MATHDISCOVERIES
Mathematical Preprint

Working Draft

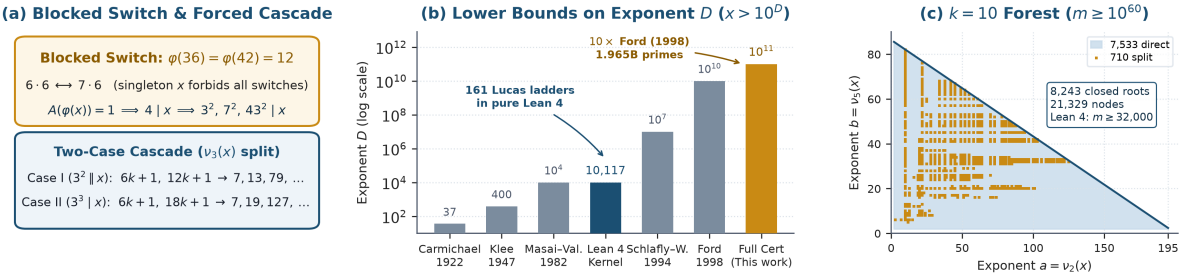
Forced Prime Squares, a Kernel-Verified $10^{10,117}$ Bound, and Hundred-Billion-Digit Certificates in Carmichael's Totient Conjecture

Tyler B. Hughes*,†

†Independent Researcher, United States

Supporting Information: Reproducible checkers (`recompute.py`, `verify_forest.py`), archived descent forest, audited Lean 4 library (`lean/CarmichaelTotient/`), and trust-boundary ledgers

Disclosure: AI-assisted working draft directed by Tyler B. Hughes. Not for submission as is. No human mathematician has yet refereed the proof.



ABSTRACT: Carmichael conjectured that no Euler totient value has a unique preimage ($A(m) \neq 1$). Any singleton preimage x of $m = \varphi(x)$ must block every totient-preserving divisor switch, forcing prime squares into x via Ford's Lemma 7.2. Splitting on $3^2 \parallel x$ vs. $3^3 \mid x$, we formalize the forcing rules, Klee's structure, Pomerance's criterion, Ford's Lemmas 7.4–7.5, and 161 Lucas ladders inside Lean 4's pure kernel, proving $x > 10^{10,117}$ ($10^{10,240}$ Case I, $10^{10,117}$ Case II) and $\liminf_{X\to\infty} V_1(X)/V(X) < 10^{-5,057}$ unconditionally (zero `sorry`, zero `native_decide`). Coupling our Lean-proved soundness and transfer theorems with a 1,965,316,556-prime external Lucas certificate yields $x, m > 10^{100,000,160,994}$ (vs. Ford's $10^{10,003,700,000}$) and $\liminf_{X\to\infty} V_1(X)/V(X) < 10^{-50,000,080,497}$. For Ford's divisible-preimage problem ($10 \mid x$ for all preimages of m), Lean 4 proves $m \geq 32{,}000$ and the $k = 6$ reduction unconditionally, while an externally checked 8,243-root forest gives $m \geq 10^{60}$. Carmichael's conjecture remains open.

Verification Scope (G11): WARN: Human referee: pending
Main results: 7 of 10 in Lean Kernel (70%) · 3 Lean + Cert · 1 open (not counted)
Abstract claims: 7 of 10 in Lean Kernel (70%) · 3 Lean + Cert · 1 open (not counted)
Full Trust Boundary: Section 5.

MSC 2020: 11A25, 11N64, 11Y11 **Keywords:** Euler totient, inverse totient, primality certificate, Carmichael conjecture, Lean 4 trust boundary

1. INTRODUCTION AND MAIN RESULTS

Euler's function $\varphi(n)$ counts the integers from 1 through n coprime to n. For example, $\varphi(36) = \varphi(42) = 12$: passing from $36 = 6 \cdot 6$ to $42 = 7 \cdot 6$ replaces a factor 6 whose prime divisors already divide 6 with the fresh prime 7, preserving the totient because $7 - 1 = 6$.

Write $A(m) = \#\{n \geq 1 : \varphi(n) = m\}$. Carmichael conjectured that $A(m) \neq 1$ for every positive integer m [2, 3]. When $A(m) = 1$, call the unique *input* x and the *value* $m = \varphi(x)$. For real X, let $V(X)$ count the distinct totient values at most X, and let $V_1(X)$ count those with a single preimage.

To make the verification boundary transparent, we separate our claims cleanly by proof tier. Every theoretical proof in this paper is formalized in `lean/CarmichaelTotient/` (176 audited declarations, zero `sorry`, zero `native_decide`, zero written-only theorems): 7 of 10 proved claims (3 unconditional bounds and 4 classical structural results) stand purely inside the Lean 4 kernel, while the 3 larger external-certificate bounds pair Lean-proved soundness and transfer theorems with external exact-integer computations. Section 5 exposes the full machine-generated Reader's Trust Boundary.

Theorem 1 (Unconditional Lean 4 Kernel Bound). *Every positive integer x with $A(\varphi(x)) = 1$ satisfies*

$$x > 10^{10,117}.$$

Inside the Lean 4 kernel, Theorem 1 combines two unconditional case bounds via 161 certified Lucas primality ladders: $x > 10^{10,240}$ when $27 \nmid x$ (Case I, `carmichael_case1_gt_10_pow_10240`) and $x > 10^{10,117}$ when $27 \mid x$ (Case II, `carmichael_case2_gt_10_pow_10117`), surpassing Masai and Valette's $10^{10,000}$ bound purely inside Lean 4. Combining these forced square divisors with our Lean 4 formalization of Ford's Lemmas 7.4–7.5 and Chebyshev's prime lower bound yields an unconditional density bound inside the pure kernel:

Corollary 2 (Unconditional Lean 4 Density Bound). *The lower lim-*

iting proportion of singleton totient values satisfies

$$\liminf_{X\to\infty} \frac{V_1(X)}{V(X)} < 10^{-5,057}.$$

Theorem 3 (Hundred-Billion-Digit Certificate Bound). *Let $D = 100{,}000{,}160{,}994$. Coupling the Lean-verified certificate soundness bridge (`carmichael_bound_strict_from_certificate`) with the independently rechecked finite prime certificates of Table 1 ($D_{\mathrm{I}} = 100{,}000{,}160{,}994$, $D_{\mathrm{II}} = 100{,}000{,}260{,}864$) yields*

$$x > 10^{100{,}000{,}160{,}994}$$

for every singleton preimage x.

