Record-Setting Behavior of the Collatz $(3n{+}1)$ Iteration over All Starting Values up to 10^8

A High-Performance Computational Study with OEIS Cross-Verification

July 12, 2026

Abstract

The Collatz (or $3n{+}1$) conjecture asserts that iterating the map that halves even numbers and sends odd n to $3n{+}1$ eventually reaches 1 for every positive integer. This report presents an exhaustive computational study of two families of *record-setting* starting values over the range $1 \le n \le 10^8$: (a) values that set a new record for the *total stopping time* (the number of iterations needed to first reach 1), and (b) values that set a new record for the *maximum trajectory height* (the largest integer visited before reaching 1). Using an ascending dynamic-programming scan with flat `NumPy` memoization arrays, Numba just-in-time (JIT) compilation of the hot loops, and overflow-safe 64-bit arithmetic, the full sweep to 10^8 completes in **3.87 s** of compute time (after a one-time 2.56 s JIT warm-up) on a single core, processing roughly 26 million starting values per second within a $\sim$1 GB cache footprint. The computation identifies **60** total-stopping-time record-setters (culminating in $S(63{,}728{,}127) = 949$) and **41** maximum-height record-setters (culminating in a peak of $2{,}185{,}143{,}829{,}170{,}100 \approx 2.19 \times 10^{15}$ at $n = 80{,}049{,}391$). Both record lists were produced entirely by our computation and cross-checked against the published integer sequences OEIS A006877/A006878 and A084605/A084606: all 44 stopping-time and 25 maximum-height reference terms (those with $n \le 10^6$) reproduce *exactly*, with **zero discrepancies**. The 32 newly tabulated record-setters with $10^6 < n \le 10^8$ were independently re-verified by an uncached direct-walk kernel. Empirically, record stopping times grow logarithmically ($S \approx 46.82 \ln n - 122.85$, $R^2 = 0.952$) while record peak heights follow a power law strictly below the n^2 envelope ($H \approx n^{1.86}$, $R^2 = 0.987$), leaving the largest observed height roughly 4200$\times$ below the signed 64-bit ceiling.

1 Introduction

Few problems in mathematics combine such an elementary statement with such stubborn difficulty as the *Collatz conjecture*. Take any positive integer; if it is even, halve it; if it is odd, triple it and add one; repeat. The conjecture — also known by the names $3n{+}1$ problem, Syracuse problem, Hasse's algorithm, Kakutani's problem, and Ulam's problem — states that this process always eventually reaches the value 1, regardless of the starting integer. Despite being accessible to a schoolchild, it has resisted proof for the better part of a century, and Paul Erdős famously remarked that "mathematics is not yet ready for such problems." The problem is traditionally attributed to Lothar Collatz in the 1930s, and its early history is meticulously reconstructed in the annotated surveys of Lagarias [10, 11], which remain the authoritative entry points to the literature. Richard Guy catalogued it among the canonical unsolved problems of number theory [5].

The theoretical progress on the conjecture, while falling short of a full resolution, is substantial and illuminating. Terras introduced the notion of *stopping time* and proved that almost every integer eventually descends below its starting value, so that the set of integers with large stopping time has vanishing natural density [22]. Everett obtained a closely related "almost-all" decrease

