Algorithmic Number Theory Symposium XVII, Groningen, July 6-10, 2026

# ALGORITHMS FOR CARMICHAEL NUMBERS

ANDREW SHALLUE AND JONATHAN WEBSTER

Abstract. We address the computational complexity of algorithms that find all Carmichael numbers less than some specified bound $B$. These algorithms depend on the query "Is $n$ a Carmichael number?" We show CARMICHAELS is in P, where only the run-time is conditioned on the ERH. This enables a heuristically optimal tabulation algorithm, which is the first asymptotic improvement to tabulation algorithms in the 50 years since Swift first described the prime-by-prime approach [40]. We implemented a related algorithm that tabulated 100 times further while only doing about 5 times the work of the prior tabulation. We found $308,279,939$ Carmichael numbers less than $10^{24}$ and we provide some statistics on these numbers.

## 1. Introduction

*Carmichael numbers* are the composite numbers $n$ satisfying $b^n \equiv b \pmod{n}$ for every integer $b$. Fermat's little theorem says that the congruence must hold whenever $n$ is a prime and so these numbers might also be called *absolute Fermat pseudoprimes*. In [5], Robert Carmichael proved that such numbers must be square-free and satisfy $(p-1) \mid (n-1)$ for every prime $p \mid n$. He also gave four examples including 561, the smallest Carmichael number[1]. Given their status as pseudoprimes and the many open conjectures regarding their behavior, finding examples of Carmichael numbers and tabulating them has been an ongoing project for the last 50 years.

In 1975, J.D. Swift described an algorithm used to find all Carmichael numbers up to $10^9$ [40]. Every tabulation thereafter followed the same basic approach. The improvements offered by Jaeschke [18], Pinch [27], and Shallue & Webster [37, 38] all involved faster ways of handling a small set of inputs. Thus while the tabulation bound was pushed (eventually to $10^{22}$), the dominating asymptotic for all these approaches remained Knödel's bound, generated algorithmically by constructing Carmichael numbers prime by prime.

**Theorem 1** ([20])**.** *Let $C(B)$ be the count of Carmichael numbers less than $B$. Then*

$$C(B) < B \exp\left\{\left(-\frac{1}{\sqrt{2}} + o(1)\right)\sqrt{\ln B \ln_2 B}\right\}.$$

*Let $\mathcal{K}(B)$ denote this bound on the count of Carmichael numbers.*

In the above, $\ln_2 n$ denotes the twice-iterated natural logarithm, and $\ln_k n$ will denote the $k$-fold iterated natural logarithm.

Shortly after Knödel proved his result, Erdős found a better upper bound [9]. We state an improved version proven by Pomerance, Selfridge, and Wagstaff.

[1] In 1899, Korselt had previously proven this result but gave no examples. In 1885, Šimerka found examples but did not state how they were found.

**Theorem 2** ([34, Theorem 6])**.** *For each $\epsilon > 0$, there is a $B_0(\epsilon)$ such that for all $B > B_0(\epsilon)$, we have*

$$C(B) \leq B \exp\left\{\frac{-(1-\epsilon)\ln B \ln_3 B}{\ln_2 B}\right\}.$$

*Let $\mathcal{E}(B)$ denote this bound on the count of Carmichael numbers.*

In [33, 34], a heuristic argument is given that $C(B) \sim \mathcal{E}(B)$. However, the best proven lower bounds are of the form $B^\alpha$ for some fixed $\alpha$. Following the seminal work of [2], where the infinitude of Carmichael numbers was first proven with $\alpha = 2/7$, we have a series of refinements improving the exponent to 0.332 [13], then 0.33336704 [14], and finally 0.3389 [23]. Those strictly guided by empirical observations may find the lower bounds more convincing than the heuristic argument because at our current tabulation range the largest exponent observed is 0.35.

The major result of the present work is an algorithm with $\mathcal{E}(B)$ as its asymptotic running time. This supersedes all previous work on Carmichael number tabulation, giving the first asymptotic improvement since Swift first described his approach.

**Theorem 3.** *For each $\epsilon > 0$, there is an $B_0(\epsilon)$ such that for all $B > B_0(\epsilon)$, there is an algorithm that can find all Carmichael numbers less than $B$ in at most $\mathcal{E}(B)$ queries of "Is $n$ a Carmichael number?".*

Thus the computational complexity of answering the query "Is $n$ a Carmichael number?", which we denote as CARMICHAELS, is of central importance to the present work. This query is also of independent interest; it is a bit of folklore that Carmichael numbers are easy to factor based on the belief that most Carmichael numbers are smooth. Since some Carmichael numbers are difficult to factor, factoring seems unavoidable, and general factorization algorithms are not in **P**, the following theorem is noteworthy.

**Theorem 4.** *CARMICHAELS is in **P**, where correctness is unconditional and the run-time analysis is conditioned on the Extended Riemann Hypothesis (ERH).*

Theorems 3 and 4 together justify the asymptotic running time of $\mathcal{E}(B)$. If the heuristic argument of [33, 34] is correct, then we can further conclude that our algorithm is optimal.

A pleasing feature of the present work is that the Fermat definition of Carmichael numbers plays a central role, where other tabulations ignore it in favor of Korselt's criterion. Swift wrote, "The computer programs used depended explicitly on congruence properties of the CN's with respect to their component primes rather than on the pseudoprimality with respect to any particular base."

In [38, Theorem 8] we proved that $\mathcal{K}(B)$ is algorithmically realized. The primary algorithmic advance in the present work is our use of Theorem 4 to search for composite $R$ that completes a preproduct $P$ to form Carmichael numbers of the form $n = PR$. This provides an algorithmic realization of $\mathcal{E}(B)$. Relaxing the requirement for completions to be prime is the source of the separation between $\mathcal{K}(B)$ and $\mathcal{E}(B)$.

In the next section we establish notations and state some elementary theorems used in tabulation algorithms. In Section 3 we state the central algorithm to prove the results on CARMICHAELS. Since we claim that our new tabulation algorithms supersedes prior work, we use Section 4 to provide a historical survey of the prime-by-prime algorithms. The section concludes with a heuristic argument that they

all had $\mathcal{K}(B)$ as the run-time. Section 5 gives two asymptotically faster algorithms. The first of which is a practical algorithm that was implemented and the second is a heuristically optimal algorithm. Section 6 presents results of the computation and the tabulation of Carmichael numbers up to $10^{24}$. We conclude by presenting some related problems.

## 2. Number theoretic results on Carmichael numbers

We frequently recognize Carmichael numbers $n$ via their prime factorization.

**Theorem 5** (Korselt's Criterion)**.** *A composite number $n$ is a Carmichael number if and only if $n$ is squarefree and $(p-1) \mid (n-1)$ for all prime divisors $p$ of $n$.*

Robert Carmichael proved Korselt's criterion in the context of the reduced totient function or the Carmichael function $\lambda(n)$,

$$\lambda\left(p^{\alpha}\right) = \begin{cases} \phi\left(p^{\alpha}\right) & \text{if } \alpha \leq 2 \text{ or } p \geq 3 \\ \frac{1}{2}\phi\left(p^{\alpha}\right) & \text{if } \alpha \geq 3 \text{ and } p = 2 \end{cases},$$

$$\lambda(n) = \lambda\left({p_1}^{\alpha_1} \cdots {p_k}^{\alpha_k}\right) = \operatorname{lcm}\left(\lambda\left({p_1}^{\alpha_1}\right), \ldots, \lambda\left({p_k}^{\alpha_k}\right)\right),$$

where the $p_i$ are distinct prime divisors of $n$ and $\phi(n)$ is Euler's totient function. Another function used in this work is $\omega(n)$, which returns the number of distinct prime factors of $n$. We can now state Korselt's criterion in terms of $\lambda(n)$.

**Theorem 6** (Korselt's Criterion)**.** *A composite number $n$ is a Carmichael number if and only if $\lambda(n) \mid (n-1)$.*

We reserve the following notation.

- Let $P_k = \prod_{i=1}^{k} p_i$ where $p_i < p_j$ iff $i < j$ and call $P_k$ a *preproduct*.
- We drop the subscript $k$ from $P_k$ when the context does not require a specific count of primes.
- We let $p$ be the greatest prime factor of $P$.
- We will use $q$ or $r$ to denote primes.