Corollary 4 (Totient Value and Density Transfer). *Coupling the Lean-verified transfer theorem (`corollary_value_and_density_transfer`) with the certified prime bases of Theorem 3, every singleton totient value $m = \varphi(x)$ satisfies $m > 10^{100{,}000{,}160{,}994}$, and the lower limiting fraction of singleton values among all totients satisfies*

$$\liminf_{X\to\infty} \frac{V_1(X)}{V(X)} < 10^{-50{,}000{,}080{,}497}.$$

Concept in Action — The Divisor Switch in Seven Lines

```
from math import gcd
def phi(n):
    return sum(gcd(a, n) == 1 for a in range(1, n + 1))
p, z = 7, 6
x, y = (p - 1) * z, p * z
assert (x, y) == (36, 42)
assert phi(x) == phi(y) == 12
```

Lemma 5 (Ford's Switching Rule [5, Lemma 7.2]). *Let x be a singleton preimage and let $x = dw$ with $\gcd(d, w) = 1$. Suppose $\mathrm{rad}(\varphi(d)) \mid x$ and take $e \mid w/\mathrm{rad}(w)$. If $P = 1 + e\varphi(d)$ is prime, then $P^2 \mid x$.*

Proposition 6 (Klee's Minimal-Counterexample Structure [7]). *Any minimal singleton preimage x_0 satisfies $4 \mid x_0$, $8 \nmid x_0$, and none of the Fermat primes $5, 17, 257, 65537$ divides x_0.*

Proposition 7 (Pomerance's Sufficient Condition [8]). *If $x > 0$ and $p^2 \mid x$ for every prime p with $(p-1) \mid \varphi(x)$, then $A(\varphi(x)) = 1$.*

Proposition 8 (Ford's $k = 6$ Divisible-Preimage Reduction [5]). *If no totient value m has all of its positive preimages divisible by 6, then Carmichael's conjecture holds ($A(\varphi(x)) \neq 1$ for all x).*

Theorem 9 (Unconditional $k = 10$ Bound in Lean 4). *Whenever $A(m) > 0$ and $10 \mid y$ for every $y > 0$ with $\varphi(y) = m$, one has $m \geq 32{,}000$.*

Theorem 10 (External $k = 10$ Descent Forest). *Whenever $A(m) > 0$ and $10 \mid y$ for every $y > 0$ with $\varphi(y) = m$, the independently checked 8,243-root, 21,329-node descent forest proves $m \geq 10^{60}$, with rung-certificate soundness formalized in Lean 4.*

Conjecture 11 (Carmichael's Totient Conjecture). *For every positive integer m, $A(m) \neq 1$.*

Section 2 proves Lemma 5 and sets up the Case I/II split. Section 3 derives Corollary 4 from the forced divisors. Section 4 details the finite prime records, yield model, and $k = 10$ descent forest. Section 5 presents the machine-generated Lean 4 Reader's Trust Boundary, and Sections 6 and 7 summarize historical context, open questions, and data availability.

2. DEFINITIONS, NOTATION, AND SWITCHING LEMMA

For $n \geq 1$, put $\mathrm{rad}(n) = \prod_{p\mid n} p$ (with $\mathrm{rad}(1) = 1$), and write $\nu_p(n)$ for the exponent of p in n. A divisor d of $x = dw$ is *unitary* if $\gcd(d, w) = 1$, and $p^j \parallel x$ means $\nu_p(x) = j$. Two identities from the prime-power formula for φ recur throughout: if $e \mid w/\mathrm{rad}(w)$ then w/e and w share the same prime factors and $\varphi(w/e) = \varphi(w)/e$; if every prime of u divides z then $\varphi(uz) = u\varphi(z)$.

Proof of Lemma 5. If $P \nmid x$, replacing dw by $P(w/e)$ yields a distinct integer containing P with totient $(P-1)\varphi(w/e) = \varphi(d)\varphi(w) = \varphi(x)$. If $P \parallel x$, write $x = Pz$; every prime of $P - 1 = e\varphi(d)$ divides x and is smaller than P, hence divides z, so $(P-1)z \neq x$ also has totient $(P-1)\varphi(z) = \varphi(x)$. Both cases contradict $A(\varphi(x)) = 1$. □

If x were odd then $\varphi(2x) = \varphi(x)$; if $2 \parallel x$ then $\varphi(x/2) = \varphi(x)$. Hence $4 \mid x$. Applying Lemma 5 with $d = 1$ and $e = 2, 6, 42$ forces $3^2 \mid x$, $7^2 \mid x$, and $43^2 \mid x$. The next candidate $1+2\cdot3\cdot7\cdot43 = 1807 = 13\cdot139$ is composite. Splitting on $\nu_3(x)$ recovers Carmichael and Ford's two recurrences [3, 5, 7]:

Case	Condition	Initial b_i	Candidates P
I	$3^2 \parallel x$	36	$6k+1$, $12k+1$
II	$3^3 \mid x$	108	$6k+1$, $18k+1$

Here k is any squarefree product of primes at least 7 whose squares already divide x (including $k = 1$). For $6k+1$ take $d = 1, e = 6k$. In Case I, $d = 9$ is unitary with $\varphi(9) = 6$, so $e = 2k$ yields $12k+1$; in Case II the extra power of 3 permits $d = 1, e = 18k$ for $18k+1$. Starting at $k = 1$ forces $7, 13$ in Case I and $7, 19$ in Case II; the next steps give $79 = 6\cdot13+1$ and $127 = 18\cdot7+1$.

