite and we know of no examples of this behavior.

Either the Fermat case or the Lucas case can individually be checked up to a bound B in $O(B^{1/2})$ time and such primes may be then tested against the other condition. In the very unlikely scenario that such a prime does exist, we refer the reader to Section 6 of [Pin00] in order to account for square factors dividing challenge pseudoprimes. Given how exceedingly rare we believe these are, we deal no further with square factors and assume a squarefree challenge pseudoprime.

Proposition 5. *Let $n = p_1 p_2 \cdots p_t$ be a (b, P, Q)-challenge pseudoprime,*

$$L = \operatorname{lcm}(\ell_b(p_1), \dots, \ell_b(p_t)) \quad \textit{and} \quad W = \operatorname{lcm}(\omega(p_1), \dots, \omega(p_t)).$$

Then $\gcd(L, W) \le 2$, $\gcd(n, L) = 1$, *and* $\gcd(n, W) = 1$.

Proof. We have $b^{n-1} \equiv 1 \pmod{p_i}$ and hence $n \equiv 1 \pmod{\ell_b(p_i)}$. We also have $U_{n+1} \equiv 0 \pmod{p_i}$ and hence $n \equiv -1 \pmod{\omega(p_i)}$. So $\ell_b(p_i) \mid (n-1)$ and $\omega(p_i) \mid (n+1)$ and this holds for all $p_i \mid n$. Therefore, $L \mid (n-1)$ and $W \mid (n+1)$. Then $\gcd(L, W) \le \gcd(n-1, n+1) \le 2$. Since n is relatively prime to both $n+1$ and $n-1$, the other two gcds are as claimed. □

We call a preproduct k *admissible* if the gcd statements in Proposition 5 are satisfied. The concept of admissibility is extremely useful in limiting the preproducts under consideration. As one example, very few primes p with $\epsilon(p) = 1$ are admissible, since in this case $\ell_b(p)$ and $\omega(p)$ both divide $p-1$, increasing the likelihood that $\gcd(\ell_b(p), \omega(p)) > 2$. In private correspondence, Paul Pollack gave a heuristic argument suggesting around $\log(x)$ such primes up to x.

3.2. *Conditions on p_t given k.*

Henceforth, we assume that $k = p_1 \cdots p_{t-1}$ and that k is admissible.

Proposition 6. *If $n = kp$ is a (b, P, Q)-challenge pseudoprime then p is a divisor of*

$$\gcd(b^{k-1} - 1, U_{k-\epsilon(k)}).$$

Proof. Recall that $b^{n-1} \equiv 1 \pmod{n}$ and $U_{n+1} \equiv 0 \pmod{n}$. We rewrite $n-1 = kp-1 = k(p-1)+k-1$. Since $\ell_b(p)$ divides $p-1$ and $n-1$ we conclude $\ell_b(p) \mid (k-1)$. Thus, $p \mid (b^{k-1} - 1)$.

Similarly $n+1 = kp - \epsilon(p)\epsilon(k) = k(p-\epsilon(p)) + k\epsilon(p) - \epsilon(p)\epsilon(k) = k(p-\epsilon(p)) + \epsilon(p)(k-\epsilon(k))$. Since $\omega(p)$ divides $p - \epsilon(p)$ and $n+1$ we conclude $\omega(p) \mid (k - \epsilon(k))$. Thus, $p \mid U_{k-\epsilon(k)}$. □

Proposition 7. *If $n = kp$ is a (b, P, Q)-challenge pseudoprime then*

$$p \equiv \begin{cases} k^{-1} \pmod{L}, \\ -k^{-1} \pmod{W}, \end{cases}$$

where

$$L = \operatorname{lcm}(\ell_b(p_1), \dots, \ell_b(p_{t-1})) \quad \textit{and} \quad W = \operatorname{lcm}(\omega(p_1), \dots, \omega(p_{t-1})).$$

Proof. Since $n = kp$ is a challenge pseudoprime, we have that $b^{kp-1} \equiv 1 \pmod{p_i}$, where p_i is any prime factor of k, and so $\ell_b(p_i) \mid (kp-1)$. Thus, $p \equiv k^{-1} \pmod{\ell_b(p_i)}$. We also know that $\omega(n+1) \equiv 0 \pmod{n}$, and hence that it is congruent to 0 modulo p_i. Thus, $\omega(p_i) \mid (kp+1)$ so that $p \equiv -k^{-1} \pmod{\omega(p_i)}$.