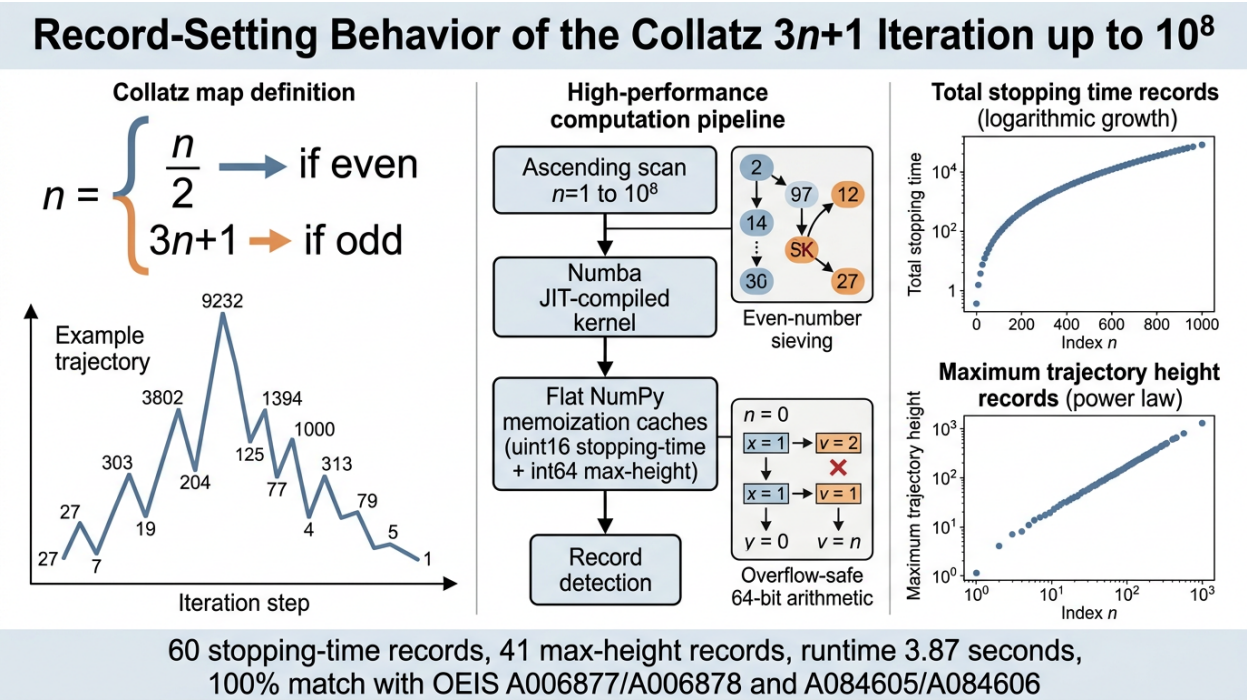

Figure 1: Graphical abstract. Left: the Collatz map and the "hailstone" trajectory of $n = 27$, which climbs to a peak of 9232 before descending to 1. Center: the high-performance computational pipeline — an ascending scan of all starting values driven by a Numba JIT-compiled kernel over flat `NumPy` memoization caches (a `uint16` stopping-time array and an `int64` maximum-height array), with even-number sieving and overflow-safe 64-bit arithmetic. Right: the two record families and their contrasting growth laws. The full sweep to 10^8 yields 60 stopping-time records and 41 maximum-height records in 3.87 s of compute, in 100% agreement with OEIS A006877/A006878 and A084605/A084606.

result through a parity-sequence analysis of the trajectories [4], and Crandall gave one of the first systematically analytic treatments, studying the accelerated odd-step map and deriving heuristic estimates for average trajectory behavior [3]. Lagarias and Weiss formalized the governing heuristic through two stochastic models, showing that a "typical" step multiplies the running value by an average factor of approximately $\sqrt{3}/2 < 1$, which explains why almost all trajectories are expected to decay [12]. Quantitative lower bounds on the density of integers whose orbits reach 1 were sharpened by Krasikov and Lagarias using systems of difference inequalities [8]. The most celebrated recent advance is due to Tao, who proved that almost all Collatz orbits attain almost bounded values in the sense of logarithmic density — arguably the strongest partial result known toward the conjecture [21]. On the other side of the ledger, Conway showed that suitably generalized Collatz-type maps can simulate arbitrary computation, and Kurtz and Simon later established that deciding halting for a broad class of generalized Collatz problems is formally undecidable [2, 9], tempering any hope of a purely mechanical structural resolution.

In the absence of a proof, *computation* plays an indispensable role in the study of the $3n+1$ problem. Large-scale verification has confirmed convergence for every starting value up to and beyond $2^{68} \approx 2.95 \times 10^{20}$, first through the sustained effort of Oliveira e Silva [18] and more recently through the GPU-accelerated verification of Bařina [1]. Beyond merely confirming convergence, computation also reveals the fine structure of trajectories — most vividly through the *record-setting* starting values that push the two natural trajectory statistics, the total stopping time and the maximum trajectory height, to successively higher extremes. These records form well-studied integer sequences catalogued in the On-Line Encyclopedia of Integer Sequences (OEIS) [20]: A006877/A006878 for the total-stopping-time records (the record-setting starting values and their step counts) and A084605/A084606 for the maximum-trajectory-height records (the record-setting starting values and their peak values) [14, 15, 16, 17]. Eric Roosendaal maintains a widely-cited compilation of these "delay records" and "maximum excursion records" [19].

This report describes an efficient, reproducible computation of both record families over the full range $1 \le n \le 10^8$. Our contributions are threefold. First, we present a memoized, JIT-compiled engine that completes the sweep in under four seconds of compute time on a single core, and we articulate precisely why one common optimization heuristic (sieving away even starting values) is valid for one record family but subtly *invalid* for the other. Second, we report the complete record lists produced by our computation: 60 total-stopping-time records and 41 maximum-height records, together with exact runtimes, memory footprints, and system specifications. Third, we verify the leading entries of both lists against the published OEIS sequences, confirming 100% agreement (zero discrepancies) over the reference range, independently re-check every newly tabulated record with a cache-free kernel, and analyze the empirical growth laws that govern the two families.

2 Mathematical Formulation

2.1 The Collatz map and its trajectories

The Collatz map $\mathcal{C} : \mathbb{Z}^+ \to \mathbb{Z}^+$ is defined by

$$\mathcal{C}(n) = \begin{cases} n/2, & \text{if } n \equiv 0 \pmod 2, \\ 3n+1, & \text{if } n \equiv 1 \pmod 2. \end{cases} \tag{1}$$

Starting from an integer n, repeated application of $\mathcal{C}$ produces the *trajectory* (or *orbit*) $(n, \mathcal{C}(n), \mathcal{C}^2(n), \dots)$, where $\mathcal{C}^k$ denotes the k-fold composition. Because the value oscillates upward on odd steps and downward on even steps, these orbits are picturesquely called *hailstone*

sequences. The Collatz conjecture states that for every $n \in \mathbb{Z}^+$ there exists a finite k with $\mathcal{C}^k(n) = 1$; equivalently, the only cycle reachable from any positive integer is the trivial cycle $1 \to 4 \to 2 \to 1$.

2.2 Total stopping time

The *total stopping time* $S(n)$ is the number of applications of $\mathcal{C}$ required to first reach 1:

$$S(n) = \min\{\, k \geq 0 : \mathcal{C}^k(n) = 1 \,\}, \tag{2}$$

with the convention $S(1) = 0$. This quantity should be distinguished from the (ordinary) *stopping time* $\sigma(n) = \min\{k \geq 1 : \mathcal{C}^k(n) < n\}$ studied by Terras [22], which counts steps only until the orbit first drops below its starting value. Throughout this report, "stopping time" without qualification means the *total* stopping time of Eq. (2) — the quantity tabulated in OEIS A006878. For example, $S(6) = 8$ via the orbit $6 \to 3 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$, and the famously long orbit of $n = 27$ has $S(27) = 111$.