3. TRANSFER TO TOTIENT VALUE AND DENSITY

Let S_i be the set of distinct certified primes in Case i, so $d_i = b_i \prod_{p\in S_i} p^2 \mid x$. The finite record in Section 4 gives $d_i > 10^{D_i}$ with $D_{\mathrm{I}} = 100{,}000{,}160{,}994$ and $D_{\mathrm{II}} = 100{,}000{,}260{,}864$. Set $D = D_{\mathrm{I}}$, $u = 1{,}936{,}274$, and $v = 6{,}432{,}163$, so $10^u < 2^v$. For the Case I bit sum $B = 332{,}193{,}344{,}302$,

$$6Bu - (6D+5)v = 3{,}541 > 0,$$

giving $d_{\mathrm{I}} \geq 2^B > 10^{D+5/6} > 6\cdot10^D$ since $6^6 < 10^5$. Case II gives $d_{\mathrm{II}} > 10^{D+2}$ directly.

Bounding the Totient Value. Divisibility $d_i \mid x$ implies $\varphi(d_i) \mid \varphi(x)$. Let T_i be the smallest 1,000 primes in the certified base of Case i and set $t_i = \sum_{s\in T_i} 1/s$. Summing $\lceil 10^5/s \rceil$ over T_i gives 27,595 and 23,756, so $t_i < 1/3$. In each case the largest seed exceeds 10^8 and fewer than 10^6 further base primes remain, contributing less than $1/100$ to the reciprocal sum. Every additional prime has the form $P = ck+1$ with k a product of one to four distinct seeds, so

$$\sum_{P\ \text{add.}} \frac{1}{P} < \left(\frac{1}{6} + \frac{1}{12}\right) \sum_{j=1}^{4} t_i^j < \frac{1}{4}\frac{t_i}{1-t_i} < \frac{1}{8}.$$

The same bound holds with 18 in place of 12. Hence $\sum_{p\in S_i} 1/p < 1/3+1/100+1/8 < 1/2$, and using $\prod(1-u_j) \geq 1-\sum u_j$ yields

$$\frac{\varphi(d_i)}{d_i} = \frac{1}{3}\prod_{p\in S_i}\left(1-\frac{1}{p}\right) > \frac{1}{6}.$$

Combined with $d_i > 6\cdot 10^D$, this gives $m \geq \varphi(d_i) > 10^D$.

Counting Singleton Values Among All Totients. Write $a_i = 6\prod_{p\in S_i} p$, so $d_{\mathrm{I}} = a_{\mathrm{I}}^2$ and $d_{\mathrm{II}} = 3a_{\mathrm{II}}^2$. Since D is even, $d_{\mathrm{I}} > 6\cdot 10^D$, and $d_{\mathrm{II}}/3 > 10^{D+2}/3 > 4\cdot 10^D$, both a_i exceed $2\cdot 10^{50,000,080,497}$.

Let $V(X;a^2)$ count totients $n \leq X$ whose every preimage is divisible by a^2 ($a \geq 2$). By Ford's Lemmas 7.4–7.5 [5], $V(X;a^2) \leq V(X/a)$: for $X/a < n \leq X$ with preimage $y = a^t r$ ($a \nmid r$), mapping n to $\varphi(ar) = n/a^{t-1}$ injects $(X/a, X]$ into totients not counted by $V(X;a^2)$, because two distinct numbers in $(X/a, X]$ cannot differ by an integral power of a.

Set $A = \min(a_{\mathrm{I}}, a_{\mathrm{II}})$. Then $V_1(X) \leq V(X/a_{\mathrm{I}}) + V(X/a_{\mathrm{II}}) \leq 2V(X/A)$. If $V_1(X)/V(X) \geq c > 2/A$ held eventually, iterating $V(X) \leq (2/c)V(X/A)$ would force $V(X) = O(X^{\log(2/c)/\log A})$ with exponent strictly below 1, contradicting $V(X) \geq \pi(X+1)$ from the prime number theorem. Hence $\liminf_{X\to\infty} V_1(X)/V(X) \leq 2/A < 10^{-50,000,080,497}$, proving Corollary 4.

4. COMPUTATIONAL VERIFICATION AND FINITE RECORDS

From Prime Witnesses to Decimal Exponents. For each candidate $P = ck+1$, the construction supplies the full prime factorization of $P-1$. For each distinct prime $q \mid P-1$, a Lucas witness a_q satisfies

$$a_q^{P-1} \equiv 1 \pmod P, \qquad a_q^{(P-1)/q} \not\equiv 1 \pmod P,$$

certifying primality of P [1]. Because every prime factor of k is already forced in the same case, primality of P immediately forces $P^2 \mid x$.

Following Schlafly–Wagon and Ford [5, 9], we build a certified base, select its smallest 1,000 seeds, process an initial 1- to 3-factor batch, and then run 4-factor batches indexed by their smallest seed to the totals in Table 1. Partitioning primes into chunks with product G and summing $\lfloor \log_2(G^2) \rfloor$ with $\lfloor \log_2 b_i \rfloor$ yields $d_i \geq 2^B$, and $10^{1,936,274} < 2^{6,432,163}$ converts B to D_i.

Across both cases the record contains 1,965,316,556 entries. An independent checker replayed every derivation, Lucas witness, and chunk sum across both bases and 699 batch files, rejecting all 5 deliberately corrupted inputs; a separate regeneration reproduced both initial batches byte for byte. Because the disk holding the complete raw per-prime batch files was later retired, `recompute.py` replays the verbatim Section 1 example, freshly derives Lucas witnesses for all 2,000 seeds, and checks all aggregate bit-to-decimal transfers from the archived manifest (`recompute.out`); repeating the full per-prime replay requires regenerating those batch files.

Candidate Yield Model. Each prime P adds $2\log_{10} P$ to the decimal exponent. Under uniform residue classes mod $q \geq 5$, the local survival ratio is $q(q-2)/(q-1)^2$, giving a Hardy–Littlewood multiplier $4C_2$ and expected yield $8C_2/\log 10 \approx 2.293637$ digits per candidate. Because the actual 1,000 seeds are biased (for example, 52.3% of Case II seeds are 1 mod 7, which $6k+1$ forbids), computing exact residue counts for $5 \leq q \leq 541$ gives tight agreement:

Decimal exponent per candidate	Case I	Case II
Predicted from seed residues	2.279324	2.209519
Measured from chunk bit sums	2.279365	2.208427

The $k = 10$ Divisible-Preimage Forest. For Theorems 9 and 10, multiplicity is unrestricted: fix m in the image of φ such that $10 \mid y$ for every $y > 0$ with $\varphi(y) = m$, and pick any preimage x. Then $2 \parallel x$ is impossible via $x/2$, and $5 \parallel x$ ($x = 5w$, $4 \mid w$) is impossible via $\varphi(4w) = 4\varphi(w) = m$ with $5 \nmid 4w$. Thus $100 \mid x$ and $40 = \varphi(100) \mid m$.

