Arbitrarily long avoidance of 3 in Euclid–Mullin sequences

P. Graham
https://orcid.org/0009-0007-8493-1505

October 26, 2026

Abstract

For a prime seed p_1, define p_{k+1} to be the least prime divisor of $1+p_1 \cdots p_k$. For every positive integer n, there is a prime seed for which none of the first n terms is 3. This answers the first of Donkin's two questions in Cameron's problem list. The proof chooses a finite increasing sequence of auxiliary primes and forces the sequence to begin with the seed, 2, and those primes. Booker and Irvine used the Chinese remainder theorem to prescribe prime-divisor paths, including constructions from prime starting values. Here congruences also exclude every smaller prime factor, making the prescribed divisors the least factors. Dirichlet's theorem then supplies a prime seed. The argument, including the congruence construction and its translation into the sequence, is formalized in Lean. The separate question of a single prime seed whose sequence avoids 3 forever remains open.

1 Introduction

The Euclid–Mullin rule extends a finite list of primes by the least prime divisor of one plus their product. Starting at 5, for example, the sequence begins $5, 2, 11, 3$. Mullin's original least-factor sequence starts at 2, so its second term is 3 [1]. Write $\mathrm{lpf}(a)$ for the least prime divisor of an integer $a > 1$. Throughout this paper a prime seed p_1 determines the sequence

$$p_{k+1} = \mathrm{lpf}(1 + p_1 \cdots p_k) \qquad (k \geq 1). \tag{1}$$

Theorem 1. *For every positive integer n, there is a prime p_1 for which the sequence (1) satisfies $p_j \neq 3$ for $1 \leq j \leq n$.*

This is the first question of Problem 19 in Cameron's archived list, attributed there to Steve Donkin [2, Problem 19]. The seed is included among the first n terms and may depend on n. Cameron's later Problem 109 and blog discussion report John Bray's computations of the first position at which 3 appears for prime seeds up to two million [2, Problem 109], [3]. The least seeds whose first 3 appears at each position from 1 to 15 were tabulated by Labos Elemer and extended by David Wasserman [4]. The second question in Problem 19 asks for one prime seed avoiding 3 forever; it remains open.

The closest method is Booker and Irvine's use of the Chinese remainder theorem to prescribe successive prime divisors from a chosen starting value. Their Lemma 2.2 gives an arithmetic progression, coprime to its modulus, of starting values for which the specified primes divide the numbers formed by adding one to the current product, allowing any prime factor, not necessarily the least, at each step. Remark 2.16 obtains prime starting values for loops (two distinct paths

between the same two nodes) in their graph [5]. Here we also exclude every divisor smaller than the prescribed prime. Counting supplies a residue for each sufficiently large prime modulus; the essential choice is to make all auxiliary primes congruent to 1 modulo the small primes, where counting alone does not suffice.

Paris's related Lean formalization proves that a fixed sequence of prime factors, each at most y, is preserved when the starting values agree modulo an integer divisible by every prime at most y [6, §4.3, item (i)].

Other Euclid–Mullin variants use different selection rules. For the rule permitting prime factors of sums of complementary subproducts, Booker's Theorem 1 shows that any finite set of primes can be extended to a sequence containing every prime [7]. Booker proved that the largest-factor sequence starting at 2 omits infinitely many primes [8, Theorem 1]; Pollack and Treviño gave an elementary proof [9, §1, Theorem (Booker)].

The proof chooses increasing auxiliary primes with the small-prime congruences just described. It then selects compatible seed residues at each prime, combines them by the Chinese remainder theorem, and obtains a prime seed by Dirichlet's theorem. Section 2 gives the construction; Section 3 identifies the formal theorem and its accompanying proof.

This work has not yet been reviewed by experts. The proofs, the Lean formalization and the text were produced by AI models (from OpenAI and Anthropic), working under the author's direction.

2 Proof of the main theorem

We prescribe an initial segment $p, 2, q_1, \dots, q_m$ and find a prime seed p by congruences. Fix a positive integer m. Let

$$m + 2 < q_1 < \cdots < q_m$$

be primes satisfying $q_i \equiv 1 \pmod{\ell}$ for every prime $\ell \le m + 2$. Such primes exist by Dirichlet's theorem in the progression 1 modulo the product of those small primes. Define

$$C_i = 2 \prod_{j<i} q_j \qquad (1 \le i \le m),$$

where an empty product is 1. If a seed p is followed by $2, q_1, \dots$, the integer whose least factor must be q_i is $pC_i + 1$.

Lemma 2. *For each prime ℓ, one can choose a nonzero residue r_ℓ modulo ℓ so that $r_2 = 1$ and, for every $1 \le i \le m$,*

$$\begin{aligned} C_i r_\ell + 1 &\equiv 0 \pmod{\ell} && \text{if } \ell = q_i, \\ C_i r_\ell + 1 &\not\equiv 0 \pmod{\ell} && \text{if } \ell < q_i. \end{aligned}$$

Proof. There are three cases. If $\ell \le m+2$, then $C_i \equiv 2 \pmod{\ell}$ for every i. For $\ell = 2$, take $r_2 = 1$; every $C_i r_2 + 1$ is odd. For odd $\ell \le m + 2$, take $r_\ell = -1$; then $C_i r_\ell + 1 \equiv -1 \pmod{\ell}$. None of the q_i equals such an ℓ, so these choices give all the required conditions.

Suppose next that $\ell = q_k$ for some k. The number C_k is invertible modulo ℓ, since its prime factors are 2 and the earlier, smaller q_j. Take $r_\ell = -C_k^{-1}$. This residue is nonzero and makes $C_k r_\ell + 1$ vanish. For $i > k$, the product C_i contains q_k, so $C_i r_\ell + 1 \equiv 1 \pmod{\ell}$. For $i < k$, the inequality $\ell < q_i$ does not hold, and no exclusion is required.