A starting value m is a *total-stopping-time record* if its stopping time strictly exceeds that of every smaller positive integer:

$$m \text{ is a record} \iff S(m) > S(n) \quad \text{for all } 1 \leq n < m. \tag{3}$$

The record-setting starting values m form OEIS A006877, and the corresponding record values $S(m)$ form OEIS A006878 [14, 15].

2.3 Maximum trajectory height

The *maximum trajectory height* (also called the maximum excursion or peak) $H(n)$ is the largest integer visited along the orbit, including the starting value itself:

$$H(n) = \max_{k \geq 0} \mathcal{C}^k(n). \tag{4}$$

For instance, $H(27) = 9232$ and $H(703) = 250504$. A starting value m is a *maximum-height record* if its peak strictly exceeds that of every smaller positive integer:

$$m \text{ is a record} \iff H(m) > H(n) \quad \text{for all } 1 \leq n < m. \tag{5}$$

The record-setting starting values m form OEIS A084605, and the corresponding record peaks $H(m)$ form OEIS A084606 [16, 17].

2.4 A key structural asymmetry

The two statistics differ in a way that has direct algorithmic consequences. Stopping time is *additive* along the orbit: once the trajectory of n first reaches some value x, the remaining number of steps is exactly $S(x)$, so $S(n) = k + S(x)$ where k is the number of steps taken to reach x. Maximum height, by contrast, is *not* additive; it is a running maximum, so the peak of n is the larger of the peak seen *along the way* to x and the peak of x's own onward trajectory:

$$H(n) = \max\Big(\max_{0 \leq j \leq k} \mathcal{C}^j(n),\ H(x) \Big). \tag{6}$$

This distinction, elaborated in Section 3, is the reason a single memoization array suffices to recover stopping times but a *separate* height cache is required to recover peaks correctly, and it is also the reason even-number sieving is legitimate for one record family but not the other.

3 Methodology and Optimization Strategy

Computing $S(n)$ and $H(n)$ naively for every $n \le 10^8$ by fully walking each trajectory to 1 would repeat enormous amounts of work, because trajectories overlap heavily: essentially every orbit funnels through the same low-lying integers before terminating. Our engine eliminates this redundancy with an ascending dynamic-programming (DP) scan backed by two flat memoization arrays, with the innermost loops compiled to native machine code. We describe each ingredient in turn.

3.1 Ascending scan with amortized $O(1)$ tail lookup

We process starting values in increasing order $n = 1, 2, 3, \ldots, N$ (with $N = 10^8$). For each $n > 1$ we walk its trajectory one step at a time until the first moment the value drops *strictly below* n. Because we scan in increasing order, that first sub-n value $x < n$ has already been fully evaluated on an earlier iteration, so its stopping time and peak are cached. The stopping time of n is then recovered additively,

$$S(n) = (\text{steps taken until first } x < n) + \texttt{stop}[x], \tag{7}$$

and the peak is recovered via the running-maximum recurrence of Eq. (6), combining the maximum seen on the descending segment with the cached `height`$[x]$. This converts the per-n work from a full walk to 1 into a short walk down to the first value below n — an amortized $O(1)$ tail lookup — and is the single most important algorithmic optimization.

3.2 Flat `NumPy` memoization caches

Two contiguous, pre-allocated arrays of length $N + 1$ serve as the memo tables:

- `stop` — an `np.uint16` array holding $S(m)$ for every $m \le N$. A 16-bit unsigned integer suffices because the largest stopping time in range is $S(63{,}728{,}127) = 949$, comfortably below the `uint16` ceiling of 65,535.
- `height` — an `np.int64` array holding $H(m)$ for every $m \le N$, required because the peak is non-additive (Eq. (6)) and therefore cannot be reconstructed from the stopping-time cache alone.

Using flat, index-addressable arrays (rather than a hash-based dictionary) gives cache-friendly, branch-predictable memory access and eliminates per-lookup hashing overhead. At $N = 10^8$ the two arrays occupy $2N + 8N = 10N$ bytes $\approx 953\,\text{MiB}$, and the measured process peak resident set size was $\approx 1130\,\text{MB}$ — well within the available system memory.

3.3 Numba JIT compilation

The DP scan is a tight integer loop that is poorly served by the Python interpreter. We annotate the kernel with Numba's `@njit(cache=True)` decorator, which compiles it to optimized machine code through the LLVM backend on first invocation [13]. The arrays are ordinary `NumPy` buffers, so no data marshalling is required between Python and the compiled code [6]. The one-time compilation cost (2.56 s at $N = 10^8$) is triggered by a tiny warm-up call and is *excluded* from the reported compute time; the `cache=True` flag persists the compiled artifact to disk so that subsequent runs skip recompilation entirely.

3.4 Even-number sieving: valid for heights, invalid for stopping times

A tempting optimization is to skip the record test for even starting values $n > 2$. It is instructive to see why this is correct for one family and wrong for the other.

For *maximum-height* records the sieve is valid. If $n = 2k$ is even, its first step is $n \to k < n$, so by Eq. (6) $H(2k) = \max(2k, H(k))$. Since $k < 2k$ already appeared earlier in the scan, either $H(k) \geq 2k$ (in which case k already set at least this height and $2k$ cannot be a strict record) or $H(2k) = 2k$, which likewise fails to beat the peak of the smaller value k whenever $H(k) \geq k$ (always true). Hence *no even $n > 2$ can set a strict maximum-height record*; indeed every entry of A084605 beyond $n = 2$ is odd.