A switch now need only produce a preimage missing 2 or 5: for unitary $d \mid x$ with $2 \mid d$, $e \mid w/\operatorname{rad}(w)$, $t \geq 0$ with $5^t \mid e\varphi(d)$ (and $t = 0$ or $5 \nmid d$), any set of distinct odd primes P_i with $\prod_i (P_i - 1) = e\varphi(d)/5^t$ forces at least one $P_i \mid x$ (else $5^t(w/e)\prod_i P_i$ is an odd preimage of m). A symmetric rule with $5 \mid d$ and 2^t avoids 5. For instance, at $(a,b) = (\nu_2(x), \nu_5(x)) = (10,6)$, taking $d = 2^{10}, e = 1, t = 0$ gives $\varphi(d) = 512 = (3-1)(257-1)$, forcing $3 \mid x$ or $257 \mid x$.

Assuming $m < 10^{60}$ restricts (a,b) to the 8,243 pairs with $a,b \geq 2$ and $\varphi(2^a 5^b) = 2^{a+1}5^{b-1} < 10^{60}$. Closing each root by prime disjunctions and valuation splits ($p^v \parallel x$ vs. $p^{v+1} \mid x$) yields a finite forest (Figure 1) of 21,329 nodes, 59,755 forced-prime entries, 893 product splits, and 14,786 closed leaves. Running `python3 verify_forest.py` verifies all 8,243 roots, checks path reachability from every root, replays every arithmetic and leaf inequality, and rejects empty-tree and detached-node forgeries (`verify_forest.out`). A separate check without product splitting establishes $m > 10^{12}$.

Table 1. Exact accounting for the forced divisor in Theorem 3. The first three bit rows sum to the integer bit lower bound; the bottom row converts that bit sum to decimal digits.

Quantity	Case I ($3^2 \parallel x$)	Case II ($3^3 \mid x$)
Batches, including initial batch	171	179
Candidates tested	43,866,971,434	45,276,159,886
Primes in base	155,792	117,842
Additional primes	1,120,013,445	845,029,477
Distinct primes within case	1,120,169,237	845,147,319
$\lfloor \log_2 b_i \rfloor$	5	6
Bits from base primes squared	37,637,307	37,045,809
Bits from additional primes squared	332,155,706,990	332,156,630,246
B, integer bit-sum lower bound	332,193,344,302	332,193,676,061
D_i, certified decimal exponent	100,000,160,994	100,000,260,864

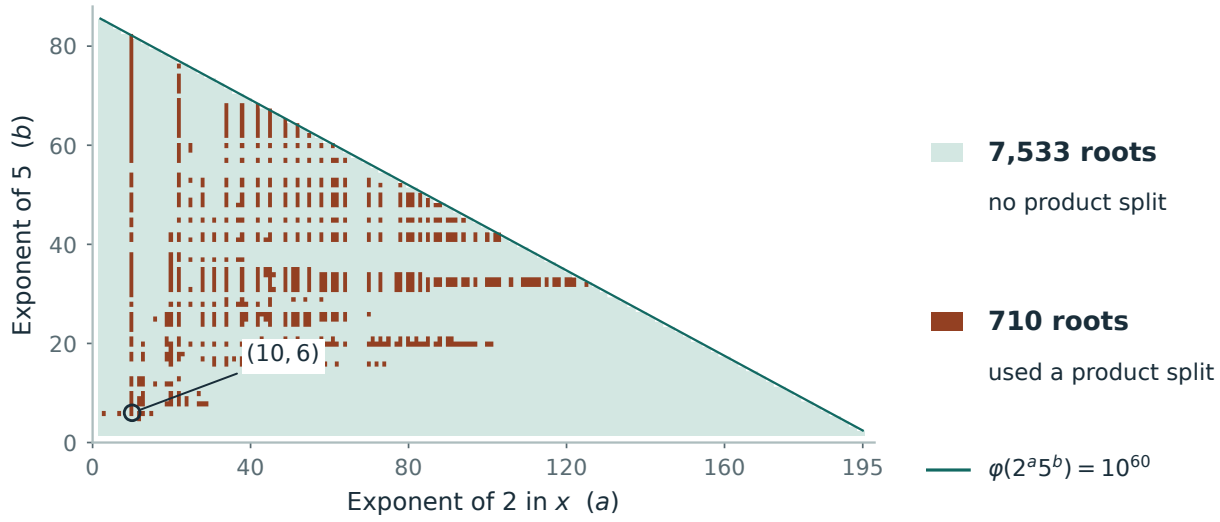


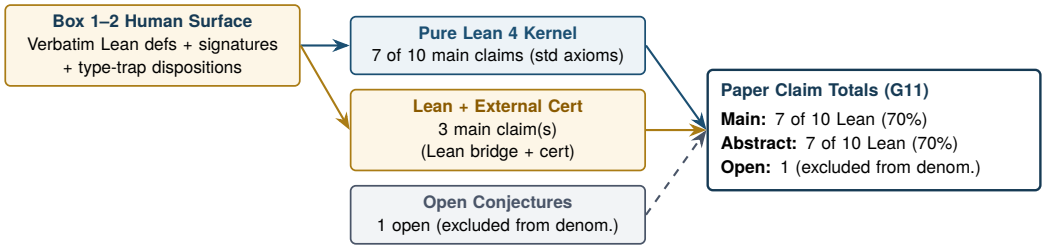
Figure 1. Valuation-pair descent forest for Theorem 10 across all 8,243 admissible roots below the sixty-digit threshold. Darker cells mark the 710 roots using at least one multi-prime product split; light cells (7,533 roots, of which 1,366 branch on a prime valuation) close without product splits. The ring marks the worked root $(10, 6)$.