**Theorem 7** ([27, Proposition 1])**.** *Let $n = P_d < B$ be a Carmichael number.*

1. *Let $k < d$. Then $p_{k+1} < (B/P_k)^{1/(d-k)}$ and $p_{k+1} - 1$ is relatively prime to $p_i$ for all $i \leq k$.*
2. *$P_{d-1}p_d \equiv 1 \pmod{\lambda(P_{d-1})}$ and $p_d - 1$ divides $P_{d-1} - 1$.*
3. *Each $p_i$ satisfies $p_i < \sqrt{n} < \sqrt{B}$.*

Part 2 of Theorem 7 can be generalized: we have $PR \equiv 1 \pmod{\lambda(P)}$ for arbitrary positive $R$ whenever $n$ is Carmichael and $n = PR$. The quantity $P^{-1} \pmod{\lambda(P)}$ is used frequently throughout and we let $r^{\star}$ denote the least positive integer in that residue class. This might be considered an unconventional naming choice but it is due to the fact that we associate this quantity to $R$, the remaining factor to be constructed. So, the completion $R$ is in the arithmetic progression

$$\mathcal{R} = \{r^{\star} + k\lambda(P), \text{ for } 0 \leq k \leq \lfloor B/(P\lambda(P)) \rfloor\}.$$

Products $P$ where $p_j - 1$ is relatively prime to all $p_i$ for $i < j$ are called *cyclic*. If there exists a prime $q$ and a cyclic $P$ such that $Pq$ is also cyclic we say that $q$ is *admissible* to $P$.

Part 2 of Theorem 7 also implies that the number of Carmichael numbers with a fixed second largest prime factor is finite. A stronger result implies that the number of Carmichael numbers with a fixed third largest prime factor is also finite.

**Theorem 8.** *Let $n = Pqr$ with $p < q < r$ be a Carmichael number. There are integers $2 \le D < P < C$ such that, putting $\Delta = CD - P^2$, we have*

$$q = \frac{(P-1)(P+D)}{\Delta} + 1, \quad r = \frac{(P-1)(P+C)}{\Delta} + 1, \quad P^2 < CD < P^2\left(\frac{p+3}{p+1}\right).$$

A version of Theorem 8 was proved by Beeger for $P$ prime [4], and Duparc generalized it to composite $P$ [8]. Jaeschke seems to be the first to incorporate an algorithmic version [18]. The version above matches Pinch's use in [27].

## 3. CARMICHAELS IS IN P, CONDITIONALLY

We address the computational complexity of CARMICHAELS, the decision problem "Is $n$ a Carmichael number?" The chief difficulty lies in answering "yes" to this query. We present the prior work on this problem, and show how answering "yes" is related to integer factorization. We give a brief review of Miller's primality test; it is motivational and necessary for our own approach. This leads to a simplistic Miller-inspired way of resolving the query. We present a modification to this method that gives the complete prime factorization of Carmichael numbers, which means the algorithm will be unconditionally correct. This will demonstrate that CARMICHAELS is in P where only the run-time is conditioned on the ERH. We conclude with a note on the bit complexity of CARMICHAELS.

### 3.1. Prior work.

In 1978, the authors of [34] said that it was "usually much harder to show that a given large number is Carmichael than it is to show that it is a [base $b$ Fermat pseudoprime]." They noted the obvious approach of factoring $n$ to invoke Korselt's criterion and that $n$ may be hard to factor. Then they proposed an alternative way where divisors of $n-1$ are used to construct potential $p-1$ and thereby find factors of $n$. Thus at first glance, the computational complexity of CARMICHAELS is that of integer factorization (of either $n$ or $n-1$), which we account as a difficult problem.

### 3.2. The strong Fermat test.

The weak Fermat test (checking if $b^n \equiv b \pmod n$) is easy to implement and has (after factoring $b$ out) a difference of squares factorization. Miller used the following theorem to design a reliable primality test [25] that exploits these easy-to-recognize algebraic factors.

**Theorem 9** ([7, Theorem 3.5.1])**.** *Suppose $n$ is an odd prime and $n - 1 = 2^s n'$ where $n'$ is odd. If $b$ shares no common factors with $n$ then either $b^{n'} \equiv 1 \pmod n$, or there exists $i$ such that $0 \le i < s$ and $b^{2^i n'} \equiv -1 \pmod n$.*

We call using Theorem 9 to detect if a number is composite a *strong Fermat test*. Under the ERH, a witness for compositeness will be found for some $b < 2\ln^2 n$ [7, Theorem 3.5.12]. This almost establishes PRIMES is in P; correctness and runtime are contingent on the ERH.

3.3. **CARMICHAELS is in P, conditionally.** We follow Miller's approach to establish that CARMICHAELS is in P: use the weak Fermat test for every $b \in [2, 2\ln^2 n)$. Failure to witness that $n$ is composite implies it is a Carmichael number[2]. The correctness and run-time are conditioned on the ERH (for further details, see the proof of Theorem 4).

It is unfortunate that this method does not provide the prime factors of $n$, since the complete prime factorization forms a certificate of the fact that $n$ is Carmichael. Another practical weakness of this method is that correctness is conditional; a weakness which is especially unfortunate if used in a tabulation algorithm. These two reasons provide the motivation to study algorithms for factoring Carmichael numbers.

3.4. **Factoring Carmichael numbers.** It has long been known that it is simple to split $n$ if the weak Fermat test fails to detect $n$ is composite and the strong Fermat test detects $n$ is composite [3, 26]. Therefore, Carmichael numbers are easily split. To provide a complete prime factorization, splitting a number is often sufficient. We recursively invoke the algorithm on the smaller factors until we have the complete prime factorization. Unfortunately Fermat splitting cannot proceed in this way; it is unlikely that a composite divisor will be a Fermat pseudoprime. As noted in [41], "This algorithm might not factor $n$ completely into prime factors."

For the rest of the subsection, we assume that $n$ is a Carmichael number and we wish to provide a proof of that fact by exhibiting its prime factorization. We give a high-level description of how the splittings witness all possible factorizations, state the algorithm, and prove it is unconditionally correct.

We can better understand how the splitting occurs via the (repeated) difference of squares factorization of $b^{n-1}-1$. Let $n'$ be the largest odd divisor of $n-1$ and $s = \nu_2(n-1)$. Then

$$b^{n-1} - 1 = (b^{n'} - 1) \prod_{i=0}^{s-1} (b^{2^i n'} + 1).$$

If $n$ divides the left side of the equation (the weak Fermat test does not detect $n$ is composite), then its prime factors are found in the algebraic factors on the right side. When $n$ is a strong pseudoprime, all prime factors of $n$ align in a single algebraic factor. Otherwise, the prime factors are distributed in a predictable way across two or more factors. Here is a visual representation of the $s = 3$ case. Since $s = 3$ and $n$ is a Carmichael number, we also know $\nu_2(q-1) \le 3$ for all $q \mid n$.

$$b^{8n'} - 1 = \underbrace{(b^{4n'} + 1)\overbrace{(b^{2n'} + 1)\underbrace{(b^{n'} + 1)(b^{n'} - 1)}_{\nu_2(q-1)=1}}^{\nu_2(q-1)=2}}_{\nu_2(q-1)=3}$$

Since Fermat's little theorem says $b^{q-1} \equiv 1 \pmod q$, the braces show where the divisibility occurs by the 2-valuation of $q-1$. Further, the divisibility will be in the leftmost factor (as delimited by the braces) whenever $b$ is not a square mod $q$.

[2]Or $n$ could be an absolute pseudoCarmichael; that is, it could be prime. We assume that these algorithms are invoked on composite integers, and that composite inputs are guaranteed by some other means, e.g. [1] .

We compute $\gcd(n, b^{2^i n'} + 1)$ for each $0 \leq i < s$ to split these prime factors from others. These gcd computations correspond to the exponentiations required for the strong Fermat test, and allow multiple non-trivial factors of $n$ to be found with a single $b$. Previous literature on this problem (e.g., [3, 26]) focused only on the smallest valuation which was guaranteed to witness a splitting.

We have to keep track of the information obtained from the splittings. A high-level description of how to answer the query "Is $n$ a Carmichael number?" is that we iterate over different choices of $b$. For each $b$ we will likely either

(1) witness $n$ is composite with the weak test and answer "no", or
(2) progress towards a complete prime factorization with the strong test.

We appeal to Korselt's criterion when we find a complete prime factorization of $n$. More robustly, we have:

**Algorithm 1** Returns the truth value of "$n$ is a Carmichael number."

**Require:** $n$, a positive composite integer with $s = v_2(n-1)$.
1: Initialize $b = 1$, and empty queues PRIMES, NEXT_COMP, and CUR_COMP.
2: Put $n$ into CUR_COMP.
3: **while** CUR_COMP is not empty **do**
4: Assign $b = \text{NextPrime}(b)$
5: Compute $X[i] \equiv b^{2^i n'} \pmod{n}$ for $0 \leq i < s$
6: **if** $b^n \not\equiv b \pmod{n}$ **return** False
7: **for each** $m$ in CUR_COMP, **pop** $m$ and **do**
8: **for** $i < s$ **do**
9: Calculate $g = \gcd(m, X[i] + 1)$
10: $m = m/g$
11: **if** $1 < g$ **then** put $g$ into NEXT_COMP or PRIMES (as appropriate)
12: **end for**
13: **if** $1 < m$ **then** put $m$ into NEXT_COMP or PRIMES (as appropriate)
14: **end for**
15: Swap NEXT_COMP and CUR_COMP
16: **end while**
17: **return** $\lambda(n) \mid (n-1)$

**Remarks:**

- Lines 5-6 are all part of one square-and-multiply exponentiation ladder that represent the strong and weak Fermat tests.
- Lines 11 and 13 are, perhaps, understated. Putting these values in the queues requires a primality check. Any time a composite check is done, we could also check that the number is not a nontrivial power, returning "False" if detected. Letting $P$ be the product of the known prime factors, we could also return "False" if $P$ is not cyclic or if $\lambda(P) \nmid (n-1)$. If this is done as part of lines 11 and 13, then line 17 could read "**return** True".
- If prime or composite factors of $n$ are already known, initialize PRIMES and CUR_COMP with the respective factors.