Now, $\ell_b(p_i) \mid (kp-1)$ for all $p_i \mid k$ if and only if $L \mid (kp-1)$. A similar statement holds for W, which completes the proof. □

4. Algorithm

Our basic strategy follows that found in [SW17]. Find all pseudoprimes with t prime factors for each $t \geq 2$ in turn. For a given t, we analyze all preproducts k with $t-1$ prime factors. The question for each preproduct is whether there exists a prime p that makes $n = kp$ a challenge pseudoprime. For small preproducts, this question can be answered with a gcd computation. For large preproducts, we instead use a sieve.

```
Algorithm 1: Tabulating squarefree challenge pseudoprimes.
   Input : bound $B$, positive integer $b \geq 2$, Lucas sequence parameters $(P, Q)$.
  Output : list of $n \leq B$ which are $(b, P, Q)$-challenge pseudoprimes.
Create an array of size $\sqrt{B}$ with entry $i$ containing the smallest prime factor of $i$;
for primes $p \leq \sqrt{B}$ do
    Compute $\ell_b(p)$, $\omega(p)$ and only keep prime $p$ if $\gcd(\ell_b(p), \omega(p)) \leq 2$;
    Update preproduct list;
    for new preproducts $k$ do
        if $k \leq X$ then
            do GCD step;
        else
            do Sieve step;
```

The above suggests storing all such primes up to $\sqrt{B}$ along with allowable preproducts, but space constraints would prohibit this strategy in practice. Construction of composite preproducts may be done with a combination of storing the 3-tuple $(p, \ell_b(p), \omega(p))$ for small primes and creating them on the fly for large primes, where the distinction is dependent upon space constraints. To efficiently create them, one may use an incremental sieve or a segmented sieve to generate factorizations of consecutive integers so that we may quickly compute $\ell_b(p)$ from the factorization of $p-1$ and $\omega(p)$ from the factorization of $p - \epsilon(p)$.

To tabulate Baillie-PSW pseudoprimes, one tabulates all pseudoprimes for each D in the sequence. Each discriminant performs a trial division so that successive computations will remove the next small prime from consideration, making the algorithm progressively more efficient.

4.1. ***Algorithm details and correctness proof.*** We update the preproduct list as follows. For each existing admissible preproduct k', create a new preproduct $k = k'p$ and check that it is also admissible. Recall that $k = \prod p_i$ is admissible if $\gcd(L, W) \leq 2$, where $L = \mathrm{lcm}_i(\ell_b(p_i))$ and $W = \mathrm{lcm}_i(\omega(p_i))$.

The GCD step involves computing and then factoring $\gcd(b^{k-1} - 1, U_{k-\epsilon(k)})$. For each prime p dividing the gcd with $p > P(k)$, we build $n = kp$ and apply the Fermat test and the Lucas test to determine if it is a challenge pseudoprime. Importantly, both b^{k-1} and $U_{k-\epsilon(k)}$ can be computed using a standard "double-and-add" strategy at a cost of $O(\log k)$ arithmetic operations. With such large inputs, it is vital to use a gcd algorithm asymptotically faster than the Euclidean algorithm. The solution is a discrete fast Fourier transform method that requires $\widetilde{O}(n)$ operations on n-bit inputs [SZ04].

For the sieve step, we check primes p in the range $p_{t-1} < p < B/k$ that fall into the arithmetic progression given by Proposition 7. For each such prime, we again construct $n = kp$ and apply the tests directly to see if it is a challenge pseudoprime.