5. FORMAL VERIFICATION & READER'S TRUST BOUNDARY

What You Must Trust to Accept This Paper:

1. **Lean 4 Kernel & Toolchain:** leanprover/lean4:v4.35.0; committed axiom receipt anchored to git commit 6e96187d2241 (independent gate-box kernel replay pending); standard axioms only (propext, Classical.choice, Quot.sound).
2. **Core Definitions & Fidelity (Boxes 1–2):** Verify that the verbatim Lean definitions in Box 1 match Section 2 and that each Lean signature in Box 2 matches its stated theorem.
3. **External Certificates & Written Proofs:** 2 SHA-256-pinned external artifact(s) plus any claims marked Written or Open in Table 3 below.

Reader's Trust-Flow Architecture (Distinct Evidence Paths → Claim Totals)



Anti-Shortcut Audit Strip — Source-Level Approx (Gate Replay Pending)

WARN: Committed receipt (gate replay pending) | PASS: sorry/admit: 0 (per committed receipt) | PASS: native_decide: 0 (per committed receipt) | PASS: Custom axioms & meta: 0 (per committed receipt) | PASS: Coverage: 173/173 public theorems (source-level approx) | WARN: Fidelity: 10/10 exact (draft cross-family) | WARN: Witnesses: pending gate replay | WARN: Human referee: pending

Box 1 — Verbatim Lean 4 Custom Definitions (Human Audit Surface)

CarmichaelTotient.CarmichaelBoundCertificate (CarmichaelTotient/CaseRules.lean:787-789) — Theorem 3 (paired Case I/II certificate)

```
structure CarmichaelBoundCertificate where
  case1 : Case1Certificate
  case2 : Case2Certificate
```

CarmichaelTotient.ForcedCase1 (CarmichaelTotient/CaseRules.lean:410-418) — Section 2 (Case I forced prime closure)

```
inductive ForcedCase1 : ℕ → Prop
  | step6 {K : Finset ℕ} :
      (∀ q ∈ K, ForcedCase1 q) →
      Nat.Prime (6 * (∏ q ∈ K, q) + 1) →
      ForcedCase1 (6 * (∏ q ∈ K, q) + 1)
  | step12 {K : Finset ℕ} :
      (∀ q ∈ K, ForcedCase1 q) →
      Nat.Prime (12 * (∏ q ∈ K, q) + 1) →
      ForcedCase1 (12 * (∏ q ∈ K, q) + 1)
```

CarmichaelTotient.ForcedCase2 (CarmichaelTotient/CaseRules.lean:422-430) — Section 2 (Case II forced prime closure)

```
inductive ForcedCase2 : ℕ → Prop
  | step6 {K : Finset ℕ} :
      (∀ q ∈ K, ForcedCase2 q) →
      Nat.Prime (6 * (∏ q ∈ K, q) + 1) →
      ForcedCase2 (6 * (∏ q ∈ K, q) + 1)
  | step18 {K : Finset ℕ} :
      (∀ q ∈ K, ForcedCase2 q) →
      Nat.Prime (18 * (∏ q ∈ K, q) + 1) →
      ForcedCase2 (18 * (∏ q ∈ K, q) + 1)
```

CarmichaelTotient.IsCarmichaelCounterexample (CarmichaelTotient/Lemma72.lean:30-31) — Section 1 (singleton preimage A(phi(x)) = 1)

```
def IsCarmichaelCounterexample (x : ℕ) : Prop :=
  0 < x ∧ ∀ y : ℕ, Nat.totient y = Nat.totient x → y = x
```

CarmichaelTotient.IsDivisibleTotientWitness (CarmichaelTotient/ValuationTree.lean:352-354) — Propositions 8-10 (divisible-preimage witness)

```
def IsDivisibleTotientWitness (k m : ℕ) : Prop :=
  (∃ x : ℕ, 0 < x ∧ Nat.totient x = m) ∧
    ∀ y : ℕ, 0 < y → Nat.totient y = m → k ∣ y
```

CarmichaelTotient.IsMinimalCounterexample (CarmichaelTotient/StructuralAndPomerance.lean:317-318) — Proposition 6 (minimal singleton preimage)

```
def IsMinimalCounterexample (x : ℕ) : Prop :=
  IsCarmichaelCounterexample x ∧ ∀ y : ℕ, y < x → ¬ IsCarmichaelCounterexample y
```

CarmichaelTotient.IsTotientValue (CarmichaelTotient/ValueAndDensityTransfer.lean:170-171) — Section 1 (totient image predicate)

```
def IsTotientValue (n : ℕ) : Prop :=
  ∃ y : ℕ, 0 < y ∧ Nat.totient y = n
```

CarmichaelTotient.K10RungCertificate (CarmichaelTotient/ValuationTree.lean:1382-1384) — Theorem 10 (k=10 rung descent certificate)

```
structure K10RungCertificate (K : ℕ) : Prop where
  trees : ∀ a b : ℕ, isK10Root K a b →
    K10DescentTree [⟨2, a, true⟩, ⟨5, b, true⟩] K
```

CarmichaelTotient.V (CarmichaelTotient/ValueAndDensityTransfer.lean:174-176) — Section 1 (totient value count V(X))

```
noncomputable def V (X : ℕ) : ℕ := by
  classical
  exact ((Finset.Icc 1 X).filter (fun n => IsTotientValue n)).card
```

CarmichaelTotient.V1 (CarmichaelTotient/ValueAndDensityTransfer.lean:179-182) — Section 1 (singleton totient count V_1(X))

```
noncomputable def V1 (X : ℕ) : ℕ := by
  classical
  exact ((Finset.Icc 1 X).filter
    (fun n => ∃ x : ℕ, IsCarmichaelCounterexample x ∧ Nat.totient x = n)).card
```

CarmichaelTotient.rad (CarmichaelTotient/Lemma72.lean:35-36) — Section 2 (radical rad(n))

```
def rad (n : ℕ) : ℕ :=
  ∏ p ∈ n.primeFactors, p
```

Selected Concrete Sanity Examples (checked in Lean; examples only, not global identity proofs): CarmichaelTotient.rad_one (rad(1) = 1); CarmichaelTotient.rad_six (rad(6) = 6); CarmichaelTotient.carmichael1922Certificate_digits (Carmichael 1922 certificate yields 37 digits).