**Example 1.** *Consider* $n = 3494075153422874350506032047195872 01$*, the least Carmichael number with* 20 *prime factors. A number enclosed with parentheses denotes a composite and these numbers are in* CUR_COMP *for the next choice of* $b$*.*

- $b = 2$: $n = 17 \cdot 97 \cdot 193 \cdot 641 \cdot (24261623)(8987)(7855279748239573)$.
- $b = 3$: $24261623 = 71 \cdot 73 \cdot (4681)$, $8987 = 11 \cdot (817)$, $7855279748239573 = 41 \cdot 113 \cdot (256477)(2257)(2929)$.
- $b = 5$: $4681 = (4681)$, $817 = 19 \cdot 43$, $256477 = 13 \cdot (19729)$, $2257 = 37 \cdot 61$, $2929 = 29 \cdot 101$.
- $b = 7$: $4681 = 31 \cdot 151$, $19729 = 109 \cdot 181$.

*We have the complete prime factorization of $n$ and can verify that it is a Carmichael number with Korselt's criterion.*

**Theorem 4.** CARMICHAELS *is in* **P**, *where correctness is unconditional and the run-time analysis is conditioned on the Extended Riemann Hypothesis (ERH).*

*Proof.* Algorithm 1 is unconditionally correct; lines 6 and 17 exit with no error. We need to show that it finishes in the required time constraint. Both cases use Theorem 1.4.5 of [7].

Under the ERH and for composite $n$ that are not Carmichael numbers, a witness for compositeness is found (see line 6) for some $b < 2\ln^2 n$. Since $n$ is not Carmichael, there is a prime $q$ with $q \mid n$ and $(q-1) \nmid (n-1)$. The ERH allows us to find an element $b < 2\ln^2 n$ whose order does not divide $n-1$.

We now establish that under the ERH and for $n$ that are Carmichael numbers, Algorithm 1 finds the complete prime factorization from a set of $b < 2\ln^2 n$. It is sufficient to show that any two distinct primes may be split from each other (see line 9); consider $m = q_1 q_2$ a product of two distinct primes, a composite divisor of $n$ with $v_2(q_i - 1) = s_i \leq v_2(n-1)$ for $i = 1, 2$.

Case 1: When $s_1 = s_2$, the ERH guarantees the existence of a $b$ that is a square mod $q_1$ but not a square mod $q_2$ (or vice-versa) in the interval $[2, 2\ln^2 q_1 q_2)$.

Case 2: If $s_1 < s_2$, it is sufficient to find $b$ that is not a square mod $q_2$. The ERH says this will happen for some $b$ in the interval $[2, 2\ln^2 q_2)$. □

What is the contrast between using Miller's test to address PRIMES and using Algorithm 1 to address CARMICHAELS? Correctness. For Miller's test, a "yes" answer is contingent on the ERH. For Algorithm 1, if the ERH is wrong it only means that we wait longer than we had expected.

One can re-do all of Sections 3.2-3.4, and follow a Rabin-inspired approach to primality testing where $b$ are selected uniformly at random [35]. Everything follows through similarly except counting arguments (see [3, 26]) take the place of appealing to the ERH. The end result is the following theorem.

**Theorem 10.** CARMICHAELS *is in* **ZPP** *(zero-error probabilistic polynomial time).*

The interested reader may find the details in [42, Theorem 3.9].

3.5. **Bit complexity of CARMICHAELS.** Having established the computational complexity of CARMICHAELS, we briefly address the bit complexity. The average case (and the empirically dominating case) corresponds to detecting $n$ is composite with the first base-2 Fermat test; any practical improvements to Algorithm 1 should be focused on this case. So, we prove a bound on the worst case.

**Theorem 11.** *The bit complexity of Algorithm 1 is* $O(\ln^{10} n \ln_2^{c_0} n)$.

*Proof.* There is a factor of $\ln^2 n$ coming from the number of Fermat bases. We can bound the loop in line 7 by a factor of $\ln n$ because that bounds the number

of factors that can multiply to $n$. We bound the loop in line 8 by a factor of $\ln n$ because $s$ is bounded by the bit-length of $n$. The largest cost per iteration of the while loop occurs in lines 11 and 13: prime testing required to place $g$ and $m$ in their respective queues. The Lenstra–Pomerance variant [22] of AKS [1] accomplishes this in time $O(\ln^6 n \ln_2^{c_o} n)$ for some effectively computable constant $c_o$. This gives a bit-complexity of $O(\ln^{10} n \ln_2^{c_0} n)$. □

We doubt this result is sharp for CARMICHAELS as a general problem. First and foremost, we suspect PRIMES is sharper than $O(\ln^6 n \ln_2^{c_o} n)$, and any improvement on PRIMES immediately carries over to this result. Second, there are possible theoretical and structural changes to Algorithm 1 that could be considered to make the bound sharper. We leave this open for others to investigate.

## 4. Algorithms akin to Knödel's Argument - A survey

Section 4.1 provides a brief sketch of the algorithmic background for tabulating Carmichael numbers. Section 4.2 surveys prior work that advanced the algorithmic theory of Carmichael tabulation. Section 4.3 summarizes [38] to provide a unifying understanding of all prior tabulation algorithms. Section 4.4 concludes with some of the observed problems with the prior tabulations and gives a motivating example to demonstrate the power of Algorithm 1.

### 4.1. **Algorithmic Basics.**

Since tabulations algorithms require a degree of proficiency with several elementary algorithms from number theory, we highlight some of the subproblems that arise. We sketch solutions to two of them (for more details see [42, Chapter 4]). Then we provide a brief sketch of the two ways in which a preproduct $P$ was completed to find either the set of primes $r$ so that $n = Pr$ is a Carmichael number or pairs of primes $q, r$ so that $n = Pqr$ is a Carmichael number.

#### 4.1.1. *Basic Number Theoretic Algorithms.*

We assume a familiarity with the ability to use a variant of the sieve of Eratosthenes to quickly factor any number in an interval (cf. [7, Chapter 3.2]). Our implementation used a bit-packed incremental sieve [15]. Implicit in the sieve is the ability to sieve elements in an arithmetic progression, which we use in Section 5.1.

There are many low-level choices that can made regarding elementary computations that do not effect the asymptotic arguments but have significant impact on the wall-time of a computation. Here are two such examples. We check if $P$ is cyclic by checking congruence conditions with respect to the primes dividing $P$ rather than compute $\gcd(P, \phi(P))$. We compute $\lambda(Pq) = \operatorname{lcm}(\lambda(P), q-1)$ rather than treat this as a *de novo* computation from the prime factorization of $Pq$.

#### 4.1.2. *Algorithms for Completing Preproducts.*

First, we show how Theorem 7 is used to find all primes $r$ so that $Pr$ is a Carmichael number. Then we show how Theorem 8 is used to find all $R$ with $\omega(R) = 2$ so that $PR$ is a Carmichael number.

To complete $P$ with a single prime, we use Part 2 of Theorem 7 and recognize it as a "divisor in residue class" problem. There are efficient algorithms that solve this problem [21, 6] and we incorporated them into the tabulation of [38]. We refer to an algorithm that can find the set of primes so that $Pr$ is a Carmichael number as an **SPC** (single prime completion) algorithm. A well-written **SPC** algorithm should be no slower than checking the arithmetic progression $\mathcal{R}$ for primes.

To complete $P$ with exactly two primes, we use Theorem 8. Jaeschke and Pinch [18, 27] iterate over the quantities $D$ and $C$ to find $R = qr$ and this was effective for $P$ prime. In [37], we showed that you can iterate over the quantities $D$ and $\Delta$ to handle composite $P$ nearly as efficiently. We also provided asymptotic analysis of these algorithms and argued that if $P < B^{1/3}$, it was better to invoke this algorithm than perform an exhaustive search for the two primes.