Theorem 8. *Algorithm 1 correctly tabulates all squarefree (b, P, Q)-challenge pseudoprimes up to B.*

Proof. Suppose that $n \le B$ is a (b, P, Q)-challenge pseudoprime. Then we can write $n = p_1 \cdots p_t = kp_t$. By Proposition 5, $\gcd(L, W) \le 2$, and this is true whether L, W are computed for each of the p_i separately, for k, or for n as a whole. Thus, limiting our preproduct list to admissible k is valid. Note that any prime $p \mid k$ satisfies $p \le B^{1/2}$, so finding all primes up to $B^{1/2}$ is sufficient, if space intensive.

Given k, it follows from Propositions 6 and 7 that p_t is a divisor of $\gcd(b^{k-1} - 1, U_{k-\epsilon(k)})$ and that

$$p_t \equiv \begin{cases} k^{-1} \pmod{L}, \\ -k^{-1} \pmod{W}. \end{cases}$$

Note that k^{-1} exists modulo L and modulo W because $\gcd(n, L) = \gcd(n, W) = 1$. Thus, the algorithm will find p_t either through the GCD step or the Sieve step.

Finally, there is no chance of false positives because each potential pseudoprime is subjected to the necessary Fermat and Lucas tests. □

5. Reciprocal sums involving order

The next two sections develop a proof of the asymptotic running time of Algorithm 1. This proof depends on finding upper bounds on the sum over primes

$$\sum_p \frac{1}{p \cdot \operatorname{lcm}(\ell_b(p), \omega(p))}.$$

Since such results are of independent interest, we spend some time here developing the appropriate theory. A general observation is that in order to bound a reciprocal sum of a function $f(n)$, it is not sufficient to know that $f(n)$ is usually large. Instead, we need a precise bound on how often $f(n) \le y$ for a range of values y.

The first step is to prove a slight generalization of a known lemma. Our proof will follow closely the version found as Lemma 3 in [Mur88]. Let b be the base of the Fermat test, and let $\beta = \alpha/\bar{\alpha}$, where $\alpha, \bar{\alpha}$ are the roots of $x^2 - Px + Q$. In this context let Δ be the squarefree part of the discriminant of $x^2 - Px + Q$. Define Γ as the subgroup generated by β of the unit group of the ring of integers of $\mathbb{Q}(\sqrt{\Delta})$, and let Γ_p be the reduction of Γ modulo p.

Note that $\beta^n = 1$ if and only if $\alpha^n = \bar{\alpha}^n$ if and only if $U_n = 0$. Thus $\omega(p) = |\Gamma_p|$.

Lemma 9. *Let $\Gamma = \langle \beta \rangle$ be a nontorsion subgroup of $\mathbb{Q}(\sqrt{\Delta})$. Then there are $O(y^2)$ primes p such that $|\Gamma_p| \le y$, where the constant depends on β.*

Proof. Let n be a positive integer less than y, and consider $\beta^n - 1$. Since $\beta \in \mathbb{Q}(\sqrt{\Delta})$, so is $\beta^n - 1$. Analyzing the numerator, it is straightforward to show that the numerator of $\beta^n - 1$ is at most c^n, where c is a constant depending on P and Q.

Now, define $S = \{\beta^n : 0 \le n \le y\}$. If $|\Gamma_p| \le y$ then two elements of S are equal modulo p, i.e., $\beta^{n_1} = \beta^{n_2} \pmod p$. Without loss of generality, assume $n_1 \ge n_2$ so that $m = n_1 - n_2$ is nonnegative. Then $\beta^{n_1 - n_2} = \beta^m = 1 \pmod p$ with $0 \le m \le y$. Then thinking of $\beta^m - 1$ as an element of $\mathbb{Q}(\sqrt{\Delta})$, we have $\beta^m - 1 = \gamma_1 + \gamma_2\sqrt{\Delta}$, and $\beta^{n_1-n_2} = 1 \pmod p$ implies p divides the numerators of the rational numbers γ_1 and γ_2.

