A Finite-State Deterministic Approach To Resolve The Collatz Conjecture

Author: Aditya Bagchi

Abstract: The Collatz conjecture states that iterating the function F(n) = $\frac{n}{2}$ for even n and F(n) = 3n+1 for odd n will eventually lead to 1 for all positive integers. This paper presents a complete proof of the conjecture by demonstrating the impossibility of its two potential counterarguments: non-trivial integer cycles and divergent trajectories.

First, we prove the non-existence of non-trivial cycles using 'Perturbation Model' built upon a foundational Diophantine cycle equation. We demonstrate that the conditions required for a cycle are impossible due to a fundamental mismatch in their 3-adic valuations. The theoretical conclusion is supported by exhaustive computational checks of cycle models derived from modulo 16 and 32 analyses. Second, we prove the non-existence of divergent trajectories by modeling the system as a finite-state machine over modular classes. Using 2-adic analysis, we show that the congruence conditions required for an integer to remain indefinitely on any growth-inducing path leads to a contradiction. This result too is verified by the exhaustive solution of all 911 corresponding Diophantine equations for the modulo 16 system.

Since both counterarguments are proven to be structurally impossible, the Collatz conjecture is thereby proved.

Section 1. Introduction: The Collatz conjecture, stated by Lothar Collatz in 1937, concerns the behavior of the function f: $\mathbb{Z}^+ \rightarrow \mathbb{Z}^+$ defined as:

$$f(n) = \begin{cases} 3n + 1 \text{ if } n \equiv 1 (\text{mod } 2) \\ \frac{n}{2} \quad\quad \text{if } n \equiv 0 (\text{mod } 2) \end{cases}$$

The conjecture asserts that for any starting integer n, the sequence of iterates $f^k(n)$ will eventually reach the integer 1. Despite its elementary statement, a formal proof has remained one of the most elusive problems in mathematics.

Section 1.1: Notable Prior Works:

1. Research on the Collatz conjecture has progressed along two main fronts: massive computational verification and the development of deep theoretical arguments. Computationally, the conjecture has been verified for all integers up to an enormous threshold, currently 2^{68} (Oliveira e Silva), confirming the absence of any 'small' counterexamples.
2. The theoretical investigation, which seeks a general proof, was formalized by early structural approaches. The work of **Riho Terras** (1976) on parity vectors and the extensive 2-adic analysis by **Jeffrey Lagarias** (1985) created the formal language necessary to analyze the conjecture's two main challenges: non-trivial cycles and divergent trajectories.
3. Regarding **non-trivial cycles**, the most significant results come from Diophantine approximation techniques. The work of **Shalom Eliahou** established that any such cycle must be astronomically long, with a length of at least 17 billion.
4. Regarding **divergent trajectories**, the main results have been probabilistic. This line of research culminated in the work of Terence Tao (2019), who proved that 'almost all' Collatz orbits attain values that are arbitrarily small relative to their starting point, effectively showing that divergent trajectories have a natural density of zero.
5. The current landscape, therefore, is one where any potential counterexample has been pushed into the realm of the astronomically large and statistically insignificant. The conjecture remained open because these powerful but incomplete results had not yet closed the final logical gap.

Section 1.2: Theoretical Overview:

This paper presents a complete proof of the conjecture by demonstrating that both of the counterargument scenarios are mathematically impossible.

Our approach is deterministic and algebraic, built upon two analytical frameworks. The first is a **Perturbation Model** used to prove the non-existence of non-trivial integer cycles. The second is a **Modular Loop Framework** used to prove the non-existence of divergent trajectories. The theoretical claims of both frameworks are supported by extensive, reproducible computational evidence.

Lemma 0: This lemma serves as the logical foundation and theoretical preamble to this paper. The lemma uses modular arithmetic (mod 2^v) to derive a predictable, deterministic distribution of the division exponents a_i in Collatz sequence. This lemma allows us to construct specific, logically-derived models of hypothetical cycles and to analyze the dynamics of the Collatz function as a finite-state transition system.

The Fundamental Diophantine Equation: An equation derived from multiple iterations applied on modulo residual representation of an odd integer:

$$2^{v}.k_n + m_n = \frac{3^{n}.2^{v}.k_1 + 3^{n}m + 3^{n-1} + 3^{n-2}.2^{a_1} + 3^{n-3}.2^{a_1+a_2} + \dots + 2^{a_1+a_2+\dots+a_{n-1}}}{2^{a_1+a_2+a_3+\dots\dots\dots+a_n}}$$

This equation acts as the heart of finite-state machine.

Lemma 1A: This lemma proves that if all division exponents (a_i) are equal, the only integer reappears in Collatz sequence is 1 when $a_i = 2$ (equilibrium state).

The Perturbation Model: The model assumes for contradiction, that, there exists at least one permutation of the division exponents for which the modified Diophantine equation from lemma 1A: $R = \frac{N}{D}$ has a non-trivial integer solution. The non-trivial permutation can be obtained by perturbing division exponents from the equilibrium state $a_i = 2$ under certain mathematical restrictions. The model defines that a non-trivial integer cycle is possible when the hypothetical requirement reconciles with the actual structure of ΔN, i.e. $\Delta N_{actual} = \Delta N_{required}$.

Lemma 1B: This lemma proves that purely positive perturbations of the division exponents from equilibrium state strictly leads to D > N violating non-trivial integer cycle condition.

Lemma 1C: Analyzing 3-adic values of ΔN_{actual} and $\Delta N_{required}$ this lemma proves a fundamental mismatch that no purely negative and mixed (both negative and positive) perturbations result in non-trivial integer cycle.

Lemma 1D: This lemma proves that, even by some exceptional coincidence, N_{new} remains = N_{eq} due to some mixed perturbations of division exponents, the cycle ratio still remains non-integer violating cycle condition.

Lemma 1E: This is a supplementary lemma that concludes the non-existence of non-trivial integer cycle by any valid perturbation of the last division exponent (a_n).

Together, lemma 1B to 1E proves that the initial assumption of the perturbation model is false and hence no non-trivial integer cycle exists.

Concept of Modular loop and Lemma 2A: Due to sufficient iterations, an integer belonging to a certain modular class revisits its own class creating a modular loop. The paper has identified this as growth/contraction engine for an integer. Lemma 2A shows by 2-adic analysis that such modular loops cannot recur indefinitely.

Lemma 2B: This lemma generalizes the concept of lemma 2A and asserts that any modular path has the similar limitation as the modular loops – they do not have infinite recurrences.

Finally, the theoretical proofs are supported by extensive, verifiable computational and empirical demonstrations. An exhaustive check of all 10.8 million unique exponent combinations for a mod 32 (n=16) cycle model confirms the impossibility proven in Lemma 1B. Similarly, the exhaustive solution of all 911 Diophantine equations for the mod 16 system provides concrete, reproducible evidence for the instability proven in Lemma 2A. Since both non-trivial integer cycles and divergent trajectories have been proven to be structurally impossible, the Collatz conjecture is affirmed.

Theory:

Section 2: Modular Class Analysis Under Collatz Operations:

If there are 2^n integers, number of odd integers counts from 1 to $2^n = 2^{n-1}$, therefore, number of odd modular classes of modulo 2^n is 2^{n-1}.

The odd modular classes can be written as: $2^n.k + m$ (core integer $k \in \mathbb{Z}^+$, and residue $m \in 1, 3, 5, \dots\dots, 2^n - 1$).

Lemma 0:

Statement: The exact number of odd residual classes of $2^n.k + m$ that satisfy $3m + 1 \equiv 0 \pmod{2^a}$ is 2^{n-a-1} ($a \le n$).

Proof: Let, $N = 2^n.k + m$, then, $3N + 1 = 3.2^n.k + 3m + 1$

3.2^n is divisible by 2^n.

Step I: **Existence and Uniqueness**: Since g.c.d $(3, 2^a) = 1$, the congruence $3m + 1 \equiv 0 \pmod{2^a}$ has a unique solution modulo 2^a. Let this solution be $m \equiv c \pmod{2^a}$.