4.2. **The Survey.** For Carmichael numbers with $d$ prime factors, prime-by-prime completion methods will create $P_{d-1} = P_{d-2}p_{d-1}$. This cyclic number satisfies $P_{d-2}p_{d-1}^2 < B$ so that there is enough room for one more prime. The count of these numbers matches $\mathcal{K}(B)$ [38, Theorem 8] and serves as the asymptotic analysis of all of these algorithms since producing these inputs is the most expensive step. Our survey below focuses on algorithmic advancements. Several authors have extended the tabulation bound using known algorithms, including $B = 25 \cdot 10^9$ in [34], $B = 10^{13}$ in [19], $B = 10^{14}$ in [12], and $B = 10^{22}$ in [10].

4.2.1. *Swift's Tabulation.* The first systemic tabulation of Carmichael numbers was done by J.D. Swift in 1975 on a CDC 1604 and his bound was $10^9$ [40]. He conducted four different tabulation regimes. One each for preproducts with $d = 2, \ldots, 5$ prime factors. This sets the stage for all tabulations that follow.

4.2.2. *Jaeschke's Tabulation.* In 1990, Gerhard Jaeschke found all Carmichael numbers less than $10^{12}$ [18]. Jaeschke had special routines for $d = 3$ and $d = 4$ for when $P$ was small that were like the $CD$ method of Section 4.1.2. The set of inputs for his algorithm were preproducts $P$ such that $Pp < 10^{12}$.

4.2.3. *Pinch's Tabulations.* In 1993, Richard Pinch tabulated up to $10^{15}$ [27]. Over the next fourteen years he used the same algorithm to extend the bound by orders of magnitude up to $10^{21}$ [28, 29, 30, 31].

One novel aspect of his tabulation was the "large prime variation" which created numbers divisible by a prime greater than $B^{1/3}$. This stage provided some inspiration for our own approach. He used the Fermat test as a rejection mechanism. For prime $P > B^{1/3}$, he created $n = PR$ with $R \in \mathcal{R}$ and for each one he tested if $2^n \equiv 2 \pmod{n}$. For those that passed, he used trial division to factor $R$ in order to apply Korselt's criterion. He argued that using the Fermat test as a filter would cause the cost of factoring to be a lower-order cost in the context of a complete tabulation. We repeat his argument and offer a different perspective on his conclusion that is relevant for our present work.

The cost to perform the trial division is $O(\sqrt{B/P}) = O(B^{1/3})$ for each $R$, which ranges in size from $B^{1/3}$ to $B^{2/3}$, coming from a given $P$. Let $C_B$ be the count that pass to the next stage. The count of all $PR$ is

$$\sum_{B^{1/3}<P<B^{1/2}} \frac{B}{P(P-1)} = O(B^{2/3}).$$

The total cost is $O(B^{2/3} + C_B B^{1/3})$, where the first term is the cost to create and filter all inputs and the second term is the cost of factoring the $R$ for the base-2 Fermat pseudoprimes that pass the filter. He concludes that as $C_B B^{1/3}$ is "noticeably smaller than $B^{2/3}$, the large-prime variation gives an improvement."

What are we counting with $C_B$? The base-2 Fermat pseudoprimes. The count of all base-2 Fermat pseudoprimes less than $B$ is conjectured to be of the form

$B \exp\{-\ln B \ln_3 B/\ln_2 B\}$. We are concerned with a subset of size $O(B^{2/3})$ and we extrapolate the exponential term as a probability-like filter. We conjecture that $C_B \sim B^{2/3} \exp\{-\ln B \ln_3 B/\ln_2 B\}$. Viewed this way, Pinch's claim that $C_B B^{1/3}$ is noticeably smaller than $B^{2/3}$ is not a product of asymptotic analysis but of the empirical rarity of pseudoprimes in the tabulation range. Theoretically, $B^{2/3}$ is dominated by the nearly linear $C_B B^{1/3}$ term!

An alternative expression for the cost is $O(B^{2/3} + C_B C_F)$, where $C_B$ is as before and $C_F$ is the cost to factor. It should be clear than any of the general deterministic factoring algorithms will cause the $C_B C_F$ term to dominate. However, if we use Algorithm 1 then this term is, indeed, the lower-order term.

4.3. **Tabulation for $B = 10^{22}$.** We encourage the reader to consult [27, 37, 38] for a thorough explanation, and we give a concise description suitable for analysis and understanding. Our approach was bifurcated first by the size of $P$. For large preproducts $P$ the tabulation was accomplished by a series of individual tabulations organized by the count of primes dividing $P$.

Using the algorithms presented in [37], the small preproduct regime considers all $P < X$ and constructs $n = Pqr$. The cost of this phase is $O((X \ln X)^2)$. So long as $X$ is chosen so that $(X \ln X)^2$ is dominated by $\mathcal{K}(B)$, this regime is a lower-order cost in the asymptotic analysis. In practice, we chose $X$ to slightly exceed $B^{1/3}$ and, by doing so, this regime finds all Carmichael numbers less than $B$ with exactly 3 prime factors. The choice of $X$ does not impact the correctness of what follows.

The large preproduct regime constructs $P$ in a prime-by-prime fashion. Here, the tabulation regime is further distinguished by the count of primes in a preproduct. We considered $\omega(n) = d$ for $4 \le d \le 13$: ten different tabulations. We give pseudocode for the $d = 5$ algorithm and then give a unifying description of all the tabulation regimes.

**Algorithm 2** Carmichael numbers $n < B$ with $\omega(n) = 5$

**Require:** The tabulation bound $B$ and the cross over bound $X$

**for each** $q_1$ in $(1, \sqrt[5]{B})$ **do**
  Let $P_1 = q_1$
  **for each** $q_2$ admissible to $P_1$ in $(q_1, \sqrt[4]{B/P_1})$ **do**
    Let $P_2 = P_1 q_2$
    **for each** $q_3$ admissible to $P_2$ in $(\max\{q_2, X/P_2\}, \sqrt[3]{B/P_2})$ **do**
      Let $P_3 = P_2 q_3$
      **for each** $q_4$ admissible to $P_3$ in $(q_3, \sqrt{B/P_3})$ **do**
        Call **SPC** on $P_4 = P_3 q_4$
      **end for**
    **end for**
  **end for**
**end for**

Note $P_3$, which has enough room for two more primes, exceeds $X$. In this way, we do not duplicate work associated to the $P < X$ regime.

In general, the outer loops correspond to creating preproducts missing more than one prime: for $k > 1$ and at the $d - k$ level, a search commences for primes $q$ in $(p, \sqrt[k]{B/P_{d-k}})$ admissible to $P_{d-k}$. At the $d - 3$ level, we need to ensure that the

preproduct constructed exceeds $X$. Then every distinct tabulation regime always had three specific levels:

(1) The $d-3$ level: $P_{d-3}$ searches for $q$ up to $\sqrt[3]{B/P_{d-3}}$. The starting point for $q$ is the larger of $X/P_{d-3}$ or $p$ to ensure $P_{d-2} = P_{d-3}q > X$.
(2) The $d-2$ level: $P_{d-2} > X$ searches for $q$ in $(p, \sqrt{B/P_{d-2}})$.
(3) The $d-1$ level is the inner-loop and calls **SPC**.

A given preproduct $P$ may appear in multiple tabulation regimes. If $P$ has enough room for three more primes, then it also has enough room for two more primes and one more prime. However, it is important to know when a given preproduct will have only enough room for one more prime. If $P_{d-1} = P_{d-2}q$ where $q \geq \sqrt[3]{B/P_{d-2}}$, then $P_{d-1}$ can only be completed with a single prime.

4.4. **Observed Problems with Prior Tabulations.** In [38], we showed that there were faster algorithms available for the **SPC** step. Timing information showed that these methods were ineffective. The common case was $P_{d-1}\lambda(P_{d-1}) > B$. When this happened, **SPC** frequently consisted of two steps: compute $r^\star$ and discard $Pr^\star$ for being too large. There were other downstream effects. First, the cost of getting to the inner loop exceeded the work done in the inner loop. This meant that future implementations would be focused on the creation of cyclic products. This is another hint that our approach was not optimal; the algorithmic realization of the counting argument of these products is not done via a prime-by-prime approach. Second, this made parallel distribution of the work difficult. We chose to distribute at the $\lfloor d/2 \rfloor$ level which was an ad hoc choice made in an attempt to balance two key problems. Distributing it higher up was unbalanced and distributing it further down duplicated too much work.

We summarize the two tabulation regimes in the following way.

(1) The small case: "Tabulate all $n = PR$ with $P < X$, and $\omega(R) = 2$."
(2) The large case: "Tabulate all $n = PR$ with $P > X$, and $\omega(R) = 2$."

The former step had run-time roughly quadratic in $X$ and the latter had run-time matching Knödel's bound. The small case removed an asymptotically small set of inputs from the large case. However, completing these inputs as if they were in the large case would have been expensive. We conclude with a practical example to demonstrate the power of Algorithm 1.