Finally, suppose that $\ell > m+2$ and that ℓ is not one of the q_i. For each i, the equation $C_i r + 1 = 0$ in the field $\mathbb{Z}/\ell\mathbb{Z}$ excludes at most one residue. Together with $r = 0$, at most $m+1$ residues are forbidden. Since $m+1 < \ell$, a permitted residue exists. It satisfies the stronger condition $C_i r + 1 \neq 0$ for every i. □

Lemma 3. *For primes $q_1, \ldots, q_m$ with $m+2 < q_1 < \cdots < q_m$ and $q_i \equiv 1 \pmod{\ell}$ for every prime $\ell \leq m+2$, there is a prime $p > 3$ whose sequence (1) begins $p, 2, q_1, \ldots, q_m$.*

Proof. Set $B = q_m$ and let M be the product of all primes at most B. For each prime $\ell \leq B$, choose the residue supplied by Lemma 2. The Chinese remainder theorem gives a residue a modulo M satisfying $a \equiv r_\ell \pmod{\ell}$ for every such ℓ. Every r_ℓ is nonzero, so $\gcd(a, M) = 1$. Dirichlet's theorem therefore gives a prime $p > 3$ with $p \equiv a \pmod{M}$.

The first term after p is 2, because p is odd. For each i, the congruences give $q_i \mid pC_i + 1$ and exclude every prime $\ell < q_i$ from dividing $pC_i + 1$. All these primes were included among the moduli, since $q_i \leq B$. Thus

$$\operatorname{lpf}(pC_i + 1) = q_i.$$

Induction on i now proves the claimed prefix: after $p, 2, q_1, \ldots, q_{i-1}$, the product of the terms already obtained is exactly pC_i. □

Proof of Theorem 1. Given $n \geq 1$, choose the auxiliary primes with $m = n$ and apply Lemma 3. The resulting prefix is $p, 2, q_1, \ldots, q_n$. Here $p > 3$, and each $q_i > n + 2 \geq 3$. Consequently all these terms avoid 3, in particular the first n. □

3 Formal verification

The argument is formalized in Lean 4 [10] using mathlib [11]. The accompanying Lean project is supplied as supplementary material, with checking instructions in `lean/README.md`.[1] The formal theorem `Donkin.arbitraryDelay` states

$$\begin{aligned} &\forall n \in \mathbb{N}, \quad n > 0 \implies \\ &\quad \exists p \in \mathbb{N}, \quad p \text{ is prime } \wedge \left(\forall j \in \mathbb{N},\ j < n \implies \operatorname{term}(p, j) \neq 3\right). \end{aligned}$$

The formal least-factor function is `Nat.minFac`. The definition has $\operatorname{term}(p, 0) = p$ and

$$\operatorname{term}(p, k+1) = \operatorname{lpf}(P(p,k) + 1), \qquad P(p, 0) = p, \quad P(p, k+1) = P(p, k) \operatorname{lpf}(P(p, k) + 1).$$

Thus $P(p, k)$ is the product of terms with indices $0, \ldots, k$; the term with index j is p_{j+1} in (1). For a prime seed, $P(p, k) + 1 \geq 3$, within the domain of lpf.

Lemmas 2 and 3, including the induction establishing the prefix, are checked in Lean. Dirichlet's theorem is supplied by mathlib's formal proof. Every mathematical step is formalized; the proofs use only Lean's standard classical axioms, specified in the supplementary checking notes.

[1] https://github.com/6/clank-lab/tree/cameron-donkin-v1/results/cameron-donkin

References

[1] A. A. Mullin, *Recursive function theory (a modern look at a Euclidean idea)*, Bulletin of the American Mathematical Society **69** (1963), 737. doi:10.1090/S0002-9904-1963-11017-4.

[2] P. J. Cameron, *Problems*, archived problem list, page dated August 31, 2013. https://cameroncounts.github.io/web/QM/oldprob.html. Accessed October 26, 2026.

[3] P. J. Cameron, *A prime problem*, October 5, 2012. https://cameroncounts.wordpress.com/2012/10/05/a-prime-problem/.

[4] Labos Elemer, *A093780*, The On-Line Encyclopedia of Integer Sequences, May 3, 2004; extended by David Wasserman, April 12, 2007. https://oeis.org/A093780. Accessed October 26, 2026.

[5] A. R. Booker and S. A. Irvine, *The Euclid–Mullin graph*, Journal of Number Theory **165** (2016), 30–57. doi:10.1016/j.jnt.2016.01.013. Numbering follows arXiv:1508.03039v2.

[6] M. Paris, *The Euclid–Mullin sequence as an orbit: reformulations, a composite floor, and population laws (with a Lean 4 formalization)*, preprint, repository snapshot September 24, 2026. `paper/short/main.tex`. github.com/marcellop71/EM, commit d04b90f.

[7] A. R. Booker, *A variant of the Euclid–Mullin sequence containing every prime*, Journal of Integer Sequences **19** (2016), Article 16.6.4. arXiv:1605.08929v1.

[8] A. R. Booker, *On Mullin's second sequence of primes*, Integers **12A** (2012), Article A4. arXiv:1107.3318v2.

[9] P. Pollack and E. Treviño, *The primes that Euclid forgot*, American Mathematical Monthly **121** (2014), 433–437. doi:10.4169/amer.math.monthly.121.05.433.

[10] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in Automated Deduction—CADE 28, Lecture Notes in Computer Science **12699** (2021), 625–635. doi:10.1007/978-3-030-79876-5_37.

[11] The mathlib Community, *The Lean mathematical library*, Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (2020), 367–381. doi:10.1145/3372885.3373824.