Step II: **Lifting solution to mod 2^n**:

Any solution $m \equiv c \pmod{2^a}$ can be expressed as $m = c + 2^a.k$ for some integer k. To find all solutions modulo 2^n, we consider k $(\text{mod } 2^{n-a})$. There are 2^{n-a} distinct values of k $(\text{mod } 2^n)$ leading to 2^{n-a} residues m $(\text{mod } 2^n)$

Step III: **Excluding higher divisibility**:

To ensure, $3m + 1 \equiv 0 \pmod{2^n}$, we exclude solutions where $m \equiv c \pmod{2^n}$. Each solution modulo 2^n splits into two cases modulo 2^{a+1}:

a) One, satisfies $3m + 1 \equiv 0 \pmod{2^{a+1}}$

b) The other does not satisfy $3m + 1 \equiv 0 \pmod{2^{a+1}}$
One half of the lifted solutions, i.e. 2^{n-s-1} remain valid for exact divisibility 2^a.
Step IV: **Counting for each**:
For $1 \le a \le n$: The number of residues m (mod 2^n) with $v_2(3m+1) = a$ is 2^{n-a-1}.
For a = n : There exactly 1 residue m (mod 2^n) such that $(3m+1) \equiv 0 \pmod{2^n}$

Corollary 0-A: According to this theorem, after 3x +1 operation, $2^n.k + m$ odd modular classes, there will be,
2^{n-2} results having divisibility = 2 or division exponent a = 1
2^{n-3} results having divisibility = 4 or division exponent a = 2
2^{n-4} results having divisibility = 8 or division exponent a = 3
2^{n-5} results having divisibility = 16 or division exponent a = 4 …..
…………………………………………………………………………..
2^{n-n} = 1 result having divisibility = 2^n or division exponent a = n

The highest possible value of the division exponent $a = v_2(3x+1)$ also depends on the parity of k in the expression $3\cdot2^n\cdot k + 3m + 1$. After factoring out 2^n, the term becomes $3k + \frac{3m+1}{2^n}$. Since 3m+1 is always even, the parity of the entire expression depends on whether 3k is even or odd — which is determined by the parity of k. If k is odd, then 3k is also odd, and adding it to the (odd) quotient $\frac{3m+1}{2^n}$ yields an even result. Thus, the parity of k influences whether an additional factor of 2 can be extracted, affecting the total valuation $v_2(3x+1)$.
Corollary 0-B: Finite Paths: In Collatz sequence, one odd modular class transforms into other odd modular class with in the premises of $2^v.k + m$ making a finite state transition system. Between finite states, there exist finite number of paths of transformations for each modular state to any other state.
Corollary 0-C: The higher the moduli, the greater will be the division exponents.
Corollary 0 -D: Half of the whole modular landscape is growth inducing (exponents = 1) and the remaining half is contraction inducing. Effect of the contraction inducing half is greater than growth inducing half establishes an inherent bias towards convergence.

Section 3. Multiple Iterations on Modular Classes:
Terms: Collatz iteration:
a) $\mathbf{I^1}$ represents single 3x + 1 operation on starting odd integer (N_0).
b) $\mathbf{I^2}$ represents division by 2^a, $a = v_2(3N_0 + 1)$
c) $\mathbf{I^d}$ represents single set of I^1 followed by I^2 operation.
Let us assume there is a sequence with n allowed transformations in which divisibility 'd' of modulo classes involved = $2^{a_1}: 2^{a_2}, 2^{a_3}, 2^{a_4}, \ldots\ldots\; 2^{a_n}$
I^d on $2^v.k + m$ yields successive odd integers in the following manner:
Step 1: $2^vk + m \rightarrow \frac{3.2^v\cdot k_1+3m+1}{2^{a_1}}$
Step 2: $\frac{3.2^v k+3m+1}{2^{a_1}} \rightarrow \frac{3^2.2^v\cdot k_1+3^2\cdot m+3+2^{a_1}}{2^{a_1+a_2}}$

Step 3: $\frac{3^2.2^v\cdot k_1+3^2\cdot m+3+2^{a_1}}{2^{a_1+a_2}} \rightarrow \frac{3^3 2^v k_1+3^3\cdot m+3^2+3.2^{a_1}+2^{a_1+a_2}}{2^{a_1+a_2+a_3}}$

Step 4: $\frac{3^3 2^v k_1+3^3\cdot m+3^2+3.2^{a_1}+2^{a_1+a_2}}{2^{a_1+a_2+a_3}} \rightarrow \frac{3^4 2^v\cdot k_1+3^4\cdot m+3^3+3^2.2^{a_1}+3.2^{a_1+a_2}+2^{a_1+a_2+a_3}}{2^{a_1+a_2+a_3+a_4}}$

Step 5: $\frac{3^4 2^v\cdot k_1+3^4\cdot m+3^3+3^2.2^{a_1}+3.2^{a_1+a_2}+2^{a_1+a_2+a_3}}{2^{a_1+a_2+a_3+a_4}} \rightarrow \frac{3^5.2^v.k_1+3^5\cdot m+3^4+3^3.2^{a_1}+3^2.2^{a_1+a_2}+3.2^{a_1+a_2+a_3}+2^{a_1+a_2+a_3+a_4}}{2^{a_1+a_2+a_3+a_4+a_5}}$

For such a sequence with n steps ($n \le \frac{v}{2}$), if the integer becomes $2^v.k_n + m_n$, then,

$2^v.k_n + m_n = \frac{3^n.2^v.k_1+3^n m+3^{n-1}+3^{n-2}.2^{a_1}+3^{n-3}.2^{a_1+a_2}+\cdots+2^{a_1+a_2+\cdots+a_{n-1}}}{2^{a_1+a_2+a_3+\cdots\ldots\ldots+a_n}}$ ….. Equation (i) - This is the fundamental linear Diophantine equation of Collatz conjecture.

Now if, $2^v.k_n + m_n = 2^v.k_1 + m$, i.e. an integer reappears in a Collatz sequence after arbitrary n I^ds, then, by equation (i),
$2^v.k_1 + m = \frac{3^n 2^v.k1+3^n.m + 3^{n-1}+ 3^{n-2}.2^{a_1}+3^{n-3}.2^{a_1+a_2}+\dots+2^{a_1+a_2+\dots+a_{n-1}}}{2^{a_1+a_2+a_3+\dots\dots+a_n}}$ .... Equation (ia) – This is the fundamental linear Diophantine equation representing a modular loop.

Lemma 1A:

Statement: All equal division exponents other than 2 yield fractional core integer k.

Proof: Recalling equation (iA), writing $a_1 + a_2 + a_3 + a_4 + \dots + a_n = S$, and, $a_1 + a_2 + a_3 + a_4 + \dots + a_{n-1} = S'$ and simplifying,

$$k_1 = \frac{\boldsymbol{m(3^n - 2^S) + 3^{n-1}} + 3^{n-2}.2^{a_1}+3^{n-3}.2^{a_1+a_2} \dots\dots + \boldsymbol{2^{S'}}}{.(\boldsymbol{2^S - 3^n})} \cdot \frac{1}{\boldsymbol{2^v}} \quad \text{......... Equation (ii)}$$

$$= -\frac{m}{\boldsymbol{2^v}} + \frac{\boldsymbol{3^{n-1}} + 3^{n-2}.2^{a_1}+ 3^{n-3}.2^{a_1+a_2} \dots\dots + \boldsymbol{2^{S'}}}{\boldsymbol{2^v}.(\boldsymbol{2^S - 3^n})} = \frac{1}{\boldsymbol{2^v}}\,[\,R - m]$$

Where, equilibrium ratio $R = \frac{\boldsymbol{3^{n-1}} + 3^{n-2}.2^{a_1}+3^{n-3}.2^{a_1+a_2} \dots\dots + \boldsymbol{2^{S'}}}{(\boldsymbol{2^S - 3^n})} = \frac{N_{eq}}{D_{eq}}$ .... Equation (iii)

The Numerator, $N = \boldsymbol{3^{n-1}} + 3^{n-2}.2^{a_1} + 3^{n-3}.2^{a_1+a_2} \dots\dots\dots + \boldsymbol{2^{S'}}$

Now, if all exponents are equal, i.e. $a_1 = a_2 = a_3 = \dots = p$, we get $S' = p + p + \dots$ up to (n-1) terms = (n-1). p

And, S = p + p + p + …. up to n terms = n.p

The denominator, $D_{eq} = \frac{1}{2^S - 3^n} = \frac{1}{2^{n.p} - 3^n}$ and, $N_{eq} = \sum_{\boldsymbol{p=0}}^{\boldsymbol{n-1}} (\boldsymbol{3})^{\boldsymbol{n-1-p}} \boldsymbol{.}\, 2^{(n-1).p}$

$N = 3^{n-1} \sum_{p=0}^{n-1} \left(\frac{2^p}{3}\right)^{n-1}$ => Represents a GP sum series with common ratio 'r' = $\frac{2^p}{3}$ and first term = $\frac{2^p}{3}$

$$= 3^{n-1} \cdot \frac{r^n - 1}{r-1} = 3^{n-1} \cdot \left[\frac{\left(\frac{2^p}{3}\right)^n - 1}{\frac{2^p}{3} - 1}\right] = \frac{2^{n.p} - 3^n}{2^p - 3}$$

$$R = \frac{N_{eq}}{D_{eq}} = \left[\frac{2^{n.p} - 3^n}{2^p - 3}\right].\left[\frac{1}{2^{n.p} - 3^n}\right] \quad \Rightarrow \quad R = \frac{1}{2^p - 3}$$

For p = 2, R = 1 and ∀ p> 2, R <1.