Since Γ is nontorsion, $\beta^m - 1 \ne 0$. Thus $\beta^m - 1 = 0 \pmod p$ for only finitely many p, and this is limited by the numerator being at most c^m. For any given $m = n_1 - n_2 \le y$, there are $O(m) = O(y)$ primes dividing the numerators of both γ_1 and γ_2, where the constant depends on the choice of β. Thus, the total number of primes with $|\Gamma_p| \le y$ is $O(y^2)$. □

The next lemma will be essential in the analysis of the sieve step of Algorithm 1. The authors are very grateful to an anonymous referee for suggesting the usage of the Cauchy–Schwarz inequality, thus improving the bound from $\tilde{O}(X^{-2/3})$ to $\tilde{O}(X^{-1})$.

Lemma 10. *We have*

$$\sum_{\substack{X<p<B \\ \gcd(\ell_b(p),\omega(p))\le 2}} \frac{1}{p \cdot \operatorname{lcm}(\ell_b(p), \omega(p))} = \tilde{O}(X^{-1}),$$

where the sum is over primes and the implicit logarithm factor depends on B, b, P, Q.

Proof. We first utilize the fact that $\gcd(\ell_b(p), \omega(p)) \le 2$ for all primes in the sum, which along with the Cauchy–Schwarz inequality produces the upper bound

$$\sum_{X<p<B} \frac{2}{p \cdot \ell_b(p)\omega(p)} \le \Bigg(\sum_{X<p<B} \frac{1}{p \cdot \ell_b(p)^2}\Bigg)^{\frac{1}{2}} \Bigg(\sum_{X<p<B} \frac{1}{p \cdot \omega(p)^2}\Bigg)^{\frac{1}{2}}.$$

To bound these new sums, we break into two pieces depending on whether $\ell_b(p)$ is greater or less than y (similarly, whether $\omega(p)$ is greater than or less than y).

In the case where $\ell_b(p)$ is small we will use partial summation, and thus require a bound on the count of primes p with $\ell_b(p) \le y$. By [MS87, Lemma 1], we know there are $O(y^2)$ primes with $\ell_b(p) \le y$. Using partial summation, we then have

$$\sum_{\substack{X<p<B \\ \ell_b(p)\le y}} \frac{1}{\ell_b(p)^2} = \frac{1}{y^2} \cdot O(y^2) - \int_1^y O(t^2) \cdot -2t^{-3} \, \mathrm{d}t = O(1) + O(\log y)$$

and so

$$\sum_{\substack{X<p<B \\ \ell_b(p)\le y}} \frac{1}{p \cdot \ell_b(p)^2} \le \frac{1}{X} \sum_{\substack{X<p<B \\ \ell_b(p)\le y}} \frac{1}{\ell_b(p)^2} \le O\bigg(\frac{\log y}{X}\bigg).$$

In the case where $\ell_b(p)$ is large we bound as follows:

$$\sum_{\substack{X<p<B\\ \ell_b(p)>y}} \frac{1}{p\cdot\ell_b(p)^2} \le \frac{1}{y^2}\sum_{X<p<B}\frac{1}{p} \le O\biggl(\frac{\log B}{y^2}\biggr).$$

Balancing the two cases gives $\sum_{X<p<B} 1/(p\ell_b(p)^2) = \widetilde{O}(X^{-1})$.

We now shift to considering $\omega(p)$. The only quadratic cyclotomic polynomials are x^2+1 and $x^2 \pm x+1$. Since our parameter choices result in $\Delta \neq -1,-3$, the only roots of unity in $\mathbb{Q}(\sqrt{\Delta})$ are ± 1. Since we assume D is not a square, we further know that $\beta \neq \pm 1$. From this we conclude that $\langle\beta\rangle$ is nontorsion, and thus by Lemma 9 there are at most $O(y^2)$ primes with $\omega(p) \le y$. Using the same argument as above, we conclude $\sum_{X<p<B} 1/(p\omega(p)^2) = \widetilde{O}(X^{-1})$. The result then follows. □

6. Algorithm analysis

In this section we provide an asymptotic analysis of Algorithm 1. Recall the restrictions on parameter choices laid out in Section 2. First we find the cost of the GCD step.