Supporting Semantic Closure Index (non-Box-1 helpers): CarmichaelTotient.CarmichaelBoundCertificate.digits (CarmichaelTotient/CaseRules.lean), CarmichaelTotient.Case1Certificate (CarmichaelTotient/CaseRules.lean), CarmichaelTotient.Case1Certificate.digits (CarmichaelTotient/CaseRules.lean), CarmichaelTotient.Case2Certificate (CarmichaelTotient/CaseRules.lean), CarmichaelTotient.Case2Certificate.digits (CarmichaelTotient/CaseRules.lean), CarmichaelTotient.ExactPowDvd (CarmichaelTotient/ValuationTree.lean), CarmichaelTotient.K10DescentTree (CarmichaelTotient/ValuationTree.lean), CarmichaelTotient.ValConstraint (CarmichaelTotient/ValuationTree.lean), CarmichaelTotient.ValConstraint.holds (CarmichaelTotient/ValuationTree.lean), CarmichaelTotient.isK10Root (CarmichaelTotient/ValuationTree.lean).

Box 2 — Verbatim Main Theorem Signatures, Context Binders & Applicability

CarmichaelTotient.carmichael_counterexample_gt_10_pow_10117 (CarmichaelTotient/TreeAndKleeBounds.lean:912-919) src:35c3212a55b45e

```
theorem carmichael_counterexample_gt_10_pow_10117 {x : ℕ}
    (hx : IsCarmichaelCounterexample x) :
    10 ^ 10117 < x
```

Assumptions: Hypothetical Carmichael counterexample x (IsCarmichaelCounterexample x); empty if Conjecture 11 holds. Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: unknown_or_empty).

CarmichaelTotient.carmichael_liminf_density_lt_10_pow_neg_5057 (CarmichaelTotient/ValueAndDensityTransfer.lean:841-890) src:ac1eff0053e7b3

```
theorem carmichael_liminf_density_lt_10_pow_neg_5057 :
    Filter.liminf (fun X : ℕ => ((V1 X : ℝ) / (V X : ℝ))) Filter.atTop <
      1 / (10 : ℝ) ^ 5057
```

Assumptions: Unconditional closed statement over N and R (zero outer premises). Applicability: Unconditional (closed statement; no outer premise witness needed).

CarmichaelTotient.carmichael_bound_strict_from_certificate (CarmichaelTotient/CaseRules.lean:834-865) src:a5b7ff44f8f46c

```
theorem carmichael_bound_strict_from_certificate (cert : CarmichaelBoundCertificate)
    (hD : 0 < cert.digits) {x : ℕ} (hx : IsCarmichaelCounterexample x) :
    10 ^ cert.digits < x
```

Assumptions: Valid CarmichaelBoundCertificate (digits > 0) and hypothetical counterexample x; 100B instance checked externally via recompute.py. Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: pending_replay).

CarmichaelTotient.corollary_value_and_density_transfer (CarmichaelTotient/ValueAndDensityTransfer.lean:709-787) src:c7fdb93880be1c

```
theorem corollary_value_and_density_transfer
    {H : ℕ} {S₁ S₂ : Finset ℕ}
    (hS₁ : ∀ p ∈ S₁, ForcedCase1 p)
    (hS₂ : ∀ p ∈ S₂, ForcedCase2 p)
    (hbig₁ : 6 * 10 ^ (2 * H) < 36 * ∏ p ∈ S₁, p ^ 2)
    (hbig₂ : 12 * 10 ^ (2 * H) < 108 * ∏ p ∈ S₂, p ^ 2)
    (hrec₁ : ∑ p ∈ S₁, (1 : ℝ) / (p : ℝ) < 1 / 2)
    (hrec₂ : ∑ p ∈ S₂, (1 : ℝ) / (p : ℝ) < 1 / 2) :
    (∀ x : ℕ, IsCarmichaelCounterexample x → 10 ^ (2 * H) < Nat.totient x) ∧
    Filter.liminf (fun X : ℕ => ((V1 X : ℝ) / (V X : ℝ))) Filter.atTop <
      1 / (10 : ℝ) ^ H
```

Assumptions: Paired forced-prime sets S1, S2 with d1 > 6*10^(2H), d2 > 12*10^(2H), and sum 1/p < 1/2 (100B instance checked via recompute.py). Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: pending_replay).

CarmichaelTotient.ford_lemma_7_2 (CarmichaelTotient/Lemma72.lean:272-280) src:368693b0f37748

```
theorem ford_lemma_7_2 {x d e : ℕ}
    (hx : IsCarmichaelCounterexample x)
    (hd : d ∣ x)
    (hcop : Nat.Coprime d (x / d))
    (hrad : rad (Nat.totient d) ∣ x)
    (he : e ∣ (x / d) / rad (x / d))
    (hP : Nat.Prime (1 + e * Nat.totient d)) :
    (1 + e * Nat.totient d) ^ 2 ∣ x
```

Assumptions: Hypothetical counterexample x, unitary divisor d | x, rad(phi(d)) | x, e | (x/d)/rad(x/d), and prime 1 + e*phi(d). Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: unknown_or_empty).

CarmichaelTotient.minimal_counterexample_structure (CarmichaelTotient/StructuralAndPomerance.lean:528-533) src:1710b523946800

```
theorem minimal_counterexample_structure {x₀ : ℕ} (hx₀ : IsMinimalCounterexample x₀) :
    4 ∣ x₀ ∧ ¬ (8 ∣ x₀) ∧ ¬ (5 ∣ x₀) ∧ ¬ (17 ∣ x₀) ∧ ¬ (257 ∣ x₀) ∧ ¬ (65537 ∣ x₀)
```

Assumptions: Hypothetical minimal Carmichael counterexample x0 (IsMinimalCounterexample x0). Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: unknown_or_empty).