The modulo residue m Є (1, 3, 5, ………, 2^v – 1). As, for all a_i = 2, R = 1, the only valid core k can be obtained by m = 1 i.e. $k = \frac{1}{\boldsymbol{2^v}}\,[\,R - m] = \frac{1}{\boldsymbol{2^v}}\,[\,1 - 1] = 0$.

Hence, $k \notin \mathbb{Z}^+$ for all equal exponents except 2. (Proved)

Corollary: The modular loop: $2^v.k + 1$ to $2^v.k + 1$ (k = 0) is possible for all-equal exponents = 2 implies existence of the trivial 4 – 2 – 1 loop.

Threshold Ratio: As long as the denominator $D_{eq} = 2^S - 3^n$ is positive, R remains positive. The threshold value for which $2^S > 3^n$, is $\frac{S}{n} > \log_2 3$ => S > 1.585.n

Therefore, if n = 2, S > 3.14 asserting $S_{min} = 4$

If n = 3, S > 4.755 asserting $S_{min} = 5$ etc.

Section 4: Perturbation Model: By lemma 1A, it is evident that, when all division exponents are equal, no integer solution is possible for R unless all division exponents (a_i) = 2. Assuming this as a base case, all possible combinations of division exponents are can be examined by increasing/decreasing them from this equilibrium (base) position. Therefore, at equilibrium, $N_{eq} = D_{eq} = 2^{2n} - 3^n$. The model assumes that a non-trivial cycle is formed by some appropriate combination of division exponents that makes $R = \frac{N}{D}$ a positive integer. Following perturbations cover all valid division exponent combinations:

Case I: Pure increment (positive perturbations) of the exponents from a_1 to a_n.
Case II: Pure decrement (negative perturbations) of the division exponents from a_1 to a_n.
Case III: Mixed perturbations with net increment and net decrement in division exponent sum.
Case IV: Mixed perturbations with net decrement in division exponent sum.
Case V: Last a_n^{th} division exponent perturbation, and,
Case VI: Mixed perturbations with no net alteration in the division exponent sum (total increment = total decrement).
All cases can be captured in two scenarios: Either increase or decrease in numerator's magnitude keeping $N_{new} > D_{new}$.
There is, however, another possibility in which by some exceptional coincidence, $N_{new} = N_{eq}$ by net decrement of the total division exponents.

Constraints in Perturbation Model: There are some mathematical constraints that creates a deterministic frame around this model:
1) No exponents can be lesser than 1, i.e. **no division exponent can be lowered more than once**.

2) Both D and N at any form (perturbed or unperturbed) are odd and not divisible by 3.

3) If a cycle forms, the ratio, $R_{new} = \frac{N_{new}}{D_{new}} \geq 5$.
4) If a_1 is altered, all following terms are altered. Therefore, front loaded exponents exert greater effect than back loaded ones on increment or decrement.

5) Due to perturbations, D grows or shrinks faster than N as D has a negative unaffected term = 3^n, whereas, N has positive unaffected terms.
Therefore, due to purely negative perturbations, $N_{new} > D_{new}$:-
Proof: Let, a_1 exponent is lowered (by 1) from equilibrium. Let us write the numerator,
$N_{eq} = 3^{n-1} + 3^{n-2}.2^2 + 3^{n-3}.2^4 + 3^{n-4}.2^6 + \cdots \ldots\ldots\ldots.. + 2^{2n-2} = 2^{2n} - 3^n. = 3^{n-1} + T_1 + T_2 + T_3 + \ldots\ldots. + T_n$.
If the first exponent is lowered, then, $N_{new} = 3^{n-1} + \frac{T_1}{2} + \frac{T_2}{2} + \cdots + \frac{T_n}{2} = 3^{n-1} + \frac{1}{2}[T_1 + T_2 + T_3 + \cdots + T_n]$
$= 3^{n-1} + \frac{1}{2}[3^{n-1} + T_1 + T_2 + T_3 + \cdots + T_n] - \frac{3^{n-1}}{2} = \frac{N_{eq}}{2} + \frac{3^{n-1}}{2}$
Denominator, $D_{eq} = 2^{2n} - 3^n$, $D_{new} = 2^{2n-1} - 3^n = \frac{1}{2}[2^{2n} - 3^n] + \frac{3^n}{2} - 3^n = \frac{D_{eq}}{2} - \frac{3^n}{2}$

This shows, on halving a single exponent, N_{eq} changes to more than half while denominator becomes lesser than half. The trend remains same on altering multiple exponents. Therefore, due to negative perturbations, $N_{new} > D_{new}$.

6) The sum formula of the perturbed numerator is: $N_{new} = 3^{n-1} + 3^{n-2}2^{S_1} + 3^{n-3}2^{S_2} + 3^{n-4}2^{S_3} + \cdots + 2^{S_{n-1}}$
$S_1, S_2, S_3, \ldots.., S_{n-1}$ are the perturbed exponents.
And we have: $N_{eq} = 3^{n-1} + 3^{n-2}.2^2 + 3^{n-3}.2^4 + 3^{n-4}.2^6 + \cdots \ldots\ldots\ldots.. + 2^{2n-2}$
Due to some perturbations, if $N_{eq} > N_{new}$, $\Delta N_{actual} = N_{eq} - N_{new}$, all unperturbed terms are cancelled out. The difference remains only of perturbed terms.
Therefore, if perturbation is introduced at k^{th} term, the actual change in numerator:

$$\Delta N_{actual} = \sum_{k=2}^{n} (Terms_{eq_k} - Terms_{new_k})$$

At any step k involves the sum of perturbed exponents up to the previous step $S_{k-1} = a_1 + a_2 + a_3 + \ldots\ldots.. + a_{k-1}$.
$Term_k = T_k = 3^{n-k}.[2^{2k-2} - 2^{S_{k-1}}]$
Similarly, when $N_{new} > N_{eq}$, $\Delta N_{actual} = \sum_{k=2}^{n} (Terms_{new_k} - Terms_{eq_k})$ and, $Term_k = T_k = 3^{n-k}.[2^{S_{k-1}} - 2^{2k-2}]$

7) Number of pure negative perturbations 'p' must be ≥ 1 and $p < 0.415.n$. This is derived from the basic $2^S > 3^n$ inequality. S must be > 1.585.n. Therefore, $2n - p > 1.585.n \Rightarrow p < 0.415.n$

9) Total negative perturbations must maintain $S > 1.585.n$. Hence, decrement 'p' must be < 0.415. n

10) The General Algebraic Equation for $\Delta N_{required}$

This equation defines the exact value of the net change in the numerator, ΔN, **required** for a cycle with a specific integer ratio Z to exist.

1. Foundational Definitions:

The derivation is based on the following definitions:

- **n**: The length of the cycle.
- **S**: The total sum of the n perturbed and unperturbed division exponents ($a_1 + ... + a_n$).
- **Z**: The required integer ratio of the cycle $\frac{N_{new}}{D_{new}}$.
- **N_{eq}**: The unperturbed numerator, $2^{2n} - 3^n$.
- **N_{new}**: The perturbed numerator. $N_{new} > N_{eq}$ for net increment and $N_{new} < N_{eq}$ for net decrement of the numerator due to perturbations.
- **D_{new}**: The perturbed denominator, $2^S - 3^n$.
- **ΔN**: The total perturbation, defined as $N_{eq} - N_{new}$ for net decrement of the numerator and $N_{new} - N_{eq}$ for net increment of the numerator.