Theorem 11. *The asymptotic cost of the GCD step for all $k \le X$ is $\widetilde{O}(X^2) + \widetilde{O}(B^{1/2}X^{3/2})$ bit operations and space for $\widetilde{O}(B^{1/2}X^{1/2})$ words.*

Proof. As noted above, for each preproduct $k \le X$ we need to compute $b^{k-1}-1$ and $U_{k-\epsilon(k)}$ at a cost of $\widetilde{O}(k)$ bit operations, then apply a linear gcd algorithm to compute $g(k) = \gcd(b^{k-1}-1, U_{k-\epsilon(k)})$ at a cost of $\widetilde{O}(k)$ bit operations.

In factoring $g(k)$ we do not need a complete factorization; rather we need to find all primes $p < B/k$ that divide $g(k)$. Using the polynomial evaluation method of Pollard and Strassen (see [vzGG03, Theorem 19.3]) this requires $\widetilde{O}((B/k)^{1/2}\cdot\log g(k)) = \widetilde{O}((Bk)^{1/2})$ bit operations and $O((Bk)^{1/2})$ space.

The total cost in bit operations for all $k \le X$ is then

$$\sum_{k\le X} O(k) + \widetilde{O}(k) + \widetilde{O}((Bk)^{\frac{1}{2}}) = \widetilde{O}(X^2) + \widetilde{O}(B^{\frac{1}{2}}X^{\frac{3}{2}}).$$ □

Next we find the cost of the Sieve step of Algorithm 1, broken down by the number of prime factors in the preproduct.

Theorem 12. *Restrict attention to the tabulation of (b, P, Q)-challenge pseudoprimes that are square-free with $t \ge 3$ prime factors. Then the cost in bit operations of the Sieve step in Algorithm 1 is*

$$\widetilde{O}(X^{-\frac{1}{t-1}}B).$$

Proof. By construction we have $n = kp_t$, where $k > X$ and p_t is the largest prime factor dividing n. Since k is admissible, $\gcd(\ell_b(p), \omega(p)) \le 2$ for all $p \mid k$.

Let k' denote k/p_{t-1}, the product of the smallest $t-2$ primes in the preproduct. It follows that $X < k < B^{1-1/t}$ and so $X/k' < p_{t-1} < B^{1-1/t}/k'$. As t increases, k' might become larger than X.

In this case we use the alternate lower bound $p_{t-1} > X^{1/(t-1)}$. This lower bound is true because we construct k so that its prime factors are increasing, and thus if $p_{t-1} \leq X^{1/(t-1)}$ then $k \leq X$, a contradiction.

By Proposition 7 the size of the arithmetic progression to check for each preproduct k is given by $B/(k \operatorname{lcm}(L, W))$, where L and W are computed from the primes dividing k. Then the total cost in arithmetic operations for all preproducts with $t-1$ prime factors is

$$\sum_{X<k<B^{1-1/t}} \frac{B}{k \operatorname{lcm}(L,W)} \leq \sum_{k' \leq X^{1-1/(t-1)}} \sum_{\frac{X}{k'}<p_{t-1}<\frac{B^{1-1/t}}{k'}} \frac{B}{k' p_{t-1} \operatorname{lcm}(\ell_b(p_{t-1}), \omega(p_{t-1}))} + \sum_{X^{1-1/(t-1)}<k'<B^{1-2/t}} \sum_{X^{1/(t-1)}<p_{t-1}} \frac{B}{k' p_{t-1} \operatorname{lcm}(\ell_b(p_{t-1}), \omega(p_{t-1}))}.$$

For both sums the key tool will be Lemma 10. In the first case we have

$$\sum_{k' \leq X^{1-1/(t-1)}} \sum_{\frac{X}{k'}<p_{t-1}<\frac{B^{1-1/t}}{k'}} \frac{B}{k' p_{t-1} \operatorname{lcm}(\ell_b(p_{t-1}), \omega(p_{t-1}))} \leq \sum_{k'<X^{1-1/(t-1)}} \frac{B}{k'} \cdot \tilde{O}\left(\frac{k'}{X}\right) = \tilde{O}\left(\frac{B}{X^{\frac{1}{t-1}}}\right),$$

while in the second case we have

$$\sum_{X^{1-1/(t-1)}<k'<B^{1-2/t}} \sum_{X^{1/(t-1)}<p_{t-1}} \frac{B}{k' p_{t-1} \operatorname{lcm}(\ell_b(p_{t-1}), \omega(p_{t-1}))} \leq \sum_{X^{1-1/(t-1)}<k'<B^{1-2/t}} \frac{B}{k'} \cdot \tilde{O}(X^{-\frac{1}{t-1}}) = \tilde{O}\left(\frac{B}{X^{\frac{1}{t-1}}}\right).$$

Since these arithmetic operations are on integers of size at most B, the result follows. □

Note that we are only utilizing the order statements for one prime in the preproduct; utilizing more seems quite difficult.