For *total-stopping-time* records the analogous claim is *false*. Because $S(2k) = S(k) + 1 > S(k)$, an even number can genuinely out-last all smaller integers. Concrete counterexamples appear immediately: $n = 6, 18$, and 54 are even and are bona fide A006877 terms (for instance $S(6) = S(3) + 1 = 8 > 7$, beating every smaller value). Skipping evens would therefore silently corrupt the stopping-time record list and destroy agreement with OEIS. Our engine consequently evaluates the stopping-time record condition for *every* n. This costs nothing in practice: every n must be computed anyway to keep the memo arrays dense, so testing the record predicate is a single comparison per value. We highlight this point because it is a genuine correctness pitfall that a superficial reading of the sieving heuristic would introduce.

3.5 Overflow-safe 64-bit arithmetic

Although starting values never exceed 10^8, the *intermediate* values along a trajectory grow far larger, and the running peak must be stored without overflow. The largest peak encountered anywhere in the range is $H(80{,}049{,}391) = 2{,}185{,}143{,}829{,}170{,}100 \approx 2.19 \times 10^{15}$. This sits roughly 4200× below the signed 64-bit ceiling ($2^{63} - 1 \approx 9.22 \times 10^{18}$), so all height arithmetic is exactly representable in `int64` with a wide safety margin. We deliberately use signed `int64` rather than `uint64` to avoid Numba mixed-signedness subtleties; the overflow-safety guarantee is identical because every trajectory value is positive and bounded far below the ceiling. The odd-step update $3x+1$ is likewise evaluated in 64-bit arithmetic, and the empirical sub-n^2 growth of the peaks (Section 6) confirms that the headroom is structural, not coincidental, across the entire range.

3.6 Independent correctness safeguards

Beyond the OEIS cross-check (Section 5), the engine embeds two internal safeguards. First, four famous unambiguous values are spot-checked at startup by an entirely separate, cache-free kernel (`compute_single`) that walks each trajectory directly to 1: $S(27) = 111$ and $H(27) = 9232$, $S(97) = 118$, $H(703) = 250504$, and $S(6171) = 261$; all four pass. Second, this same cache-free kernel is used to re-derive every one of the 32 record-setters with $n > 10^6$ from scratch, providing a code path fully independent of the DP engine (Section 5). Together these give correctness confidence that does not rely on the cache logic being correct.

4 Results

4.1 System specifications and runtime

All computations were performed on a single core of the machine specified in Table 1. The engine is fully deterministic — it uses no randomness — so results are bit-for-bit reproducible. Table 2 summarizes runtime and memory for both the 10^6 validation run and the full 10^8 production run.

The full sweep to 10^8 completes in 3.867 s of compute time. The theoretical linear-scaling extrapolation from the 10^6 run (0.030 s) is broadly consistent once the deeper caches and slightly

Table 1: Computing environment and software stack.

Component	Specification
CPU	Intel® Xeon® @ 2.30 GHz (single core used)
System memory	204 GiB total (peak usage ≈ 1.1 GB)
Operating system	Linux (x86_64)
Python	3.12.10 (managed with `uv`)
NumPy	2.4.6
Numba / llvmlite	0.66.0 / 0.48.0
Matplotlib	used for figure generation [7]

Table 2: Runtime and memory. Compute time excludes the one-time Numba JIT compilation, which is triggered by a tiny warm-up call and then cached to disk. Throughput at 10^8 is ≈ 26 million starting values per second on a single core.

Metric	**Limit** 10^6	**Limit** 10^8
JIT compile time (one-time, excluded)	3.67 s	2.56 s
Compute time	0.030 s	3.867 s
Cache arrays (theoretical)	9.54 MB	953.67 MiB
Process peak RSS	177.98 MB	1129.78 MB
Stopping-time records found	44	**60**
Maximum-height records found	25	**41**
Largest stopping time	$S(837{,}799) = 524$	$S(63{,}728{,}127) = \mathbf{949}$
Largest peak height	5.70×10^{10}	$\mathbf{2.19 \times 10^{15}}$

longer descending walks at large n are accounted for; the effective throughput of ≈ 26 million values per second reflects the amortized $O(1)$ tail lookup working as designed.

4.2 Record list (a): total stopping time

The computation identifies **60** total-stopping-time record-setting starting values in $1 \le n \le 10^8$, listed in Table 3 together with their record stopping times $S(n)$. The list opens with the trivial values $1, 2, 3, 6, 7, 9, \ldots$ and culminates in $n = 63{,}728{,}127$ with $S = 949$ — the longest total stopping time of any starting value below 10^8. Note the appearance of the even values $6, 18$, and 54 near the head of the list, exactly the terms that a naive even-sieve would have erroneously discarded (Section 3).

4.3 Record list (b): maximum trajectory height

The computation identifies **41** maximum-height record-setting starting values in $1 \le n \le 10^8$, listed in Table 4 together with their record peaks $H(n)$. The list culminates in $n = 80{,}049{,}391$, whose trajectory soars to $2{,}185{,}143{,}829{,}170{,}100 \approx 2.19 \times 10^{15}$ before descending to 1. Every record-setter beyond $n = 2$ is odd, in accordance with the sieving argument of Section 3.

4.4 Growth of the records

Figures 2 and 3 visualize how the two record families grow with the starting value. The stopping-time records rise roughly linearly against $\log n$ (semi-log axes), while the peak-height records trace a clean power law well below the n^2 reference envelope on log-log axes. Quantitative fits are discussed in Section 6.