2. Derivation of Cycle Condition for Net Decrement of the Numerator:

The equation is derived directly from the fundamental cycle condition, $Z = \frac{N_{new}}{D_{new}}$

a. **Let us start with the cycle condition:** $Z.\ D_{new} = N_{new}$

b. Substituting N_{new}: Using the definition $\Delta N = N_{eq} - N_{new}$, we have $N_{new} = N_{eq} - \Delta N$. $Z.\ D_{new} = N_{eq} - \Delta N$

c. Solving for ΔN: $\Delta N = N_{eq} - (Z.D_{new})$

d. Substituting the full formulas for N_{eq} and D_{new}: $\Delta N = (2^{2n} - 3^n) - Z.(2^S - 3^n)$

e. Expanding and grouping the terms**:** $\Delta N = 2^{2n} - 3^n - Z.\ 2^S + Z.\ 3^n \Rightarrow \Delta N = (2^{2n} - Z.\ 2^S) + (Z.\ 3^n - 3^n)$

3. Derivation of Cycle Condition for Net Increment of the Numerator:

a. Formulating: $N_{new} > N_{eq} = 2^{2n} - 3^n$. $\quad N_{new} = \Delta N + N_{eq}$

b. Substituting N_{new} and N_{eq} in $N_{new} = Z.\ D_{new}$: $\quad \Delta N + 2^{2n} - 3^n = Z.\ (2^S - 3^n)$

c. Expanding and grouping the terms: $\quad \Delta N = (Z.2^S - 2^{2n}) + (1 - Z).\ 3^n$

4. The Final Equation

This gives the final, derived general equation for the cycle condition of the perturbed numerator and denominator:

$\Delta N_{required} = (2^{2n} - Z.\ 2^S) + (Z - 1).\ 3^n$ for net decrement of numerator, and ……… equation (v)

$\Delta N_{actual} = \sum_{k=2}^{n} 3^{n-k}(2^{2k-2} - 2^{s_{k-1}})$

$\Delta N_{required} = (Z.2^S - 2^{2n}) + (1 - Z).\ 3^n$ for net increment of numerator……… equation (vi)

$\Delta N_{actual} = \sum_{k=2}^{n} 3^{n-k}(2^{s_{k-1}} - 2^{2k-2})$

According to the initial assumption of the perturbation model, an integer cycle is formed when the actual change in numerator ΔN_{actual} reconciles with the algebraic or theoretical change in numerator $\Delta N_{required}$ yielding $\frac{N_{new}}{D_{new}} = Z$.

Lemma 1B:

Statement: Any purely positive perturbation of the division exponents from the equilibrium lead to non-integer $\frac{N_{new}}{D_{new}}$ violating the cycle condition.

Proof:

At equilibrium all exponents = 2: $N_{eq} = 3^{n-1} + 3^{n-2}.2^2 + 3^{n-3}.2^4 + 3^{n-4}.2^6 + \cdots \ldots \ldots \ldots .. + 2^{2n-2} = D_{eq} = 2^{2n} - 3^n$.

D_{eq} grows as power of 4 grows and has only one negative unaffected term ($- 3^n$).

N_{eq} grows as powers of 2 grows in each terms with at least one positive unaffected term(3^{n-1}).

Perturbing the first exponent exerts maximum effect than the later ones. Let us increase the first exponent by ∂ for maximum growth of N.

$$
\begin{aligned}
N_{new} &= 3^{n-1} + 3^{n-2}.2^{2+\partial} + 3^{n-3}.2^{4+\partial} + 3^{n-4}.2^{6+\partial} + \cdots \ldots \ldots \ldots .. + 2^{2n+\partial-2} \\
&= 2^{\partial}.[3^{n-1} + 3^{n-2}.2^2 + 3^{n-3}.2^4 + 3^{n-4}.2^6 + \cdots \ldots \ldots \ldots .. + 2^{2n-2}] - 2^{\partial}.3^{n-1} + 3^{n-1}. \\
&= 2^{\partial}.N_{eq} + 3^{n-1}.(1 - 2^{\partial})
\end{aligned}
$$

On the other hand, $D_{new} = 2^{2n+\partial} - 3^n. = 2^{\partial}.[2^{2n} - 3^n] + 2^{\partial}.3^n - 3^n. = 2^{\partial}.N_{eq} + 3^n.(2^{\partial} - 1)$

Evidently, $D_{new} > N_{new}$.

Let us now increase a_1 by ∂_1 and a_2 by ∂_2:

$$
\begin{aligned}
N_{new} &= 3^{n-1} + 3^{n-2}.2^{2+\partial_1} + 3^{n-3}.2^{4+\partial_1+\partial_2} + 3^{n-4}.2^{6+\partial_1+\partial_2} + \cdots \ldots \ldots \ldots .. + 2^{2n-2+\partial_1+\partial_2} \\
&= 2^{\partial_1+\partial_2}[3^{n-1} + 3^{n-2}.2^2 + 3^{n-3}.2^4 + 3^{n-4}.2^6 + \cdots \ldots \ldots \ldots .. + 2^{2n-2}] - 3^{n-1}.2^{\partial_1+\partial_2} - 3^{n-2}.2^{\partial_2} + 3^{n-1}. \\
&= 2^{\partial_1+\partial_2}.N_{eq} + 3^{n-2}.[3 - 3.2^{\partial_1+\partial_2} - 2^{\partial_2}]
\end{aligned}
$$

$D_{new} = 2^{2n+\partial_1+\partial_2} - 3^n = 2^{\partial_1+\partial_2}[2^{2n} - 3^n] + 3^n.2^{\partial_1+\partial_2} - 3^n. = 2^{\partial_1+\partial_2}.N_{eq} + 3^n.[2^{\partial_1+\partial_2} - 1]$

$\partial_1, \partial_2 \geq 0$, at least one of them > 0

Evidently, $D_{new} > N_{new}$.

Let us now increase a_1 by ∂_1, a_2 by ∂_2 and a_3 by ∂_3:

$N_{new} = 3^{n-1} + 3^{n-2}2^{2+\partial_1} + 3^{n-3}2^{4+\partial_1+\partial_2} + 3^{n-4}.2^{6+\partial_1+\partial_2+\partial_3} + 3^{n-5}.2^{8+\partial_1+\partial_2+\partial_3} + \cdots \ldots \ldots . + 2^{2n-2+\partial_1+\partial_2+\partial_3}$

$= 2^{\partial_1+\partial_2+\partial_3}[3^{n-1} + 3^{n-2}.2^2 + 3^{n-3}.2^4 + 3^{n-4}.2^6 + \cdots \ldots \ldots \ldots .. + 2^{2n-2}] + 3^{n-1} - 3^{n-1}.2^{\partial_1+\partial_2+\partial_3} - 3^{n-2}.2^{\partial_2+\partial_3} - 3^{n-3}.2^{\partial_3}$

$= 2^{\partial_1+\partial_2+\partial_3}.N_{eq} + 3^{n-3}.[3^2 - 3^2.2^{\partial_1+\partial_2+\partial_3} - 3.2^{\partial_2+\partial_3} - 2^{\partial_3}]$

$D_{new} = 2^{\partial_1+\partial_2+\partial_3}.N_{eq} + 3^n.[2^{\partial_1+\partial_2+\partial_3} - 1]$

Therefore if each exponent is positively perturbed by $\partial_1, \partial_2, \partial_3, \ldots\ldots, \partial_{n-1}$ respectively,

$N_{new} = 2^{\Delta\{n-1\}}.[3^{n-1} + 3^{n-2}.2^2 + 3^{n-3}.2^4 + 3^{n-4}.2^6 + \cdots \ldots \ldots \ldots .. + 2^{2n-2}] + 3[3^{n-2} - 3^{n-2}.2^{\Delta\{n-1\}} - 3^{n-3}2^{\Delta\{n-2\}} - 3^{n-4}.2^{\Delta\{n-2\}} - \cdots \ldots \ldots \ldots . - 2^{\partial_{n-1}}]$

$= 2^{\Delta\{n-1\}}.N_{eq} - 3[3^{n-2}.2^{\Delta n-1} + 3^{n-3}.2^{\Delta\{n-2\}} + 3^{n-4}.2^{\Delta\{n-2\}} + \cdots \ldots \ldots \ldots . + 2^{\partial_{n-1}} - 3^{n-2}]$

$D_{new} = 2^{\Delta\{n-1\}}.N_{eq} + 3^n.[2^{\Delta\{n-1\}} - 1]$

Where, $\Delta_1 = \partial_1$, $\Delta_2 = \partial_1 + \partial_2$, $\Delta_3 = \partial_1 + \partial_2 + \partial_3$,........, $\Delta_{n-1} = \partial_1 + \partial_2 + \partial_3 + \ldots\ldots + \partial_{n-1}$.