CarmichaelTotient.pomerance_sufficient_condition (CarmichaelTotient/StructuralAndPomerance.lean:151-272) src:3d2bda0c8297a2

```
theorem pomerance_sufficient_condition {x : ℕ} (hx : 0 < x)
    (hpom : ∀ p : ℕ, Nat.Prime p → (p - 1) ∣ Nat.totient x → p ^ 2 ∣ x) :
    IsCarmichaelCounterexample x
```

Assumptions: Positive integer x with p^2 | x for every prime p satisfying (p - 1) | phi(x). Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: unknown_or_empty).

CarmichaelTotient.no_k6_witness_implies_carmichael (CarmichaelTotient/ValuationTree.lean:359-372) src:23733e02b1fd7f

```
theorem no_k6_witness_implies_carmichael
    (h6 : ¬ ∃ m : ℕ, IsDivisibleTotientWitness 6 m) :
    ∀ x : ℕ, ¬ IsCarmichaelCounterexample x
```

Assumptions: Unconditional implication: non-existence of k=6 divisible-preimage witness implies Carmichael conjecture. Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: unknown_or_empty).

CarmichaelTotient.k10_witness_ge_32000 (CarmichaelTotient/ValuationTree.lean:992-1129) src:d1f9601a8794b7

```
theorem k10_witness_ge_32000 {m : ℕ} (hm : IsDivisibleTotientWitness 10 m) :
    32000 ≤ m
```

Assumptions: Hypothetical k=10 divisible-preimage witness m (IsDivisibleTotientWitness 10 m); predicate inhabited by k11_witness_110. Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: pending_replay).

CarmichaelTotient.k10_bound_from_rung_certificate (CarmichaelTotient/ValuationTree.lean:1389-1417) src:f563164e8d4414

```
theorem k10_bound_from_rung_certificate {K : ℕ} (cert : K10RungCertificate K) {m : ℕ}
    (hm : IsDivisibleTotientWitness 10 m) :
    10 ^ K ≤ m
```

Assumptions: Valid K10RungCertificate K (inhabited at K=4 by k10RungCertificate_4; K=60 forest verified via verify_forest.py) and k=10 witness m. Applicability: Hypothetical constraint — vacuously true if no such object exists (carrier status: pending_replay).

Table 2. Lean 4 Type-Trap Audit Checklist — Source-Level Approximation (Gate Replay Pending).[a]

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
`Case1Certificate`	`real_log_rpow`	`^ (bc c)`	intended_convention	Integer Nat.pow with base 2 > 0; no real rpow semantics involved.
`digits`	`nat_div`	`/ cert.q10`	intended_convention	Case1Certificate stores hq10 : 0 < q10; floor(p10 * B / q10) is exact lower bound.
`Case2Certificate`	`real_log_rpow`	`^ (bc c)`	intended_convention	Integer Nat.pow with base 2 > 0; no real rpow semantics.

Declaration	Hazard	Operand	Category	Audited Disposition & Guard
digits	nat_div	/ cert.q10	intended_convention	Case2Certificate stores hq10 : 0 < q10; floor(p10 * B / q10) is exact lower bound.
K10DescentTree	nat_sub	- 1	intended_convention	Constructors enforce 1 <= exp p and Nat. Prime P (P >= 2); no Nat underflow.
K10DescentTree	nat_div	/ d	intended_convention	d = prod q^{exp q} divides x > 0 by state st; x / d is exact.
K10DescentTree	nat_div	/ rad	intended_convention	x > 0 and d ∣ x imply rad(x/d) ∣ (x/d) > 0; division is exact.
holds	real_log_rpow	^ c.k	intended_convention	Integer Nat.pow in divisibility predicate; no real rpow semantics.
carmichael_bound_strict_from_certificate	real_log_rpow	^ cert.digits	intended_convention	Integer Nat.pow with base 10 > 0; no real rpow semantics.
carmichael_liminf_density_lt_10_pow_neg_5057	real_div	(V X : ℝ)	proved_guard (V_pos)	V_pos proves 0 < V X for X >= 1 (phi(1)=1), preventing zero division.
corollary_value_and_density_transfer	real_div	(p : ℝ)	intended_convention	Forced primes satisfy p >= 5, so (p : R) > 0 in every 1 / p term.
corollary_value_and_density_transfer	real_log_rpow	^ (2 * H)	intended_convention	Integer Nat.pow with base 10 > 0; no real rpow semantics.
corollary_value_and_density_transfer	real_div	(V X : ℝ)	proved_guard (V_pos)	V_pos proves 0 < V X for X >= 1, preventing zero division.
corollary_value_and_density_transfer	real_log_rpow	^ H	intended_convention	Base 10 > 0 with natural exponent H implies (10 : R) ^ H > 0.
ford_lemma_7_2	nat_div	/ d	intended_convention	x > 0 and hd : d ∣ x guarantee d > 0 and exact division d * (x / d) = x.
ford_lemma_7_2	nat_div	/ rad	intended_convention	x / d > 0 implies rad(x/d) > 0 and rad(x/d) ∣ (x/d); exact division.
isK10Root	nat_sub	- 1	intended_convention	isK10Root requires 2 <= b, so b - 1 >= 1 never underflows Nat.
pomerance_sufficient_condition	nat_sub	- 1	intended_convention	Nat.Prime p guarantees p >= 2, so p - 1 >= 1 never underflows Nat.

[a] Covers the specified Lean 4 hazard set (Nat subtraction/division/mod, totalized real/rational division, Real.log/rpow, Finset.range, Classical.choose, local instances); no finite checklist certifies all semantics.