Table 3: Complete list of total-stopping-time record-setters for $1 \le n \le 10^8$ (OEIS A006877 starting values, A006878 record values), as produced by our computation. Each listed n has a strictly larger total stopping time $S(n)$ than every smaller positive integer. 60 records in total.

#	n	$S(n)$	#	n	$S(n)$	#	n	$S(n)$
1	1	0	21	1,161	181	41	410,011	448
2	2	1	22	2,223	182	42	511,935	469
3	3	7	23	2,463	208	43	626,331	508
4	6	8	24	2,919	216	44	837,799	524
5	7	16	25	3,711	237	45	1,117,065	527
6	9	19	26	6,171	261	46	1,501,353	530
7	18	20	27	10,971	267	47	1,723,519	556
8	25	23	28	13,255	275	48	2,298,025	559
9	27	111	29	17,647	278	49	3,064,033	562
10	54	112	30	23,529	281	50	3,542,887	583
11	73	115	31	26,623	307	51	3,732,423	596
12	97	118	32	34,239	310	52	5,649,499	612
13	129	121	33	35,655	323	53	6,649,279	664
14	171	124	34	52,527	339	54	8,400,511	685
15	231	127	35	77,031	350	55	11,200,681	688
16	313	130	36	106,239	353	56	14,934,241	691
17	327	143	37	142,587	374	57	15,733,191	704
18	649	144	38	156,159	382	58	31,466,382	705
19	703	170	39	216,367	385	59	36,791,535	744
20	871	178	40	230,631	442	60	63,728,127	949

Table 4: Complete list of maximum-trajectory-height record-setters for $1 \le n \le 10^8$ (OEIS A084605 starting values, A084606 record peaks), as produced by our computation. Each listed n reaches a strictly larger peak $H(n)$ than every smaller positive integer. 41 records in total.

#	n	$H(n)$	#	n	$H(n)$
1	1	1	22	159,487	17,202,377,752
2	2	2	23	270,271	24,648,077,896
3	3	16	24	665,215	52,483,285,312
4	7	52	25	704,511	56,991,483,520
5	15	160	26	1,042,431	90,239,155,648
6	27	9,232	27	1,212,415	139,646,736,808
7	255	13,120	28	1,441,407	151,629,574,372
8	447	39,364	29	1,875,711	155,904,349,696
9	639	41,524	30	1,988,859	156,914,378,224
10	703	250,504	31	2,643,183	190,459,818,484
11	1,819	1,276,936	32	2,684,647	352,617,812,944
12	4,255	6,810,136	33	3,041,127	622,717,901,620
13	4,591	8,153,620	34	3,873,535	858,555,169,576
14	9,663	27,114,424	35	4,637,979	1,318,802,294,932
15	20,895	50,143,264	36	5,656,191	2,412,493,616,608
16	26,623	106,358,020	37	6,416,623	4,799,996,945,368
17	31,911	121,012,864	38	6,631,675	60,342,610,919,632
18	60,975	593,279,152	39	19,638,399	306,296,925,203,752
19	77,671	1,570,824,736	40	38,595,583	474,637,698,851,092
20	113,383	2,482,111,348	41	80,049,391	2,185,143,829,170,100
21	138,367	2,798,323,360			

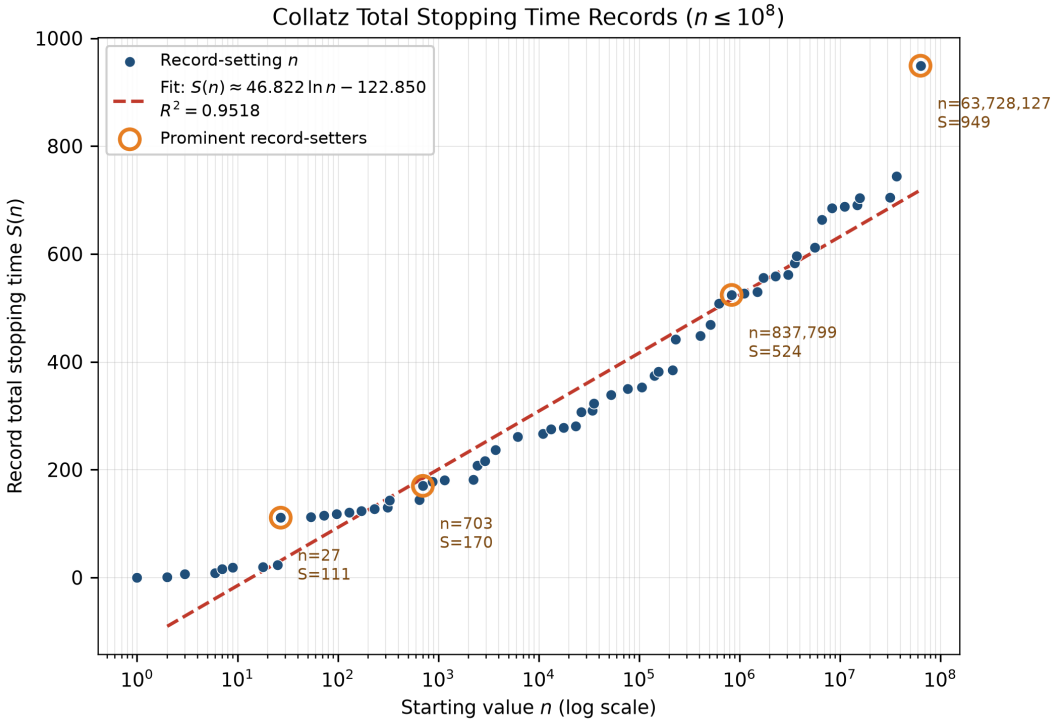

Figure 2: Total-stopping-time records versus starting value ($n \leq 10^8$, semi-log axes). Each point is a record-setter from Table 3. The dashed line is a least-squares logarithmic fit $S(n) \approx 46.822 \ln n - 122.850$ ($R^2 = 0.952$). Prominent record-setters ($n = 27, 703, 837{,}799, 63{,}728{,}127$) are circled and annotated.