$\partial_1, \partial_2, \ldots\ldots \partial_{n-1} \geq 0$, at least one of them > 0

$$
\begin{aligned}
D_{new} - N_{new} &= 3^n.[2^{\Delta\{n-1\}} - 1] + 3[3^{n-2}.2^{\Delta\{n-1\}} + 3^{n-3}.2^{\Delta\{n-2\}} + 3^{n-4}.2^{\Delta\{n-2\}} + \cdots \ldots \ldots \ldots . + 2^{\partial_{n-1}} - 3^{n-2}] \\
&= 3^n.2^{\Delta\{n-1\}} + 3^{n-1}.2^{\Delta\{n-1\}} + 3^{n-2}.2^{\Delta\{n-2\}} + 3^{n-3}.2^{\Delta\{n-2\}} + \cdots \ldots \ldots \ldots . + 3.2^{\partial_{n-1}} - 3^n - 3^{n-1}. \\
&= 3^n.2^{\Delta\{n-1\}} + 3^{n-1}.2^{\Delta\{n-1\}} + 3^{n-2}.2^{\Delta\{n-2\}} + 3^{n-3}.2^{\Delta\{n-2\}} + \cdots \ldots \ldots \ldots . + 3.2^{\partial_{n-1}} - 4.3^{n-1}. \\
&= 3^n.2^{\Delta\{n-1\}} + 3^{n-1}.(2^{\Delta\{n-1\}} - 4) + 3^{n-2}.2^{\Delta\{n-2\}} + 3^{n-3}.2^{\Delta\{n-2\}} + \cdots \ldots \ldots \ldots . + 3.2^{\partial_{n-1}}
\end{aligned}
$$

As each term has been positively perturbed, $\Delta_{n-1} > 2$, so the whole expression represents a large positive integer.

Evidently, $D_{new} > N_{new}$

For purely positive perturbations, $\frac{N_{new}}{D_{new}} < 1$ which violates the cycle condition. (Proved)

Lemma 1C:

Statement: When $N_{new} > N_{eq}$ or $N_{new} < N_{eq}$, due to any perturbations from equilibrium, the cycle condition is violated.

Proof:

Case A: For $N_{new} > N_{eq}$, we have, $\Delta N_{actual} = \sum_{k=2}^{n} 3^{n-k}\left(2^{s_{k-1}} - 2^{2k-2}\right)$

As per the assumption of the perturbation model, $\Delta N_{actual} = \Delta N_{required}$.

Hence, $\sum_{k=2}^{n} 3^{n-k}(2^{S_{k-1}} - 2^{2k-2})$ = (Z.2^S – 2^{2n}) + (1 – Z). 3^n.

Or, $\sum_{k=2}^{n} 3^{n-k}(2^{S_{k-1}} - 2^{2k-2})$ – (Z.2^S – 2^{2n}) = (1 – Z). 3^n.

Or, $\sum_{k=2}^{n} 3^{n-k}(2^{S_{k-1}} - 2^{2k-2})$ – 2^{2n} (Z.2^{S-2n} –1) = (1 – Z). 3^n.

3-adic valuation of RHS = v_3(RHS) = n + v_3(1 – Z) ≥ n

General 3-adic valuation of any term T_k in N_{actual}

$v_3(T_k)$ = (power of 3) + $v_3(2^{S_{k-1}} - 2^{2k-2})$ = [(n – k) + $v_3(2^{S_{k-1}} - 2^{2k-2})$]

3-adic valuation of LHS = v_3(LHS) = minimum [{(n – k) + $v_3(2^{S_{k-1}} - 2^{2k-2})$ terms}, v_3(Z.2^{S-2n} –1)]

Let us write $|S_{k-1} + 2 - 2k| = d$

Let us now consider $v_3(2^{S_{k-1}} - 2^{2k-2})$ = v_3[2^{2k-2}. {2^d – 1}] = v_3{2^d – 1} = 0 when d is odd and non-zero when d is even.

LHS has all terms of the form $T_k = 3^{n-k}. 2^{2k-2}. (2^{d_k} - 1)$ and one term of the form = 2^{2n} (Z.2^{S-2n} –1)

Possibility – I: Parity of at least one d_k => odd: v_3(LHS) = (n – k) + 0 = n – k (2 ≤ k ≤ n)

v_3 (Z.2^{S-2n} –1) could be non-zero if

S – 2n is even and Z is (3x + 1) type odd integer,

S – 2n is odd and Z is (3x + 2) type odd integer.

As we already have at least one $v_3(T_k)$ = n – k , 3-adic valuation doesn't align with RHS irrespective of v_3(Z.2^{S-2n} – 1).

Possibility – II: Parity of all d_k => even: As, negative perturbation for each exponent can only be 1, this scenario doesn't allow any negative perturbation or any odd number of positive perturbation. However, by lemma 1B, purely positive perturbations from equilibrium leads to $D_{new} > N_{new}$. Hence, despite the possibility that 3-adic valuation might align with RHS when perturbation is purely positive and even, the cycle condition is violated.

In either case, forbidden permutations of exponents are: a) Purely negative perturbations, b) Purely positive perturbations, c) Any sort of mixed perturbations.

Therefore, the assumption, $\Delta N_{actual} = \Delta N_{required}$ is false**. No perturbation from the equilibrium holds the cycle condition for net decrements in numerator**.

Case B: Due to some perturbations, if $N_{new} < N_{eq}$,

$\Delta N_{required}$ = (2^{2n} - Z. 2^S) + (Z - 1). 3^n and. ΔN_{actual} = $\sum_{k=2}^{n} 3^{n-k}(2^{2k-2} - 2^{S_{k-1}})$

By the same cycle condition, $\sum_{k=2}^{n} 3^{n-k}(2^{2k-2} - 2^{S_{k-1}})$ = (2^{2n} - Z. 2^S) + (Z - 1). 3^n

Or, $\sum_{k=2}^{n} 3^{n-k}(2^{2k-2} - 2^{S_{k-1}})$ – (2^{2n} - Z. 2^S) = (Z - 1). 3^n

Or, $\sum_{k=2}^{n} 3^{n-k}(2^{2k-2} - 2^{S_{k-1}})$ – 2^S. (2^{2n-S} - Z) = (Z - 1). 3^n

v_3(RHS) = n + v_3(Z – 1)

And, v_3(LHS) = minimum[n – k + $v_3\{2^{S_{k-1}}(2^{d'} - 1)$, v_3(2^{2n-S} - Z)] (d' = |2k – 2 – S_{k-1}|)

This case is governed by the similar parity conditions of d' as in Case A.

The above analysis shows, both the Case A and Case B together rules out cycle possibility due to all perturbation scenarios depicted in the model leading to $N_{new} > N_{eq}$ and/or $N_{new} < N_{eq}$.

Therefore, the initial assumption $\Delta N_{required} = \Delta N_{actual}$, is false by contradiction. The integer cycle condition is violated.

(A section 6.1 is added for empirical demonstration)

Lemma 1D:

Statement: I**f due to some mixed perturbations from base case, N_{new} remains = N_{eq}, the cycle ratio R ∉ $\mathbb{Z}^+$**.

Proof: For R ∈ $\mathbb{Z}^+$, D_{new} must be < N_{new}. This is possible if final sum of divisional exponent S = 2n – p i.e. the perturbation has a net decrement effect 'p' on the division exponent sum.

$N_{new} = N_{eq} = 2^{2n} - 3^n$.

$D_{new} = 2^{2n-p} - 3^n$.

Hence, R = $\frac{[2^{2n}-3^n]}{[2^{2n-p}-3^n]} = \frac{[2^{2n-p}-3^n+2^{2n}-2^{2n-p}]}{[2^{2n-p}-3^n]} = 1 + \frac{[2^{2n-p}\{2^p-1\}]}{[2^{2n-p}-3^n]}$

p is much less significant compared to 2n as p < 0.415.n, hence, $2^p - 1 < 2^{2n-p} - 3^n$.