**Example 2.** *Let* $P = 101 \cdot 103 \cdot 107 \cdot 109 \cdot 113 \cdot 127$ *where* $\lambda(P) = 68115600$ *and* $B = 10^{24}$. *How can we find all* $n = PR < B$*? We now have two options:*

*(1) Prime-by-prime appending in* $7 \leq d \leq 11$ *tabulations:*
  *(a) For* $d = 7$*:* $P$ *is the only preproduct at the* $d-1$ *level.*
  *(b) For* $d = 8$*:* $q$ *searched for in the interval* $(127, 757834)$ *and there are* 57255 *products* $P_{d-1} = Pq$.
  *(c) For* $d = 9$*:* 1145658 *products* $P_{d-1} = Pqr$.
  *(d) For* $d = 10$*:* 1227386 *products* $P_{d-1} = Pqrs$.
  *(e) For* $d = 11$*:* 24849 *products* $P_{d-1} = Pqrst$.
  *The* ***SPC*** *algorithm is invoked* 2455149 *times and there is significant overhead costs in creating those inputs.*
*(2) There are only* 8431 *candidates* $n = P(r^\star + k\lambda(P)) < B$. *Simply invoke Algorithm 1 on each candidate.*

Simply invoking Algorithm 1 raises new problems. Part of the prime-by-prime approach is forced on us for reasons of correctness. In order to invoke Algorithm 1 at scale, we will need a new way to account for the correctness of our algorithm. We address these issues in the next section.

## 5. Asymptotically faster algorithms

Example 2 illustrates the core inefficiency of the prime-by-prime approach: a preproduct generating 8431 candidates via Algorithm 1 instead generates over 2.4 million **SPC** invocations. This inefficiency manifests in two problems identified above. First, the common case of $P_{d-1}\lambda(P_{d-1}) > B$ means that $\mathcal{R}$ has only one element. So, the inner loop frequently does negligible useful work relative to the overhead of reaching it. An asymptotically faster algorithm must avoid these unproductive preproducts entirely. Second, the prime-by-prime structure does not naturally expose a well-balanced partition of the work, forcing the ad hoc distribution at the $\lfloor d/2 \rfloor$ level. A better algorithm should admit a parallelization scheme that arises naturally from its structure rather than as an ad hoc imposition.

Algorithm 1 resolves both problems by shifting the problem from "find all primes $r$ such that $Pr$ is a Carmichael number" to "find all $R$ in the arithmetic progression $\mathcal{R}$ such that $PR$ is a Carmichael number." This reduces the count of candidates from $\mathcal{K}(B)$ to $\mathcal{E}(B)$, and it exposes a natural partition of the work that is well-suited to parallel distribution. The two tabulation regimes of Subsection 4.4 are recast as follows:

(1) The small case: "Tabulate all $n = PR$ with $P < X$ and $\omega(R) = 2$."
(2) The large case: "Tabulate all $n = PR$ with $P < X$ and $\omega(R) \geq 3$."

This conceptual framing of the problem would not have made sense to us prior to discovering Algorithm 1; the way to achieve $\omega(R) \geq 3$ was with prime-by-prime appending. While Algorithm 1 can simply be invoked on every candidate, we use Section 5.1 to state an improved version based on sieving to reduce the amount of work required. Then we show how to partition the tabulation in a way suitable for parallel processing in Section 5.2. Sections 5.3 and 5.4 contain the implemented algorithm and the heuristically optimal algorithm.

### 5.1. $\lambda(P)$-sieving.

Previously, tabulation algorithms constructed $P$ in factored form and the search was for an $R$ that was prime. The naive thing to do would be to check $\mathcal{R}$ for primes. Using Algorithm 1, it is almost as efficient (the $\ln n$ factors may differ) to search for composite $R$. Since the elements of $\mathcal{R}$ are an arithmetic progression, with common difference $\lambda(P)$, we may sieve to filter out many candidates. This is a significant empirical improvement because sieving is much cheaper than modular exponentiation.

Use a bit vector of size $\lceil B/(P\lambda(P)) \rceil$ where the bit at index $k$ represents the number $P(r^\star + k\lambda(P))$. If we require the primes dividing $R$ to exceed $p$, then we may sieve by any prime $q \leq p$ (including the prime divisors of $P$). We can also sieve by any number inadmissible to $P$. This includes primes inadmissible to $P$, or non-cyclic odd composite numbers (see A139392 in OEIS).

We refer to sieving $\mathcal{R}$ prior to invoking Algorithm 1 for Carmichael numbers as a $\lambda(P)$-**sieve**.

**Example 2 (cont'd).** *Let* $P = 101\cdot103\cdot107\cdot109\cdot113\cdot127$ *where* $\lambda(P) = 68115600$ *and* $B = 10^{24}$. *There are only* 8431 *candidates* $n = P(r^\star + k\lambda(P)) < B$. *There are* 3886 $k$ *values that have* $r^\star + k\lambda(P)$ *divisible by a prime less than* 129 *which are found with* $\lambda(P)$*-sieving. So, Algorithm 1 is only invoked on* 4545 *candidates.*

5.2. **Partitioning Carmichael numbers by preproducts.** Let $(P, \lambda(P), b)$ be the 3-tuple representing the set of Carmichael numbers of the form $n = PR$ where the prime divisors of $R$ exceed $b$ or $R = 1$. Observe a partition of Carmichael numbers:

(1) $(3, 2, 5)$: $n$ is a multiple of 3 and the other primes exceed 5.
(2) $(5, 4, 5)$: $n$ is a multiple of 5 and the other primes exceed 5.
(3) $(15, 4, 5)$: $n$ is a multiple of 15 and the other primes exceed 5.
(4) $(1, 1, 5)$: $n$ has least prime factor exceeding 5.

Our set of jobs is of the form $(P, \lambda(P), \max\{X/P, p\})$ where $P < X$ and $Pp^3 < B$. These are the preproducts that can allow three or more primes to be appended. This is a subset of the preproducts that would have been found at the $P_{d-3}$ level. The key idea is that once we append a prime to a given $P$ we do not want to create a preproduct already in the initializing set. Either $p$ is large enough so that $Pp > X$ or we set the append bound to $X/P$ so that the first prime appended to $P$ causes the preproduct to exceed $X$.

5.3. **An improved tabulation algorithm.** We call the following algorithm on each such preproduct.

**Algorithm 3** Carmichael numbers $PR$ with $\omega(R) \geq 3$

**Require:** The tabulation bound $B$, crossover bound $X$, $P$, and $\lambda(P)$
1: **if** $P\lambda(P)^2 > B$ **then**
2: $\lambda(P)$**-sieve**
3: **else**
4: **for each** prime $q$ admissible to $P$ in $(\max\{p, X/P\}, \sqrt[3]{B/P})$ **do**
5: Recursively call this algorithm with preproduct $Pq$
6: **end for**
7: **if** $P > X$ **then**
8: **for each** prime $q$ admissible to $P$ in $(\sqrt[3]{B/P}, \sqrt{B/P})$ **do**
9: Call **SPC** on $Pq$
10: **end for**
11: **if** $P > pX$ **then**
12: Call **SPC** on $P$
13: **end if**
14: **end if**
15: **end if**

**Theorem 12.** *Algorithm 3 is correct.*

*Proof.* The branching rule found in line 1 is irrelevant for the correctness of this algorithm but is important for the run-time. Each branch is capable of independently producing a correct tabulation.

The first branch (line 2) will produce a correct tabulation because it is an exhaustive search.

The prime-by-prime branch (lines 4-14) treats a given preproduct having more than two primes to append (lines 4-6), having exactly two primes to append (lines 8-10), and as having exactly one prime to append (line 12). So this finds all possible completions in a prime-by-prime manner. □

**Remark:** The $\lambda(P)$-sieving branch can find $R$ with $\omega(R) < 3$. If it is necessary to avoid this we could discard duplicates in the post-processing sorting of the entire tabulation or we could track the recursion depth.

Since line 1 of Algorithm 3 played no role in the correctness of the algorithm, its utility is as a heuristic to complete a given preproduct as efficiently as possible. The heuristic comes from balancing $B/(P\lambda(P))$ (a crude linear estimate of $\lambda(P)$-sieving) against $\sqrt{B/P}$ (a crude estimate of the cost of the prime-by-prime branch).

We forgo a traditional analysis of this algorithm in order to state a more optimal algorithm but provide a sketch that Algorithm 3 likely matches Erdős' initial asymptotic bound. By letting $X = B^{1/2-\epsilon}$ and unconditionally entering the first preproduct expansion step (the else clause on line 3), each preproduct is expanded into a new preproduct. This set of preproducts satisfies $B^{1/2} < P < B^{2/3}$. If each of these is then used in the $\lambda(P)$-sieve branch, the cost is

$$\sum_{B^{1/2}<P<B^{2/3}} \frac{B}{P\lambda(P)} < B \exp\left\{\frac{-k \ln B \ln_3 B}{\ln_2 B}\right\},$$

for some positive constant $k$. The sum and inequality match those Erdős used to bound the count of Carmichael numbers in [9].