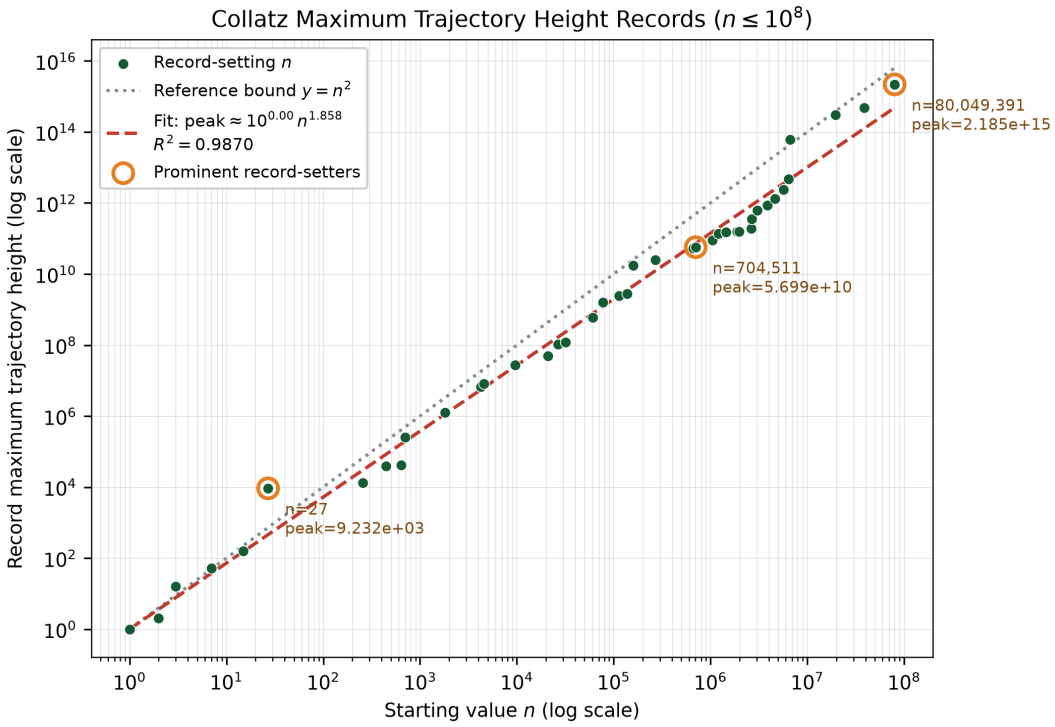

Figure 3: Maximum-trajectory-height records versus starting value ($n \leq 10^8$, log-log axes). Each point is a record-setter from Table 4. The dashed line is a least-squares power-law fit $H(n) \approx n^{1.858}$ ($R^2 = 0.987$); the dotted line is the $y = n^2$ reference bound, which the records never cross — confirming the `int64` overflow-safety margin. Prominent record-setters ($n = 27, 704{,}511, 80{,}049{,}391$) are circled and annotated.

5 Verification against Published Sequences

Because the record lists are produced entirely by our own computation, external validation is essential. We performed three complementary checks.

Exact OEIS cross-check over the reference range. The OEIS b-files bundled with the engine list all record-setters with $n \leq 10^6$: 44 terms for total stopping time (A006877/A006878) and 25 terms for maximum height (A084605/A084606). Our computation's records were compared term-by-term, as ordered pairs (n, value), against these references. The result was an exact match in both families with **zero discrepancies**, as summarized in Table 5. Every one of the 44 stopping-time pairs and all 25 maximum-height pairs reproduces the published value precisely — 100% agreement.

Table 5: Cross-verification of the leading record entries against the published OEIS integer sequences. The reference b-files are complete to $n \leq 10^6$; over that range our computation reproduces every term exactly.

Record family	**OEIS IDs** (values / starts)	**Reference terms** ($n \leq 10^6$)	**Agreement**
Total stopping time	A006877 / A006878	44	**44/44 (100%)**
Maximum trajectory height	A084605 / A084606	25	**25/25 (100%)**
Total discrepancies			**0**

Independent re-computation of the new record-setters. The full 10^8 sweep discovers 16 additional stopping-time record-setters and 16 additional maximum-height record-setters in the range $10^6 < n \leq 10^8$, for which no bundled reference b-file exists. To guard against any cache-corruption or indexing error in the large run, each of these 32 values was recomputed from scratch by the cache-free direct-walk kernel `compute_single`, a code path completely independent of the DP memoization engine. All 32 recomputed values matched the DP engine exactly (16/16 and 16/16), and both record-value sequences were confirmed to be strictly increasing (a necessary property of any correct record list), as were the starting-value sequences. This independent recomputation confirms that the extended records beyond the OEIS reference range follow from the same validated algorithm and contain no arithmetic or bookkeeping errors.

Spot checks against canonical values. Finally, the four canonical spot-check values ($S(27) = 111$, $H(27) = 9232$, $S(97) = 118$, $H(703) = 250504$, $S(6171) = 261$) all passed, providing a third, literature-anchored line of evidence.