Now we have, $2^p - 1 < 2^{2n-p} - 3^n$, 2^{2n-p} in numerator is an even integer and D_{new} is an odd integer not divisible by 3.

Therefore, R = 1 + fraction => $R \notin \mathbb{Z}^+$. (Proved)

Lemma 1E:

Statement: Altering the last division exponent a_n from base case never yields $R \in \mathbb{Z}^+$.

Proof: Increasing a_n from equilibrium lets only the denominator to grow. The numerator N_{eq} remains same. Clearly, this leads to R < 1. Therefore, we need to examine integrality of R due to lowering a_n. Perturbed value of a_n = 1.

$D_{new} = 2^{2n-1} - 3^n = \frac{1}{2}[2^{2n} - 3^n] + \frac{3^n}{2} - 3^n = \frac{1}{2}D_{eq} - \frac{3^n}{2} = \frac{1}{2}[N_{eq} - 3^n]$

$\Rightarrow R_{new} = \frac{N_{eq}}{D_{new}} = \frac{N_{eq}}{\frac{1}{2}[N_{eq}-3^n]} = \frac{2N_{eq}}{[N_{eq}-3^n]} = \frac{2[N_{eq}-3^n]}{[N_{eq}-3^n]} + \frac{2.3^n}{[N_{eq}-3^n]} = 2 + \frac{2.3^n}{[N_{eq}-3^n]} = 2 + \frac{2.3^n}{2^{2n}-2.3^n}$ = 2 + an irreducible fraction.

$\Rightarrow R \notin \mathbb{Z}^+$.

Any alteration of the last exponent a_n from equilibrium violates the cycle condition. (Proved)

Corollary: Statement: 1 is the only odd integer that can reappear in Collatz sequence.

Recalling equation (ii) and simplifying: $k = -\frac{m}{2^v} + \frac{3^{n-1} + 3^{n-2}.2^{a_1} + 3^{n-3}.2^{a_1+a_2} \;\ldots\ldots\; + 2^{S'}}{2^v.(2^S - 3^n)} = \frac{1}{2^v}[R - m]$

From Lemma 1A, 1B, 1C, 1D and 1E the only solution found for R = 1 is exponents $a_1 = a_2 = a_3 = \ldots.. a_n = 2$

If R = 1 and m = 1, k =0. Therefore, $2^v.k + m$ becomes $2^v.0 + 1 = 1$ which is the only valid modular class or odd integer. Therefore, 1 is the only odd integer can reappear in a Collatz sequence. (Proved)

Corollary: 4-2-1 is the only integer loop in Collatz sequence.

Section 5: Proof of Theorem 2: Non-Existence of Divergent Trajectories

To prove that no divergent trajectories exist, we model the Collatz process as a finite-state transition system over modular class. This allows us to analyze the long-term behavior of any integer by studying the paths it takes through this system.

Section 5.1 The Modular Loop Framework

The following definitions formalize the concepts of modular recurrence that are central to our proof against divergent paths.

Modular Loop: An odd integer is expressed in the form $2^v.k + m$, where m is an odd residue modulo 2^v and k is its core integer. These 2^{v-1} residue classes form the states of a finite-state system. A modular loop occurs when an integer, after a finite number of odd-steps, returns to the same modular residue m but with a different core integer k'. The transformation of k to k' can result in a net increase or decrease in the integer's magnitude.

Modular Path: A modular path is any finite sequence of transformations under the Collatz odd-step function, mapping an integer from an initial modular class m_i to a final modular class m_f. A modular loop is a special case of a modular path where $m_i = m_f$. Modular paths occurred in the Collatz sequence of an odd integer are the mathematical identities those tells us how an integer increases or decreases. Every modular path is associated with a linear Diophantine equation. A detailed catalogue of such equations are attached in the appendix section.

Section 5.2 The Problem of Indefinite Recurrence: A divergent trajectory, if one were to exist, would require an integer to grow without bound. Within our framework, this could only happen if the integer could follow a growth-inducing modular loop or path an infinite number of times.

The following lemmas (2A and 2B) prove that this is impossible. We will show using 2-adic analysis that the congruence conditions required for an integer to remain indefinitely on any modular path leads to a mathematical contradiction.

Lemma 2A:

Statement: No core integer can survive indefinitely in a single growth-inducing modular loop.

Proof: Let us consider a specific n-step ($n \geq 1$) modular loop on integers of the form: $2^v.k_1 + m_1$, recalling the equation (i), after first cycle:

$$2^v.k_n + m_n = \frac{3^n.2^v.k_1 + 3^n.m_1 + 3^{n-1} + 3^{n-2}.2^{a_1} + 3^{n-3}.2^{a_1+a_2} + \dots + 2^{a_1+a_2+\dots+a_{n-1}}}{2^{a_1+a_2+a_3+\dots\dots\dots+a_n}}$$

Writing $m_n = m_1$ to close the loop and simplifying:

$$k_n = \frac{\mathbf{3^n 2^v}.k_1 + \mathbf{(3^n - 2^S)}.\boldsymbol{m} + 3^{n-1} + 3^{n-2}.2^{a_1} + 3^{n-3}.2^{a_1+a_2} + \dots + 2^{a_1+a_2+\dots+a_{n-1}}}{\mathbf{2^v.2^S}}$$

This takes the form of $k_n = \frac{3^n.\mathbf{2^v}.k_1 + z}{2^p}$ .... Equation (v)

Where $z = (3^n - 2^S).m + 3^{n-1} + 3^{n-2}.2^{a_1} + 3^{n-3}.2^{a_1+a_2} + \dots + 2^{a_1+a_2+\dots+a_{n-1}}$

And, the denominator, $2^v.2^S = 2^{s+v}$ is written as 2^p such that $p = S + v$

The equation (v) is a linear Diophantine condition in k_1. It's integer solutions form an arithmetic progression.

After 2nd cycle, k_1 becomes k_n, therefore, substituting k by k_n,

$k_{2n} = \frac{3^n k_n.\mathbf{2^v}.+z}{2^p} = \frac{3^{2n}(\mathbf{2^v})^2.k_1 + \mathbf{3^n}.\mathbf{2^v}.z + \mathbf{2^p}.z}{2^{2.p}}$ .... Diophantine Equation (vi).

Solutions for k_1 represents a subset of integers capable of closing 2nd cycle in the specific modular loop.

Similarly, after r^{th} cycle, $k_{r.n} = \frac{3^{r.n}(\mathbf{2^v})^r.k_1 + \mathbf{3^{(r-1).n}}.(\mathbf{2^v})^{\mathbf{r-1}}.z + \mathbf{3^{(r-2).n}}.(\mathbf{2^v})^{\mathbf{r-2}}.z + \dots\dots\dots \mathbf{2^{(r-1)p}}.z}{2^{r.p}}$

Or, $k_{r.n} = \frac{3^{r.n}(\mathbf{2^v})^r.k + Z'}{2^{r.p}}$; $z' = 3^{r.n-1}.(2^v)^{r-1}.z + 3^{r.n-2}.(\mathbf{2^v})^{r-2}.z + \dots\dots\dots.2^{(r-1)p}.z$

$k_{r.n} = \frac{3^{r.n}(\mathbf{2^v})^r.k_1 + Z'}{2^{r.p}}$ .... Diophantine Equation (vii)

Equation (vii) – a linear Diophantine equation defines an arithmetic progression condition on k_1 imposing $3^{r.n}(2^v)^r.k_1 + Z' \equiv 0 \pmod{2^{rp}}$. It's integer solutions form a subset $A_r = \{k_1 : k_{n.r} \in \mathbb{Z}^+\}$

And, after $(r+1)^{th}$ cycle, $k_{n(r+1)} = \frac{3^{(r+1).n}(\mathbf{2^v})^r.k_1 + \mathbf{3^{r.n}}.(\mathbf{2^v})^r.z + \mathbf{3^{(r-1).n}}.(\mathbf{2^v})^{\mathbf{r-1}}.z + \dots\dots\dots 2^{r.p}.z}{2^{(r+1).p}}$ ...... Diophantine Equation (viii)

$k_{n(r+1)} = \frac{3^{r.n}(\mathbf{2^v})^r.k_1 + z''}{2^{r.p}}$; $z'' = 3^{(r+1).n-1}.(2^v)^r.z + 3^{r.n}.(2^v)^r.z + \dots\dots\dots.2^{r.p}.z$

Likewise, for equation (viii) integer solutions of k_1 form a subset $A_{r+1} = \{k_1 : k_{n(r+1)} \in \mathbb{Z}^+\}$

$A_r = \{k_1 : k_{r.n} \text{ is an integer}\} = \{k_1 : 3^{r.n}(2^v)^r.k_1 + Z_r \text{ is divisible by } 2^{r.p}\}$

Step 2: The Condition for Surviving One Cycle

For the initial integer to survive the first cycle, the resulting core k_n must also be an integer. This requires the numerator of the fraction to be perfectly divisible by the denominator. This imposes a linear congruence condition on the initial core k_1: $\mathbf{3^n.\ 2^v.\ k_1 + z \equiv 0 \pmod{2^p}}$

The set of all integer solutions k_1 to this equation forms an infinite arithmetic progression. We will call this set $\mathbf{A_1}$. Any integer whose core is not in A_1 fails to survive even a single cycle.