If the preproduct is prime and the pseudoprimes have two prime factors then the sum is easier to analyze, namely

$$\sum_{\substack{X<q<B \\ \gcd(\ell_b(q),\omega(q))\leq 2}} \frac{B}{q \operatorname{lcm}(\ell_b(q), \omega(q))},$$

which is $\tilde{O}(B/X)$ by Lemma 10.

These two theorems form the main components of the analysis of Algorithm 1.

Theorem 13. *The worst-case asymptotic running time of Algorithm 1, when restricted to constructing pseudoprimes with t prime factors, is $\tilde{O}(B^{1-1/(3t-1)})$ bit operations.*

If we ignore the cost of factoring, the running time becomes $\tilde{O}(B^{1-1/(2t-1)})$ bit operations when constructing (b, P, Q)-challenge pseudoprimes with t prime factors.

Proof. We balance the cost of the GCD step from Theorem 11 and the cost of the Sieve step from Theorem 12. The bottleneck in the GCD step is factoring, and balancing B/X with $B^{1/2}X^{3/2}$ gives $X = B^{1/5}$ and a running time with main term $B^{4/5}$ in the case $t = 2$. In practice, computing gcds was the bottleneck rather than factoring. If we assume this holds in general, the cost of the GCD step is instead $\tilde{O}(X^2)$. In the case $t = 2$, balancing X^2 with B/X gives $X = B^{1/3}$ and a running time with main term $B^{2/3}$.

For larger t, balancing $BX^{-1/(t-1)}$ with $B^{1/2}X^{3/2}$ gives $X = B^{(t-1)/(3t-1)}$ and a running time of $\tilde{O}(B^{1-1/(3t-1)})$ bit operations. Under the heuristic assumption that the cost of the GCD step is instead $O(X^2)$, balancing with $BX^{-1/(t-1)}$ instead gives $X = B^{(t-1)/(2t-1)}$ and a running time of $\tilde{O}(B^{1-1/(2t-1)})$.

Asymptotically smaller is the cost of finding all primes up to $B^{1/2}$. Applying the Fermat test and Lucas test to each composite constructed requires only $O(\log B)$ arithmetic operations per number on integers with $O(\log B)$ bits. □

7. Computational notes and conclusion

We implemented Algorithm 1 and verified there are no $(2, 1, -1)$-challenge pseudoprimes (i.e., PSW-challenge pseudoprimes) with two or three prime factors less than 2^{80}. Since there are no primes up to 2^{40} which are simultaneously Wieferich and Wall–Sun–Sun, this claim includes composites with square factors.

If such a challenge pseudoprime with two prime factors were to be found, one of the primes would be admissible while satisfying $\epsilon(p) = 1$. It is notable that we found seven admissible primes with $\epsilon(p) = 1$ while generating primes less than 2^{40}:

p	$\ell_2(p)$	$\omega(p)$
61681	40	1542
363101449	171436	1059
4278255361	80	6684774
4562284561	120	147934
4582537681	160453	1428
26509131221	748	14176006
422013019339	290442546	2906

When k had two prime factors, we found the $\gcd(b^{k-1} - 1, U_{k-\epsilon(k)})$ needed factoring more often. However, the total time spent factoring gcds was negligible. Michael Jacobson suggested batch factoring [Ber02] as one possibility for removing factoring as the bottleneck in the running time of Algorithm 1.