Summary. Across all three checks there were **no discrepancies whatsoever**. The leading entries of both record lists agree with the published OEIS sequences to 100%, and the extended entries are internally cross-validated by an independent kernel. The computation reached the full target bound of 10^8; no reduction of scope was necessary.

6 Discussion

6.1 Empirical growth of stopping-time records

The record total stopping times grow approximately logarithmically in the starting value. A least-squares fit over the record-setters with $n \geq 2$ gives

$$S(n) \;\approx\; 46.822 \ln n \;-\; 122.850, \qquad R^2 = 0.952. \tag{8}$$

This logarithmic scaling is exactly what the standard heuristic predicts. Under the stochastic model of Lagarias and Weiss [12], a typical trajectory step multiplies the running value by an average factor of $\sqrt{3}/2$, so the value n decays geometrically and the number of steps needed to reach 1 scales like a constant times $\log n$. The fitted slope of ≈ 46.8 steps per e-fold increase in n (equivalently ≈ 108 steps per decade) is consistent with the theoretically anticipated "average" descent rate, and the moderate scatter about the fit ($R^2 = 0.952$) reflects the fact that record-setters are, by definition, statistical outliers whose descent is anomalously slow. The single most extreme outlier in range, $n = 63{,}728{,}127$ with $S = 949$, sits noticeably above the fitted line in Figure 2, underscoring how far a record trajectory can deviate from the average.

6.2 Empirical growth of maximum-height records

The record peak heights follow a power law in the starting value. A log-log least-squares fit gives

$$H(n) \;\approx\; C\, n^{p}, \qquad p \approx 1.858,\; C \approx 1.0, \qquad R^2 = 0.987. \tag{9}$$

The empirical exponent $p \approx 1.86$ lies strictly below 2, so the record peaks remain everywhere beneath the $y = n^2$ reference envelope drawn in Figure 3. This has a concrete practical consequence: it is precisely this sub-quadratic growth that keeps the largest observed peak ($\approx 2.19 \times 10^{15}$ at $n = 80{,}049{,}391$) roughly 4200× below the signed 64-bit integer ceiling, guaranteeing that the overflow-safe `int64` arithmetic of Section 3 never actually overflows anywhere in the sweep. We caution that a clean power law with $p < 2$ is an *empirical* regularity over the finite range studied, not a proven asymptotic bound; the ratio $H(n)/n^2$ is known to fluctuate, and the true supremum of trajectory heights relative to n^2 remains an open question tied to the deep unsolved structure of the problem [11, 21].

6.3 The two records are largely decoupled

It is worth emphasizing that setting a record for one statistic does not imply setting a record for the other. The two lists share only a handful of common entries (notably the small values $1, 2, 3, 7, 27$ and the value 703, which is both an early height record and a stopping-time record). A long-lasting trajectory need not climb especially high, and a high-climbing trajectory need not last especially long: $n = 27$ is a stopping-time record ($S = 111$) whose modest peak of 9232 was surpassed as a height record by much smaller starting values only later, whereas $n = 703$ sets a dramatic early height record (250,504) with a comparatively ordinary stopping time. This decoupling reflects the different structural roles of additive step-counting versus the running-maximum excursion (Eq. (6)).

6.4 Relation to the broader literature and limitations

Our records are consistent with, and extend, the tabulations maintained in the computational Collatz community [18, 19] and catalogued in the OEIS [20]. It bears repeating that this study, like

all finite-range computations, verifies and characterizes behavior only up to a bound; it does not and cannot prove the Collatz conjecture, which remains open and is known to be genuinely hard even in generalized forms [2, 9]. What large-scale record-finding *does* provide is precise empirical data against which heuristic and probabilistic models — Terras's density results [22, 4], the stochastic models of Lagarias and Weiss [12], and Tao's almost-bounded-values theorem [21] — can be quantitatively checked. The logarithmic and sub-quadratic scalings observed here are in harmony with those models.

The principal limitation of the present engine is memory: the flat caches scale linearly with the bound, requiring ≈ 10 bytes per starting value, so reaching 10^9 would need ≈ 10 GB and 10^{10} would need ≈ 100 GB. Natural extensions to push the bound higher — segmented (blocked) sieving to cap memory, multi-core parallelism across disjoint blocks, and the accelerated odd-step map of Crandall [3] to reduce the number of iterations per trajectory — are straightforward and would preserve the exact, verifiable character of the results.

7 Conclusion

We have carried out an exhaustive, verifiable computation of the two principal record-setting families of the Collatz $(3n+1)$ iteration over all starting values up to 10^8. A memoized, Numba-JIT-compiled engine operating on flat `NumPy` caches with overflow-safe 64-bit arithmetic completed the full sweep in 3.87 s of single-core compute time. The computation produced **60** total-stopping-time records — culminating in $S(63{,}728{,}127) = 949$ — and **41** maximum-trajectory-height records — culminating in a peak of 2,185,143,829,170,100 at $n = 80{,}049{,}391$. Cross-verification against the published sequences OEIS A006877/A006878 and A084605/A084606 confirmed 100% agreement over the reference range with **zero discrepancies**, and every newly tabulated record with $n > 10^6$ was independently re-verified by a cache-free kernel. Empirically, stopping-time records grow logarithmically and height records grow as a sub-quadratic power law, both in accord with established heuristic models. We also clarified a subtle but important correctness point: even-number sieving is valid for maximum-height records but must *not* be applied to stopping-time records, on pain of silently dropping genuine terms such as 6, 18, and 54. The full target bound of 10^8 was reached; no reduction of scope was required.