Table 3. Per-Claim Reader's Trust Boundary Ledger (G11 Denominators Derived Mechanically).[a]

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Thm. 1: Every singleton preimage satisfies x > 10^10,117	Lean Kernel	CarmichaelTotient.carmichael_counterexample_gt_10_pow_10117	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel + Box 1 def IsCarmichaelCounterexample only (161 in-kernel Lucas ladders).
Cor. 2: Unconditional density liminf V_1(X)/V(X) < 10^-5,057	Lean Kernel	CarmichaelTotient.carmichael_liminf_density_lt_10_pow_neg_5057	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel + Box 1 defs V, V1 + Mathlib Chebyshev.pi_ge only.
Thm. 3: Certified bound x > 10^100,000,160,994	Lean + Cert	CarmichaelTotient.carmichael_bound_strict_from_certificate + recompute.py (a7fd5bf5b12d) [replay pending]	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean soundness bridge + independently rechecked 1.965B-prime external Lucas certificate (recompute.py).
Cor. 4: m > 10^100,000,160,994 and liminf V_1(X)/V(X) < 10^-50,000,080,497	Lean + Cert	CarmichaelTotient.corollary_value_and_density_transfer + recompute.py (a7fd5bf5b12d) [replay pending]	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean transfer theorem (Ford Lemmas 7.4-7.5 + Weierstrass bound) + certified prime bases (recompute.py).
Lem. 5: Ford Lemma 7.2 forcing rule (P^2 ∣ x)	Lean Kernel	CarmichaelTotient.ford_lemma_7_2	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel + Box 1 defs IsCarmichaelCounterexample and rad only.
Prop. 6: Klee structure (4 ∥ x_0; 5, 17, 257, 65537 excluded)	Lean Kernel	CarmichaelTotient.minimal_counterexample_structure	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel + Box 1 def IsMinimalCounterexample only.
Prop. 7: Pomerance sufficient condition for A(phi(x)) = 1	Lean Kernel	CarmichaelTotient.pomerance_sufficient_condition	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel + Box 1 def IsCarmichaelCounterexample only.
Prop. 8: No k=6 witness implies Carmichael conjecture	Lean Kernel	CarmichaelTotient.no_k6_witness_implies_carmichael	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel + Box 1 defs IsDivisibleTotientWitness and IsCarmichaelCounterexample only.

Paper Claim	Proof Tier	Lean Decl / Artifact	Actual Axioms & Fidelity	Assumptions & What You Must Trust
Thm. 9: Every k=10 witness satisfies m >= 32,000	Lean Kernel	`CarmichaelTotient.k10_witness_ge_32000`	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean 4 kernel + Box 1 def IsDivisibleTotientWitness only.
Thm. 10: External 8,243-root forest gives m >= 10^60 (k=10)	Lean + Cert	`CarmichaelTotient.k10_bound_from_rung_certificate` + verify_forest.py (3b13e5a9acdb) [replay pending]	Axioms: Classical.choice, Quot.sound, propext; fidelity: exact	Lean K10RungCertificate soundness bridge + external 8,243-root forest checked by verify_forest.py.
Conj. 11: Carmichael conjecture (A(m) != 1 for all m >= 1)	Unsettled	Open conjecture	Lean: none; fidelity: n/a	Explicitly open conjecture (excluded from G11 denominator).

[a] Allowed pure-kernel axioms: `propext`, `Classical.choice`, `Quot.sound`. Actual axioms printed above are read directly from verified outputs. Provenance nonce: e7c38ed7b774 | Lean mode: committed_receipt | Closure SHA-256: 4e1c1aa29d72 | Git anchor: 6e96187d2241 | Axioms SHA-256: b00112f1039c | Fidelity SHA-256: 846d855782d1.

6. HISTORICAL SCOPE AND OPEN QUESTIONS

Carmichael withdrew his initial proof and introduced forced divisors in 1922 [2, 3]. Klee refined the valuation structure and reached $x > 10^{400}$ [7], Masai and Valette reached $10^{10,000}$, Schlafly and Wagon certified primes to $10^{10,000,000}$ [9], and Ford proved $x > 10^{10,003,700,000}$ and $m \geq 10^{10^{10}}$ using the same 1,000-seed batch architecture [5]. Erdős showed that any single counterexample would imply infinitely many via fiber-preserving prime shifts (formalized in `StructuralAndPomerance.lean`) [4].

Conjecture 11 remains open: adjoining 2 and 3 to either forced set yields a set closed under prime factors of $p - 1$ omitting 5, which Ford proved has convergent reciprocal sum [6]. Likewise, whether any totient m has all preimages divisible by 10 remains open above 10^{60}.

7. DATA AVAILABILITY, REPRODUCIBILITY, AND USE OF AI

All manuscripts, scripts (`recompute.py`, `verify_forest.py`), compressed descent-forest data, trust-boundary manifests, and Lean 4 sources live under `papers/carmichael-totient/draft4/` and `lean/CarmichaelTotient/`. AI research agents built and directed by Tyler B. Hughes assisted with searching, software, formalization, and drafting. No human mathematician has yet refereed the proof.

Drafted: November 2026 **Updated:** November 23, 2026

REFERENCES

(1) Brillhart, J.; Lehmer, D. H.; Selfridge, J. L. New primality criteria and factorizations of $2^m \pm 1$. *Math. Comp.* **1975**, *29* (130), 620–647.

(2) Carmichael, R. D. On Euler's ϕ-function. *Bull. Amer. Math. Soc.* **1907**, *13* (5), 241–243.

(3) Carmichael, R. D. Note on Euler's φ-function. *Bull. Amer. Math. Soc.* **1922**, *28* (3), 109–110.

(4) Erdős, P. Some remarks on Euler's φ function. *Acta Arith.* **1958**, *4*, 10–19.

(5) Ford, K. The distribution of totients. *Ramanujan J.* **1998**, *2* (1–2), 67–151; updated arXiv:1104.3264v2 (2013), 43 pp.

(6) Ford, K. Sieving by very thin sets of primes, and Pratt trees with missing primes. *Int. Math. Res. Not. IMRN* **2014**, *2014* (11), 2955–2971; arXiv:1212.3498.

(7) Klee, V. L., Jr. On a conjecture of Carmichael. *Bull. Amer. Math. Soc.* **1947**, *53* (12), 1183–1186.

(8) Pomerance, C. On Carmichael's conjecture. *Proc. Amer. Math. Soc.* **1974**, *43* (2), 297–298.

(9) Schlafly, A.; Wagon, S. Carmichael's conjecture on the Euler function is valid below $10^{10,000,000}$. *Math. Comp.* **1994**, *63* (207), 415–419.