One of the reasons the (b, P, Q) test is effective is because of conflicting divisibility conditions. The Fermat condition requires divisibility with respect to $n - 1$. The Lucas condition (with $\epsilon(n) = -1$) requires divisibility with respect to $n + 1$. Seemingly, this conflict will happen independent of the bases chosen. However, 2047 can be checked to be a $(2, 23, 131)$-challenge pseudoprime. The authors are curious how challenging such pseudoprimes are in general. Are there bases for which the subset-product method of construction makes the challenge only moderately challenging?

The authors also note the influence on this problem of the number sought at the end of [Wil77]. That number is simultaneously a Carmichael number, a Lucas pseudoprime to all sequences of a fixed discriminant, and has $\epsilon(n) = -1$, so it would certainly be a challenge pseudoprime. Williams shows that such a number has an odd number of prime factors, has more than three prime factors, and is not divisible by 3.

We conclude by offering our own rewards for exhibiting challenge pseudoprimes:

- \$20 for a $(2, 1, -1)$-challenge pseudoprime with an even number of prime factors.
- \$20 for a $(2, 1, -1)$-challenge pseudoprime with exactly three prime factors.
- \$6 for a $(2, 1, -1)$-challenge pseudoprime divisible by 3.

References

[Ber02] Daniel Bernstein, *How to find small factors of integers*, preprint, 2002.

[Ble96] Daniel Bleichenbacher, *Efficiency and security of cryptosystems based on number theory*, Ph.D. thesis, ETH Zürich, 1996.

[BW80] Robert Baillie and Samuel S. Wagstaff, Jr., *Lucas pseudoprimes*, Math. Comp. **35** (1980), no. 152, 1391–1417. MR 583518

[CDP97] Richard Crandall, Karl Dilcher, and Carl Pomerance, *A search for Wieferich and Wilson primes*, Math. Comp. **66** (1997), no. 217, 433–449. MR 1372002

[CG03] Zhuo Chen and John Greene, *Some comments on Baillie-PSW pseudoprimes*, Fibonacci Quart. **41** (2003), no. 4, 334–344. MR 2022413

[Fei13] Jan Feitsma, *Pseudoprimes*, webpage, 2013.

[GA99] Jon Grantham and Red Alford, *List of primes*, 1999.

[Gil13] Jeff Gilchrist, *Pseudoprime enumeration with probabilistic primality tests*, webpage, 2013.

[Gra01] Jon Grantham, *Frobenius pseudoprimes*, Math. Comp. **70** (2001), no. 234, 873–891. MR 1680879

[Jae93] Gerhard Jaeschke, *On strong pseudoprimes to several bases*, Math. Comp. **61** (1993), no. 204, 915–926. MR 1192971

[JD14] Yupeng Jiang and Yingpu Deng, *Strong pseudoprimes to the first eight prime bases*, Math. Comp. **83** (2014), no. 290, 2915–2924. MR 3246815

[JQ96] Marc Joye and Jean-Jacques Quisquater, *Efficient computation of full Lucas sequences*, Electron. Lett. **32** (1996), no. 6, 537–538.

[Leh30] D. H. Lehmer, *An extended theory of Lucas' functions*, Ann. of Math. (2) **31** (1930), no. 3, 419–448. MR 1502953

[MS87] M. Ram Murty and S. Srinivasan, *Some remarks on Artin's conjecture*, Canad. Math. Bull. **30** (1987), no. 1, 80–85. MR 879875

[Mur88] M. Ram Murty, *Artin's conjecture for primitive roots*, Math. Intelligencer **10** (1988), no. 4, 59–67. MR 966133

[Nic12] Thomas R. Nicely, *The Baillie-PSW primality test*, preprint, 2012.

[Pin00] Richard G. E. Pinch, *The pseudoprimes up to* 10^{13}, Algorithmic number theory, Lecture Notes in Comput. Sci., no. 1838, Springer, 2000, pp. 459–473. MR 1850626

[Pom84] Carl Pomerance, *Are there counter-examples to the Baillie-PSW primality test?*, afterword to the doctoral thesis of A. K. Lenstra, 1984.