5.4. **A Heuristically Optimal Algorithm.** Counting Carmichael numbers in the arithmetic progression $P(r^\star + k\lambda(P))$ is central to the argument in the proof of Theorem 2. Since Algorithm 1 (in conjunction with $\lambda(P)$-sieving) allows efficient identification of Carmichael numbers in that arithmetic progression, the counting argument has an algorithmic realization.

**Theorem 3.** *For each $\epsilon > 0$, there is an $B_0(\epsilon)$ such that for all $B > B_0(\epsilon)$, there is an algorithm that can find all Carmichael numbers less than $B$ in at most $\mathcal{E}(B)$ queries of "Is $n$ a Carmichael number?".*

*Proof.* There are three sets of Carmichael numbers $n \le B$ that are counted. We show that there exists an algorithm that tabulates each of these sets in time proportional to the size of the set times the cost of answering the query, "Is $n$ a Carmichael number?" Since this cost is polynomial time, it remains to show that the inputs can be generated as claimed. The three sets are

(1) $n \le B^{1-\delta}$,
(2) $B^{1-\delta} < n \le B$ and $n$ has a prime factor $p \ge B^\delta$, and
(3) $B^{1-\delta} < n \le B$ and all the prime factors of $n$ are below $B^\delta$.

Setting $\delta = \epsilon/5$ completes the proof of Theorem 2. Here we continue describing sets in terms of the parameter $\delta$.

(1) The first set represents the small Carmichael numbers. This is a tabulation problem with a smaller upper bound. This algorithm is recursive and we apply the method of the other two sets to find these numbers if necessary[3].

[3]In reality, this set may be provided by some prior tabulation.

(2) The count of these Carmichael numbers is bounded by considering sets $P(r^\star + k\lambda(P)) < B$ indexed by $k$, where $P = p$ is a prime exceeding $B^\delta$.

Algorithmically, use a sieve to find primes in $(B^\delta, \sqrt{B}]$. For each prime $P > B^\delta$, use $\lambda(P)$-sieving. The total cost is bounded by

$$\sum_{P>B^\delta} \frac{B}{P(P-1)} < 2B^{1-\delta}$$

invocations of Algorithm 1 which is a lower order cost compared to set 3.

(3) The count of these Carmichael numbers is bounded by considering $P(r^\star + k\lambda(P)) < B$, where $B^{1-2\delta} < P \leq B^{1-\delta}$ are cyclic numbers that are $B^\delta$-smooth.

Algorithmically, we could use a sieve on this interval to find the $B^\delta$-smooth numbers and this would suffice for the purposes of our proof. However, we recommend constructing the set explicitly. This is the prime-by-prime construction technique but restricted to a table of primes less than $B^\delta$. In this way, we could have early-abort by not proceeding whenever $P\lambda(P) > B$.

Similar to the previous part, for each $P$ we use $\lambda(P)$-sieving. The number of candidates created here is $B\exp(-(1-\epsilon)\ln x \cdot \ln_3 x/\ln_2 x)$. And this is what contributes the asymptotic cost given by the work in [34].

This demonstrates that the upper bound can be algorithmically realized. □

If the bound in Theorem 2 is also a lower bound, as a heuristic argument shows it is, then the algorithm in the proof of Theorem 3 is optimal up to the cost of answering the query "Is $n$ a Carmichael number?"

We note two features about this theoretical algorithm. First, only $\lambda(P)$-sieving is used. Second, since only $\lambda(P)$-sieving is used, it is easily adapted to tabulating Carmichael numbers in an interval: choose the $k$ so that $P(r^\star + k\lambda(P))$ only lands in the given interval. It has been traditional to re-do the entire tabulation as an independent check of prior work. Suppose we want to forgo this tradition. If we trust the correctness of the current tabulation, then we may provide a tabulation of all Carmichael numbers up to $10^{25}$ by tabulating Carmichael numbers in the interval $[10^{24}, 10^{25}]$.

## 6. Carmichael numbers up to $10^{24}$

The large case for the $10^{22}$ computation from [38] took about 104 days on 32 Xeon E5-2630 processors (192 total cores). This was the prime-by-prime appending routine that had different tabulation regimes for each distinct prime count. In contrast, Algorithm 3 completed the $10^{24}$ case in about 97 days on 5 AMD EPYC 9734 processors (560 total cores).

In terms of core hours, we achieved a bound that was 100 times greater with about 2.72 times the number of core hours. This is not a fair comparison since the two processors are several generations apart[4]. We suspect the recent computation did about 5 times more work to achieve a bound 100 times greater. The 5 could possibly be even lower because the parameter $X$ was more favorable to the $10^{22}$

[4]PassMark (`https://tinyurl.com/m3nm8vdy`) shows each EPYC 9734 thread performs about 1.8 times the work of one E5-2630 thread.

| $B$ | Prime by prime | Algorithm 3 |
|---|---|---|
| $10^{14}$ | 64 | 140 |
| $10^{15}$ | 397 | 711 |
| $10^{16}$ | 2456 | 3631 |
| $10^{17}$ | 15361 | 18352 |
| $10^{18}$ | 95476 | 92891 |

TABLE 1. Timing comparison (in seconds)

tabulation. That is, for $B = 10^{22}$ we had $X = 7 \cdot 10^7 \approx B^{0.3566}$, and for $B = 10^{24}$ we had $X = 1.25 \cdot 10^8 \approx B^{0.3373}$.

Serial tabulations were performed by the codebase from [36] and the new codebase [39] on a single core of an AMD Ryzen 9 3900X processor. The problem solved was a large case tabulation; for all bounds $B$ we set $X = B^{1/3}$ and timed the tabulation of Carmichaels between $X$ and $B$. Table 1 shows that the asymptotic crossover is realized by $B = 10^{18}$. Even more encouraging to note is that Algorithm 3 has less duplicated work in its parallelization scheme than the ad hoc parallel distribution for the prime-by-prime approach.

6.1. **Statistics on the tabulation.** In the tables, we provide various statistics and counts on the tabulation. Our tables omit the lower order of magnitudes, which may be found in the previous literature or in [42, Section 7.2].

We start with the counts of Carmichael numbers by order of magnitude. Table 2 contains these counts, and four supplementary functions related to the lower-bound heuristic counting function. In [33], a more precise version of $\mathcal{E}(B)$ was stated:

$$B \exp\left\{-\frac{\ln B}{\ln_2 B}\left(\ln_3 B + \ln_4 B + \frac{\ln_4 B - 1}{\ln_3 B} + O\left(\left(\frac{\ln_4 B}{\ln_3 B}\right)^2\right)\right)\right\},$$

which served as an upper bound and a heuristic lower bound. We provide four numeric quantities associated with the above heuristic.

The first function is the approximation $C(B) = B^{j_1(B)}$. The expectation is that $j_1(B) \to 1$ as $B \to \infty$. The second function is $j_2(B)$, where

$$C(B) = B^{1-j_2(B)(\ln_3 B + \ln_4 B)/\ln_2 B}.$$

The expectation is that $j_2(B) \to 1$ as $B \to \infty$. The third is $j_3(B)$, where

$$C(B) = B \exp\left\{-\frac{j_3(B) \ln B \ln_3 B}{\ln_2 B}\right\}.$$

The expectation is that $j_3(B) \to 1$ as $B \to \infty$. Finally, consider the function $j_4(B)$ where,

$$C(B) = B \exp\left\{-\frac{\ln B}{\ln_2 B}\left(\ln_3 B + \ln_4 B + j_4(B)\right)\right\}.$$

The expectation is that $j_4(B) \to 0$ as $B \to \infty$.