Step 3: The Condition for Surviving Two Cycles

For an integer to survive a second cycle, its core after the first cycle, k_n, must itself be a valid starting core for another pass. This means we can substitute the entire expression for k_n back into the equation for k_1. This results in a new equation for the core after two cycles, k_{2n}:

$$k_{2n} = \left[\frac{3^n(2^v)^2 k_1 + z(3^n \cdot 2^v + 2^P)}{2^{2p}}\right]$$

The condition for k_{2n} to be an integer is a new, much stricter congruence on the original core k_1, this time with a modulus of 2^{2p}. The set of all integer solutions k_1 to this second equation forms the set $\mathbf{A_2}$.

Step 4: The General Condition and the Nested Sequence

We can generalize this process. For an initial core k_1 to survive r full cycles, it must satisfy a linear congruence with a modulus of 2^{rp}. The set of solutions for r cycles is $\mathbf{A_r}$.

The modulus for the $(r+1)^{th}$ cycle $\{2^{(r+1)p}\}$ is a multiple of the modulus for the r-th cycle (2^{rp}). This has a crucial consequence: any integer that satisfies the condition for r+1 cycles must also satisfy the condition for r cycles.

This means the solution sets form an infinite **nested sequence of proper subsets**: $A_1 \supset A_2 \supset A_3 \supset \ldots$

Each set is strictly smaller than the one before it because the congruence condition becomes more restrictive at each step. An integer that is a solution mod 2^{rp} is not guaranteed to be a solution mod $2^{(r+1)p}$.

Step 5: The Contradiction

For an integer to persist indefinitely—and thus form a divergent path—its core k_1 must be a member of the **infinite intersection** of all these sets, $\cap A_r$. We will now prove that this intersection is empty for any rational integer.

The proof rests on a fundamental property of integers. For any given rational integer k_1, its 2-adic valuation, $v_2(k_1)$, is a fixed, finite number. This represents the highest power of 2 that divides k_1.

However, the congruence condition for surviving r cycles, ... $\equiv 0 \pmod{2^{rp}}$, requires that the expression involving k_1 satisfy a divisibility property by a power of 2 that **grows with r**.

As r increases, the required 2-adic valuation of the expression will eventually surpass the fixed, finite 2-adic valuation of k_1 itself. This means that for any given integer k_1, there must exist a finite number of cycles, r, for which it fails to satisfy the necessary congruence condition for the next pass.

Therefore, for any given k_1, there must exist an integer r such that $k_1 \notin A_{r+1}$. This means no integer can belong to all sets in the infinite sequence. The infinite intersection is provably empty:

The analysis has clearly established that for an integer to persist through r cycles of a given n-step modular loop, its initial core integer, k_1, must belong to the solution set A_r. These sets form an infinite nested sequence of arithmetic progressions: $A_1 \supseteq A_2 \supseteq A_3 \supseteq \ldots$

For an integer to persist indefinitely—and thus form a divergent path—its core k_1 must be a member of the infinite intersection of all these sets, $\bigcap_{r=1}^{\infty} A_r$. We will now prove that this intersection is empty.

Writing the congruence after r cycles as $A_r . k_1 + B_r \equiv 0 \pmod{2^{m_r}}$, we factor $A_r = 2^{t_r}.a'_r$ with odd a'_r. The recursive definition of the coefficient is of the form $A_{r+1} = 3^n . A_r + T$, where T is a fixed term determined by a single pass through the loop. Since 3^n is odd, multiplication by this factor does not change the 2-adic valuation. Any increase in $v_2(A_{r+1})$ can only come from the addition of T, which can at most add a fixed, bounded number of powers of 2. Hence, there is a uniform constant $C \geq 0$ with $t_{r+1} \leq t_r + C$ for all r. On the other hand, the modulus exponent satisfies $m_{r+1} = m_r + p$ with $p \geq 1$ (the per-cycle increment). Since $p > C$ in the model, we obtain $m_r - t_r \to \infty$ as $r \to \infty$. Therefore, for any fixed k_1 with $v_2(k_1) = v$, there exists an R such that for all $r \geq R$, the congruence reduces (after cancelling the common factor 2^{t_r}) to a non-trivial congruence modulo $2^{m_r - t_r}$ on the odd part of k_1, which forces a genuine increasing congruence condition.

The preceding analysis establishes that for any given integer k_1 with a fixed, finite 2-adic valuation, the effective congruence condition it must satisfy becomes progressively stricter. Eventually, the required 2-adic divisibility will surpass the fixed valuation of k_1, meaning there must exist a finite number of cycles, r, for which it fails to satisfy the necessary congruence for the next pass.

Therefore, for any given k_1, there must exist an integer r such that $k_1 \notin A_{r+1}$. This means no integer can belong to all sets in the infinite sequence, and the infinite intersection is empty: $\bigcap_{r=1}^{\infty} A_r = \emptyset$.

No integer can survive in a single growth inducing modular loop. (Proved)

Corollary V: Each A_r is an infinite arithmetic progression modulo $2^{r\,p}$. However, any fixed integer k_1 can belong to only finitely many of the nested sets $A_r, A_{r+1}, \ldots$. Consequently, no modular cycles forever; each core integer eventually

yields a nonzero remainder and the loop terminates.

Lemma 2B:
Statement: No modular path can have infinite recurrences.
Perceived possibility of wondering path: Every modular path is characterized by a Diophantine equation. Let us consider an integer $x_1 = 2^v.k_1 + m_1$, after n odd steps transforms into $x_n = 2^v.k_n + m_n$, the Diophantine equation presenting this transformation: $x_n = 2^v.k_n + m_n = \frac{3^n.2^v.k_1+3^n.m_1 + 3^{n-1} + 3^{n-2}.2^{a_1}+3^{n-3}.2^{a_1+a_2}+\ldots+2^{a_1+a_2+\cdots+a_{n-1}}}{2^{a_1+a_2+a_3+\cdots\ldots\ldots+a_n}}$ $\sum a_i = S$ (i = 1 to n)

$k_n = \frac{\mathbf{3^n 2^v}.k_1 + (3^n.m_1 - \mathbf{2^S}.m_n) + 3^{n-1} + 3^{n-2}.2^{a_1}+3^{n-3}.2^{a_1+a_2}+\ldots+2^{a_1+a_2+\cdots+a_{n-1}}}{\mathbf{2^v.2^S}}$ This equation can take form of:

$k_n = \frac{3^n.\mathbf{2^v}.k_1 + X}{2^p}$ …………….. Equation (ix) similar to the equation (vii) in lemma 2A solution of which similarly represents an arithmetic progression of integer sets.
lemma 2A has proven the impossibility of indefinite recurrence of modular loop path showing eventual ejection due to eventual surpassing of 2-adic valuation required to stay in the path indefinitely.
Therefore, no path can recur indefinitely.
Corollary: Wondering path or everlasting growth of any integer is impossible in Collatz sequence.

Section 6. Empirical Validation and Demonstrations with Modulo 16 and Modulo 32:

Methodology: This section uses modulo residual classes of 16 as a tool for classification aiming to explore disciplined structures in Collatz sequence. Lower moduli (mod 2, 4 and 8) with lesser number of residual classes are too simple to capture sufficient granules in understanding the odd-to-odd transition paths of integer or modular cycles. Modulo 16 and modulo 32 provide much clearer scenario while higher moduli become progressively complex and infeasible to compute. First, we introduce a brute-force technique that examines all possible permutations of exponents in the Diophantine cycle equation. We have used Python codes for this exhaustive checking.