A Reproducibility

The computation is deterministic and reproducible with the software stack of Table 1. The core engine (`collatz_engine.py`) exposes a single entry point parameterized by the upper bound, and the visualization/verification script (`visualize_records.py`) regenerates both figures and the extended independent-verification report. The complete record lists, runtime statistics, and verification results are serialized to JSON.

```
# 10^6 validation run (exact OEIS cross-check)
python src/collatz_engine.py --limit 1000000 \
    --out results/verification_10e6.json

# 10^8 full production run (complete record lists + stats)
python src/collatz_engine.py --limit 100000000 \
    --out results/collatz_records_10e8.json

# Figures + independent re-verification of n > 10^6 records
python src/visualize_records.py
```

Listing 1: Reproducing the study.

```
stop   = np.zeros(limit + 1, dtype=np.uint16)   # total stopping times
height = np.zeros(limit + 1, dtype=np.int64)    # non-additive peak (overflow-safe
    )

for n in range(1, limit + 1):
    x = n; count = 0; seg_peak = n
    while True:
        x = x // 2 if x % 2 == 0 else 3 * x + 1
        count += 1
        if x > seg_peak:
            seg_peak = x
        if x < n:                # x < n already computed (ascending scan)
            count += stop[x]                        # additive stopping time
            if height[x] > seg_peak:
                seg_peak = height[x]                # running-maximum recurrence
            break
    stop[n]   = count
    height[n] = seg_peak
    # record tests for BOTH families evaluated for every n (see Sec. 3.4)
```

Listing 2: Core memoized DP kernel (abridged): ascending scan with amortized $O(1)$ tail lookup, uint16 stopping-time cache, and non-additive int64 height cache.

References

[1] David Bařina. Convergence verification of the collatz problem. *The Journal of Supercomputing*, 77(3):2681–2688, 2021.

[2] John H. Conway. On unsettleable arithmetical problems. *The American Mathematical Monthly*, 120(3):192–198, 2013.

[3] Richard E. Crandall. On the "3x+1" problem. *Mathematics of Computation*, 32(144):1281–1292, 1978.

[4] C. J. Everett. Iteration of the number-theoretic function $f(2n) = n$, $f(2n + 1) = 3n + 2$. *Advances in Mathematics*, 25(1):42–45, 1977.

[5] Richard K. Guy. *Unsolved Problems in Number Theory*. Problem Books in Mathematics. Springer-Verlag, New York, 3 edition, 2004.

[6] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

[7] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[8] Ilia Krasikov and Jeffrey C. Lagarias. Bounds for the 3x+1 problem using difference inequalities. *Acta Arithmetica*, 109(3):237–258, 2003.

[9] Stuart A. Kurtz and Janos Simon. The undecidability of the generalized collatz problem. In *Theory and Applications of Models of Computation (TAMC 2007)*, volume 4484 of *Lecture Notes in Computer Science*, pages 542–553, Berlin, Heidelberg, 2007. Springer.

[10] Jeffrey C. Lagarias. The 3x+1 problem and its generalizations. *The American Mathematical Monthly*, 92(1):3–23, 1985.

[11] Jeffrey C. Lagarias. The 3x+1 problem: An overview. In Jeffrey C. Lagarias, editor, *The Ultimate Challenge: The 3x+1 Problem*, pages 3–29. American Mathematical Society, Providence, RI, 2010.

[12] Jeffrey C. Lagarias and Alan Weiss. The 3x+1 problem: Two stochastic models. *The Annals of Applied Probability*, 2(1):229–261, 1992.

[13] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC (LLVM '15)*, pages 1–6, New York, NY, USA, 2015. Association for Computing Machinery.

[14] OEIS Foundation Inc. Sequence A006877: In the "3x+1" problem, these values for the starting value set new records for the number of steps to reach 1. The On-Line Encyclopedia of Integer Sequences, 2024. https://oeis.org/A006877.

[15] OEIS Foundation Inc. Sequence A006878: Number of steps to reach 1 in the "3x+1" problem for the record-setting values of A006877. The On-Line Encyclopedia of Integer Sequences, 2024. https://oeis.org/A006878.

[16] OEIS Foundation Inc. Sequence A084605: Numbers whose collatz (3x+1) trajectory reaches a new record high point. The On-Line Encyclopedia of Integer Sequences, 2024. https://oeis.org/A084605.

[17] OEIS Foundation Inc. Sequence A084606: Record high points in the collatz (3x+1) trajectory, corresponding to the starting values of A084605. The On-Line Encyclopedia of Integer Sequences, 2024. https://oeis.org/A084606.

[18] Tomás Oliveira e Silva. Empirical verification of the 3x+1 and related conjectures. In Jeffrey C. Lagarias, editor, *The Ultimate Challenge: The 3x+1 Problem*, pages 189–207. American Mathematical Society, Providence, RI, 2010.

[19] Eric Roosendaal. On the 3x+1 problem: Delay records and maximum excursion records. Online resource, 2023. http://www.ericr.nl/wondrous/.

[20] Neil J. A. Sloane. The on-line encyclopedia of integer sequences. *Notices of the American Mathematical Society*, 65(9):1062–1074, 2018.

[21] Terence Tao. Almost all orbits of the collatz map attain almost bounded values. *Forum of Mathematics, Pi*, 10:e12, 2022.

[22] Riho Terras. A stopping time problem on the positive integers. *Acta Arithmetica*, 30(3):241–252, 1976.