The functions $j_3$ and $j_4$ appear as $j$ and $k$ in the table in [33]. That table went up to $25 \cdot 10^9$. In that table and for the next few orders of magnitude, their values are trending in the expected direction. Thereafter, these two quantities trend away from their expected values. It is unclear whether $j_3$ or $j_4$ can serve in a predictive way given their divergence from expected asymptotic behavior. If

| $B$ | $C(B)$ | $j_1(B)$ | $j_2(B)$ | $j_3(B)$ | $j_4(B)$ |
|---|---|---|---|---|---|
| $10^{16}$ | 246683 | 0.337008 | 1.561018 | 1.864056 | 0.859364 |
| $10^{17}$ | 585355 | 0.339259 | 1.551898 | 1.864722 | 0.861721 |
| $10^{18}$ | 1401644 | 0.341479 | 1.543801 | 1.865223 | 0.863922 |
| $10^{19}$ | 3381806 | 0.343639 | 1.536592 | 1.865646 | 0.866053 |
| $10^{20}$ | 8220777 | 0.345745 | 1.530103 | 1.865977 | 0.868077 |
| $10^{21}$ | 20138200 | 0.347810 | 1.524189 | 1.866191 | 0.869947 |
| $10^{22}$ | 49679870 | 0.349826 | 1.518777 | 1.866320 | 0.871694 |
| $10^{23}$ | 123381982 | 0.351793 | 1.513795 | 1.866375 | 0.873323 |
| $10^{24}$ | 308279939 | 0.353706 | 1.509200 | 1.866382 | 0.874867 |

TABLE 2. Count of Carmichael numbers and numerical estimates

| $B$ | $d=3$ | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|
| $10^{16}$ | 10816 | 16202 | 55012 | 86696 | 60150 | 16348 |
| $10^{17}$ | 19539 | 25758 | 100707 | 194306 | 172234 | 63635 |
| $10^{18}$ | 35586 | 40685 | 178063 | 414660 | 460553 | 223997 |
| $10^{19}$ | 65309 | 63343 | 306310 | 849564 | 1159167 | 720406 |
| $10^{20}$ | 120625 | 98253 | 514381 | 1681744 | 2774702 | 2148017 |
| $10^{21}$ | 224763 | 151566 | 846627 | 3230120 | 6363475 | 6015901 |
| $10^{22}$ | 420658 | 232742 | 1370257 | 6034046 | 14056367 | 16005646 |
| $10^{23}$ | 790885 | 357078 | 2181324 | 11007792 | 30038569 | 40696220 |
| $10^{24}$ | 1494738 | 548265 | 3429390 | 19665194 | 62338997 | 99469866 |

TABLE 3. Counts by number of prime factors $d < 9$

anything might be conjectured from $j_3$ and $j_4$, it would be that the lower order terms in the exponential expression are incorrect as lower bounds; we believe the table is too small for them to behave consistently with the asymptotic expectations. To make predictions we will have to use $j_1$ or $j_2$.

In [11], the authors speculate that $C(B)$ will not exceed $B^{1/2}$ for any $B < 10^{100}$. Based on the data above, we conjecture that the threshold will not be crossed while $B < 10^{150}$. Justification for this conjecture will depend specifically on $j_2$. First note that the rate of decrease for $j_2$ is itself decreasing; $j_2$ decreased by $0.00591, 0.00541, 0.00498$, and $0.00459$ across the last 5 orders of magnitude. If $C(10^{100}) = 10^{50}$ then $j_2(10^{100}) \approx 1.2248$ and if $C(10^{150}) = 10^{75}$ then $j_2(10^{150}) \approx 1.2521$. Meeting 1.2248 across 76 orders of magnitude yields an average decrease of 0.0037, while meeting 1.2521 across 126 orders of magnitude yields an average decrease of 0.002. We believe that the latter is more likely, as it is further away from the smallest observed difference of 0.00459.

Tables 3 and 4 contain the distribution of Carmichael numbers by the count of prime factors. The columns represent the prime factor counts and the rows are by orders of magnitude.

In [11], it is conjectured that $C_k(B) > C_{k+1}(B)$ should hold once $B$ is large enough. At this range, the inequality only holds for $k = 3$ and $k > 8$. We conjecture that we are in the asymptotic domain for $k = 3$; that is, we think this inequality will not reverse again. We also think that we are close to $C_3(B) > C_5(B)$ and that this inequality will be obtained before $C_4(B) > C_5(B)$ is obtained. We do not think

| $B$ | $d = 9$ | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|
| $10^{16}$ | 1436 | 23 | 0 | 0 | 0 | 0 |
| $10^{17}$ | 8835 | 340 | 1 | 0 | 0 | 0 |
| $10^{18}$ | 44993 | 3058 | 49 | 0 | 0 | 0 |
| $10^{19}$ | 196391 | 20738 | 576 | 2 | 0 | 0 |
| $10^{20}$ | 762963 | 114232 | 5804 | 56 | 0 | 0 |
| $10^{21}$ | 2714473 | 547528 | 42764 | 983 | 0 | 0 |
| $10^{22}$ | 8939435 | 2347828 | 262818 | 10018 | 55 | 0 |
| $10^{23}$ | 27660029 | 9178659 | 1388714 | 81434 | 1277 | 1 |
| $10^{24}$ | 81041495 | 33199094 | 6529921 | 547590 | 15328 | 61 |

TABLE 4. Counts by number of prime factors $d \geq 9$

we are in the asymptotic range for the $k > 8$ inequalities and that these inequalities exist only because $B$ is so small that we have not found enough Carmichael numbers with $k > 8$ prime factors. We conjecture that each of these inequalities (for $k > 8$) will reverse (before reversing again). There is not enough data in the table for us to feel comfortable conjecturing when $C_4(B) > C_5(B)$.

One question explored in [11] regards imprimitive Carmichael numbers. The motivation was Chernick's family where if $6m + 1, 12m + 1$, and $18m + 1$ are all prime then their product is a Carmichael number. This example may be generalized. Every Carmichael number may be used to generate an infinite family of Carmichael numbers by considering the successive ratios of $p_i - 1$ for all primes dividing $n$ under the assumption that these ratios may be achieved by prime-tuples infinitely often. The least Carmichael number in a given family is primitive. For example, $7 \cdot 13 \cdot 19$ and $37 \cdot 73 \cdot 109$ both have the same set of successive ratios (arising from Chernick's family) but only the first is primitive.

In [11, Corollary 4], the authors prove that the set of imprimitive Carmichael numbers is $o(B^{1/3})$. In both [14, 23], the authors establish that the count of all Carmichael numbers exceeds $B^{1/3}$. Combining these two results shows that the set of imprimitive Carmichael numbers forms a relative 0-density subset of all Carmichael numbers[5]. We extend the table of imprimitive Carmichael numbers found in [11]. The column labeled "%" contains the percentage of Carmichael numbers that are imprimitive. These percentages are decreasing and these numbers make up $\approx 0.43\%$ our total tabulation, which is concordant with a 0-density subset. There are a total of 1312179 imprimitive Carmichael numbers less than $10^{24}$.

## 7. Related Problems

7.1. **Future implementation.** There are two obvious implementation improvements if someone were to continue tabulating Carmichael numbers. First, the heuristically optimal algorithm could be tried and compared to our Algorithm 3. Second, a switch away from GMP's arbitrary precision arithmetic to 128-bit arithmetic could find significant performance increases. At the time of our implementation, we did not know of any library that could perform modular exponentiations faster than GMP in our range. Afterwards, we discovered Jeffrey Hurchalla's library [17]; it seems to run about 40% faster for inputs in our range.

[5]We thank Carl Pomerance for pointing this out.

| $B$ | % | $d=3$ | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| $10^{16}$ | 2.921 | 6615 | 563 | 30 | 0 | 0 |
| $10^{17}$ | 2.363 | 12725 | 1036 | 72 | 1 | 0 |
| $10^{18}$ | 1.880 | 24396 | 1797 | 156 | 4 | 0 |
| $10^{19}$ | 1.489 | 46877 | 3193 | 264 | 8 | 1 |
| $10^{20}$ | 1.169 | 89854 | 5745 | 501 | 21 | 1 |
| $10^{21}$ | 0.915 | 173331 | 10070 | 903 | 48 | 2 |
| $10^{22}$ | 0.713 | 334737 | 17770 | 1617 | 91 | 3 |
| $10^{23}$ | 0.552 | 647265 | 30963 | 2930 | 174 | 5 |
| $10^{24}$ | 0.426 | 1253176 | 53564 | 5097 | 331 | 11 |

TABLE 5. Counts of imprimitive Carmichael numbers

7.2. **Absolute Lucas pseudoprimes.** A generalization of Carmichael numbers is to change the underlying divisibility sequence to Lucas sequences [43]. In [16], it was shown that several of the themes of [27, 37] can be applied to tabulating these numbers and it seems natural that the results in this paper could be extended as well. In particular, we believe that these numbers can also be factored efficiently akin to Algorithm 1 due to an identity comparable to the difference of squares factorization:

$$U_{2^s n'}(A,B) = V_{n'}(A,B)\prod_{i=0}^{s-1} U_{2^i n'}(A,B),$$

where $U_k(A,B)$ and $V_k(A,B)$ take their usual meanings for Lucas sequences and $n'$ is the odd part of $n-\delta_n$ with $\delta_n$ being the Jacobi symbol of $n$ with respect to the discriminant of the sequence.

7.3. **Lehmer's totient problem.** Lehmer asked if there is a composite number $n$ such that $\phi(n) \mid (n-1)$. We believe that it should be possible to adopt the themes in this paper to design an algorithm to tabulate all examples less than $B$ in time $\widetilde{O}(B^{1/2})$ [32, 24].