[PSW80] Carl Pomerance, J. L. Selfridge, and Samuel S. Wagstaff, Jr., *The pseudoprimes to* $25 \cdot 10^9$, Math. Comp. **35** (1980), no. 151, 1003–1026. MR 572872

[SW17] Jonathan Sorenson and Jonathan Webster, *Strong pseudoprimes to twelve prime bases*, Math. Comp. **86** (2017), no. 304, 985–1003. MR 3584557

[SZ04] Damien Stehlé and Paul Zimmermann, *A binary recursive gcd algorithm*, Algorithmic number theory, Lecture Notes in Comput. Sci., no. 3076, Springer, 2004, pp. 411–425. MR 2138011

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard, *Modern computer algebra*, 2nd ed., Cambridge University Press, 2003. MR 2001757

[Wil77] H. C. Williams, *On numbers analogous to the Carmichael numbers*, Canad. Math. Bull. **20** (1977), no. 1, 133–143. MR 0447099

Received 2 Mar 2018. Revised 18 Jun 2018.

ANDREW SHALLUE: ashallue@iwu.edu
Department of Mathematics, Illinois Wesleyan University, Bloomington, IL, United States

JONATHAN WEBSTER: jonathan.webster1@ucalgary.ca
Department of Mathematics, Statistics & Actuarial Science, Butler University, Indianapolis, IN, United States

VOLUME EDITORS

Renate Scheidler
University of Calgary
Calgary, AB T2N 1N4
Canada

Jonathan Sorenson
Butler University
Indianapolis, IN 46208
United States

The cover image is based on a design by Linh Chi Bui.

Electronic copies can be obtained free of charge from http://msp.org/obs/2 and printed copies can be ordered from MSP (contact@msp.org).

The Open Book Series is a trademark of Mathematical Sciences Publishers.

ISSN: 2329-9061 (print), 2329-907X (electronic)

ISBN: 978-1-935107-02-6 (print), 978-1-935107-03-3 (electronic)

First published 2019.

MATHEMATICAL SCIENCES PUBLISHERS
798 Evans Hall #3840, c/o University of California, Berkeley CA 94720-3840
contact@msp.org http://msp.org

THE OPEN BOOK SERIES 2

Thirteenth Algorithmic Number Theory Symposium

The Algorithmic Number Theory Symposium (ANTS), held biennially since 1994, is the premier international forum for research in computational number theory. ANTS is devoted to algorithmic aspects of number theory, including elementary, algebraic, and analytic number theory, the geometry of numbers, arithmetic algebraic geometry, the theory of finite fields, and cryptography.

This volume is the proceedings of the thirteenth ANTS meeting, held July 16-20, 2018, at the University of Wisconsin-Madison. It includes revised and edited versions of 28 refereed papers presented at the conference.

Edited by Renate Scheidler and Jonathan Sorenson

CONTRIBUTORS

Simon Abelard
Sonny Arora
Vishal Arul
Angelica Babei
Jens-Dietrich Bauch
Alex J. Best
Jean-François Biasse
Alin Bostan
Reinier Bröker
Nils Bruin
Xavier Caruso
Stephanie Chan
Qi Cheng
Gilles Christol
Owen Colman
Edgar Costa
Philippe Dumas
Kirsten Eisenträger
Claus Fieker
Shuhong Gao
Pierrick Gaudry
Alexandre Gélin
Alexandru Ghitza
Laurent Grémy
Jeroen Hanselman
David Harvey
Tommy Hofmann
Everett W. Howe
David Hubbard
Kiran S. Kedlaya
Thorsten Kleinjung
David Kohel
Wanlin Li
Richard Magner
Anna Medvedovsky
Michael Musty
Ha Thanh Nguyen Tran
Christophe Ritzenthaler
David Roe
J. Maurice Rojas
Nathan C. Ryan
Renate Scheidler
Sam Schiavone
Andrew Shallue
Jeroen Sijsling
Carlo Sircana
Jonathan Sorenson
Pierre-Jean Spaenlehauer
Andrew V. Sutherland
Nicholas Triantafillou
Joris van der Hoeven
Christine Van Vredendaal
John Voight
Daqing Wan
Lawrence C. Washington
Jonathan Webster
Benjamin Wesolowski
Yinan Zhang
Alexandre Zotine