7.4. **Smallest Carmichael numbers with a fixed count of prime factors.** While we claim that our current work supersedes the prior work, that does not mean the prior work is irrelevant. Using the prime-by-prime approach in conjunction with composite completions (stopping when $P\lambda(P) > B$), undergraduates at Butler University, Aidan Johnson and Sylvia Webster, were able to find the smallest Carmichael numbers with exactly $36 \le k \le 39$ prime factors. The implementation used a heuristic guess for $B$ and updated to a smaller upper bound if a new candidate Carmichael number was found. In this way we found the following numbers, extending Pinch's list by four entries:

(36) 880764796017323940654909652310158381246975294486239884339113306911241440 1
(37) 643424474668062745669025988900909119939356239127872348810496972952352967200 1
(38) 705874549382005220484659306969292831350747103767545428701712197729970811920000 1
(39) 247501859720155346004984972686963660468745572881589838402324186461731549219433760 1

## DATA AVAILABILITY

This project generated an 18.4 GB sorted text file of Carmichael numbers up to $10^{24}$. Each line contains a Carmichael number followed by its prime factorization.

- `https://blue.butler.edu/~jewebste/new_table.txt`

Gustavo Bravo, an undergraduate at the University of Calgary, built a database for divisibility queries.

- `https://blue.butler.edu/~jewebste/car/`

## Acknowledgements

We thank Richard Pinch, Carl Pomerance, Andrew Fiori, Ha Tran, Michael Jacobson, Renate Scheidler, Jon Sorenson, and the three anonymous referees for helpful comments that have made this paper better. Paul Pollack was especially helpful in clarifying some key points in Section 3. Thank you to Drew Sutherland and the Simons Collaboration in Arithmetic Geometry, Number Theory, and Computation for providing computational support. Thanks also to Frank Levinson for generous donations to Butler University that helped make the tabulation possible.

## References

1. M. Agrawal, N. Kayal, and N. Saxena, *PRIMES is in P*, Ann. of Math. (2) **160** (2004), no. 2, 781–793. MR 2123939
2. W. R. Alford, Andrew Granville, and Carl Pomerance, *There are infinitely many Carmichael numbers*, Ann. of Math. (2) **139** (1994), no. 3, 703–722. MR 1283874 (95k:11114)
3. Pierre Beauchemin, Gilles Brassard, Claude Crépeau, Claude Goutier, and Carl Pomerance, *The generation of random numbers that are probably prime*, J. Cryptology **1** (1988), no. 1, 53–64. MR 935901
4. N. G. W. H. Beeger, *On composite numbers n for which* $a^{n-1} \equiv (\mathrm{mod}\, n)$ *for every a prime to n*, Scripta Math. **16** (1950), 133–135. MR 36773
5. R. D. Carmichael, *Note on a new number theory function*, Bull. Amer. Math. Soc. **16** (1910), no. 5, 232–238. MR 1558896
6. Don Coppersmith, Nick Howgrave-Graham, and S. V. Nagaraj, *Divisors in residue classes, constructively*, Math. Comp. **77** (2008), no. 261, 531–545. MR 2353965
7. Richard Crandall and Carl Pomerance, *Prime numbers*, second ed., Springer, New York, 2005, A computational perspective. MR 2156291
8. H. J. A. Duparc, *On Carmichael numbers*, Simon Stevin **29** (1952), 21–24. MR 48492
9. P. Erdös, *On pseudoprimes and Carmichael numbers*, Publ. Math. Debrecen **4** (1956), 201–206. MR 79031
10. C. Goutier, *Readme for* $10^{22}$ *tabulation*, 2022, `https://www-labs.iro.umontreal.ca/~goutier/OEIS/A055553/readme.text`.
11. Andrew Granville and Carl Pomerance, *Two contradictory conjectures concerning Carmichael numbers*, Math. Comp. **71** (2002), no. 238, 883–908. MR 1885636
12. A. Guthmann, *On the computation of Carmichael numbers*, Tech. Report 218, Universität Kaiserslautern, 1992, `https://kluedo.ub.rptu.de/frontdoor/deliver/index/docId/5039/file/Guthmann_On+the+computation+of+Carmichael+numbers.pdf`.
13. G. Harman, *On the number of Carmichael numbers up to x*, Bull. Lond. Math. Soc. **37** (2005), no. 5, 641–650. MR 2164825
14. ———, *Watt's mean value theorem and Carmichael numbers*, Int. J. Number Theory **4** (2008), no. 2, 241–248. MR 2404800
15. S. Hartman and J. P. Sorenson, *Reducing the space used by the sieve of Eratosthenes when factoring*, Inform. Process. Lett. **188** (2025), Paper No. 106537, 4. MR 4815590
16. C. Helmreich and J. Webster, *Tabulating absolute Lucas pseudoprimes*, Integers **24A** (2024), Paper No. A8, 18. MR 4750800
17. Jeffrey Hurchalla, *Clockwork modular arithmetic library*, `https://github.com/hurchalla/modular_arithmetic`, 2025.
18. Gerhard Jaeschke, *The Carmichael numbers to* $10^{12}$, Math. Comp. **55** (1990), no. 191, 383–389. MR 1023763
19. W. Keller, *The Carmichael numbers to* $10^{13}$, Abstracts Amer. Math. Soc. **9** (1988), 328–329.

20. Walter Knödel, *Eine obere Schranke für die Anzahl der Carmichaelschen Zahlen kleiner als* $x$, Arch. Math. **4** (1953), 282–284. MR 57903
21. H. W. Lenstra, Jr., *Divisors in residue classes*, Math. Comp. **42** (1984), no. 165, 331–340. MR 726007
22. H. W. Lenstra, Jr. and C. Pomerance, *Primality testing with Gaussian periods*, J. Eur. Math. Soc. (JEMS) **21** (2019), no. 4, 1229–1269. MR 3941463
23. J. D. Lichtman, *Primes in arithmetic progressions to large moduli, and shifted primes without large prime factors*, 2022, `https://arxiv.org/abs/2211.09641`.
24. Florian Luca and Carl Pomerance, *On composite integers n for which* $\phi(n)|n-1$, Bol. Soc. Mat. Mexicana (3) **17** (2011), no. 1, 13–21. MR 2978700
25. G. L. Miller, *Riemann's hypothesis and tests for primality*, J. Comput. System Sci. **13** (1976), no. 3, 300–317. MR 480295
26. Louis Monier, *Evaluation and comparison of two efficient probabilistic primality testing algorithms*, Theoret. Comput. Sci. **12** (1980), no. 1, 97–108. MR 582244
27. R. G. E. Pinch, *The Carmichael numbers up to* $10^{15}$, Math. Comp. **61** (1993), no. 203, 381–391. MR 1202611
28. ———, *The Carmichael numbers up to* $10^{16}$, March, 1998, `arXiv:math.NT/9803082`.
29. ———, *The Carmichael numbers up to* $10^{17}$, April, 2005, `arXiv:math.NT/0504119`.
30. ———, *The Carmichael numbers up to* $10^{18}$, April, 2006, `arXiv:math.NT/0604376`.
31. ———, *The Carmichael numbers up to* $10^{21}$, May, 2007, `s369624816.websitehome.co.uk/rgep/p82.pdf`.
32. Carl Pomerance, *On composite n for which* $\varphi(n) \mid n-1$. *II*, Pacific J. Math. **69** (1977), no. 1, 177–186. MR 434938
33. ———, *On the distribution of pseudoprimes*, Math. Comp. **37** (1981), no. 156, 587–593. MR 628717
34. 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 (82g:10030)
35. Michael O. Rabin, *Probabilistic algorithm for testing primality*, J. Number Theory **12** (1980), no. 1, 128–138. MR 566880 (81f:10003)
36. Andrew Shallue, *Code for tabulating carmichael numbers*, `https://github.com/ashallue/tabulate_car/tree/master`, 2024.
37. Andrew Shallue and Jonathan Webster, *Tabulating Carmichael numbers* $n = Pqr$ *with small* $P$, Res. Number Theory **8** (2022), no. 4, Paper No. 84. MR 4493784
38. ———, *Advances in tabulating Carmichael numbers*, Res. Number Theory **11** (2025), no. 1, Paper No. 8, 16. MR 4836747
39. ———, *Carmichael algorithms*, `https://github.com/ashallue/CarmichaelAlgorithms/tree/main`, 2025.
40. J.D. Swift, *Review 13 – table of Carmichael numbers to* $10^9$, Math. Comp. **29** (1975), no. 129, 338–339.
41. Samuel S. Wagstaff, Jr., *Pseudoprimes and Fermat numbers*, Integers **24** (2024), Paper No. A23, 10. MR 4710424
42. Jonathan Webster, *Carmichael numbers: A computational perspective*, Master's thesis, University of Calgary, 2025.
43. H. C. Williams, *On numbers analogous to the Carmichael numbers*, Canad. Math. Bull. **20** (1977), no. 1, 133–143. MR 447099

(1) Illinois Wesleyan University

(2) Butler University