Number of Cases to be Examined: By lemma 0, we can predict the exact statistics of exponents we'll encounter.
For modulo 16: Total odd classes = 8, total number of exponents = 8.
Total number of ($a_i = 1$) exponents = 4, total number of ($a_i = 2$) exponents = 2, Total number of ($a_i = 3$) exponents = 1 and total number of ($a_i \geq 4$) = 1.
Hence, total possible permutations $P_{16} = \frac{16!}{(4!)\times(2!)\times(1!)\times(1!)} = 840$

For modulo 32: Total odd classes = 16, total number of exponents = 16.
Total number of ($a_i = 1$) exponents = 8, ($a_i = 2$) exponents = 4, ($a_i = 3$) exponents = 2, ($a_i = 4$) exponents = 1 and ($a_i \geq 5$) = 1.
Hence, total possible permutations $P_{32} = \frac{32!}{(8!)\times(4!)\times(2!)\times(1!)\times(1!)} = 10810800$

Similarly, for modulo 64, P_{64} = 6,488,213,333,318,400 (≈ 6.49 quadrillion)

Feasibility of Brute Force Checking: A standard desktop computer processor checks millions of permutations in a few minutes, however, if a computer checks with an incredible speed of one million cases per second, it will take 205.6 years to complete 6.49 quadrillion cases. Therefore, we have examined all cases of modulo 16 and modulo 32 only exhaustively using Python code.

Procedure: We used the Diophantine equation: $R = \frac{\mathbf{3^{n-1}} + 3^{n-2}.2^{a_1}+3^{n-3}.2^{a_1+a_2} \ldots\ldots\ldots + \mathbf{2^{S'}}}{(\mathbf{2^S} - \mathbf{3^n})}$ to check all possible permutations of the exponents using base value of highest exponent and found R to be non-integer in each and every case. By lemma 0, any one exponent can be arbitrarily large. Therefore, the code then checked all R > 1 cases separately by incrementing the highest exponent progressively unless R was brought down to < 1. No integer value of R found during this transition from R > 1 to R <1. This presents direct computational evidence in support of the theoretical finding that no non-trivial integer cycle exists. The Python codes are attached in appendix E and F for verification and

reproducibility.
Result: The 'no exception' result provides an exclusive empirical validation for lemma 1A to 1E that no non-trivial integer cycle exists.

Section 6.1: Demonstrative Example for Lemma 1C (Mixed Perturbation)

This example will show how the 3-adic valuation mismatch, which is the core of Lemma 1C's proof, holds even for a complex set of perturbations.

1. Setup and Baseline (Equilibrium State)

First, we establish the baseline 'equilibrium' values for a hypothetical cycle of length n = 10.

- **Cycle Length:** n = 10
- **Equilibrium Exponents:** {2, 2, 2, 2, 2, 2, 2, 2, 2, 2}
- **Equilibrium Sum (S_{eq}):** 2 x 10 = 20
- **Equilibrium Numerator (N_{eq}):** $2^{2n} - 3^n = 2^{20} – 3^{10}$ = 1048576 – 59049 = 989527
    - $\Delta N_{required} = (Z.2^S – 2^{2n}) + (1 – Z).\ 3^n$
    - $\Delta N_{required} = (Z.\ 2^{21} – 2^{20}) + (1 – Z).\ 3^{10}$

4. The 3-adic Valuation Proof: This is the core of the argument in Lemma 1C. We test if the fundamental cycle equation can hold by comparing the 3-adic valuations (v_3) of both sides.

- **The Core Equation:** $\Delta N_{actual} - (Z.\ 2^S – 2^{2n}) = (1 - Z).\ 3^n$
  Right-Hand Side (RHS):
    - **v_3(RHS) = v_3[(1 - 5). 3^{10}]** = 10 + v_3(1 - Z)
    - **Left-Hand Side (LHS):** $\Delta N_{actual} – (Z.\ 2^S – 2^{2n})$

General form of N
$= 3^{n-1} + 3^{n-2}.2^{a_1} + 3^{n-3}.2^{(a_1+a_2)} + 3^{n-4}.2^{(a_1+a_2+a_3)} + 3^{n-5}.2^{(a_1+a_2+a_3+a_4)} + 3^{n-6}.2^{(a_1+a_2+a_3+a_4+a_5)} + 3^{n-7}.2^{(a_1+a_2+a_3+a_4+a_5+a_6)} + 3^{n-8}.2^{(a_1+a_2+a_3+a_4+a_5+a_6+a_7)} + \cdots \ldots \ldots \ldots + 2^{(a_1+a_2+\cdots+a_{n-1})}$

At equilibrium, in this case, $N_{eq} = 3^9 + 3^8.2^2 + 3^7.2^4 + 3^6.2^6 + 3^5.2^8 + 3^4.2^{10} + 3^3.2^{12} + 3^2.2^{14} + 3.2^{16} + 2^{18}$
Or, $N_{eq} = T_1 + T_2 + T_3 + \ldots\ldots + T_{10}$.
As first perturbation introduced at a_2, first perturbed term is the 3rd term in N_{new} :
$T_3 = 3^7.\ 2^{(2+6)} = 3^7.\ 2^8$. (Perturbed a_2 = 2 + 4 = 6)
$T_4 = 3^6.\ 2^{(8+2)} = 3^6.\ 2^{10}$. (Unperturbed a_3 = 2)
$T_5 = 3^5.\ 2^{(10+2-1)} = 3^5.\ 2^{11}$. (Perturbed a_4 = 2 – 1 = 1)
$T_6 = 3^4.\ 2^{(11+2-1)} = 3^4.\ 2^{12}$. (Perturbed a_5 = 2 – 1 = 1)
$T_7 = 3^3.\ 2^{(12+2+2)} = 3^3.\ 2^{16}$. (Perturbed a_6 = 2 + 2 = 4)
$T_8 = 3^2.\ 2^{(16+2-1)} = 3^2.\ 2^{17}$. (Perturbed a_7 = 2 – 1 = 1)
$T_9 = 3.\ 2^{(17+2-1)} = 3.\ 2^{18}$. (Perturbed a_8 = 2 – 1 = 1)
$T_{10} = 3^0.\ 2^{(18+2-1)} = 2^{19}$. (Perturbed a_9 = 2 – 1 = 1)
All of the terms in N_{new} are greater than those of N_{eq},. Hence, $N_{new} > N_{eq}$.
$\Delta N_{actual} = N_{new} – N_{eq}$
$= 3^7.\ [2^8 – 2^4] + 3^6.\ [2^{10} – 2^6] + 3^5.\ [2^{11} – 2^8] + 3^4.\ [2^{12} – 2^{10}] + 3^3.\ [2^{16} – 2^{12}] + 3^2.\ [2^{17} – 2^{14}] + 3.\ [2^{18} – 2^{16}] + [2^{21} – 2^{18}]$
$= 3^7.\ 2^4[2^4 – 1] + 3^6.\ 2^6[2^4 – 1] + 3^5.\ 2^8[2^3 – 1] + 3^4.\ 2^{10}.\ [2^2 – 1] + 3^3.\ 2^{12}[2^4 – 1] + 3^2.\ 2^{14}[2^3 – 1] + 3.\ 2^{16}[2^2 – 1] + 2^{18}[2^3 – 1]$
New Exponent Vector: {2, 6, 2, 1, 1, 4, 1, 1, 1, 2}

$N_{new} = 6395648$ and $D_{new} = 2^{21} - 3^{10} = 2038103$, resulting in $N_{new} > D_{new}$.
3-adic valuation of each term in ΔN_{actual}:

Terms	v_3 formula	v_3
$3^7 . 2^4[2^4 - 1]$	$v_3\{ 3^7 . 2^4(2^4 - 1) \}$	$7 + v_3(2^4 - 1) = 8$
$3^6 . 2^6(2^4 - 1)$	$v_3\{ 3^6 . 2^6(2^4 - 1)\}$	$6 + v_3(2^4 - 1) = 7$
$3^5 . 2^8 (2^3 - 1)$	$v_3\{3^5 . 2^8(2^3 - 1) \}$	$5 + v_3(2^3 - 1) = 5$
$3^4 . 2^{10} (2^2 - 1)$	$v_3\{ 3^4 . 2^{10} (2^2 - 1)\}$	$4 + v_3(2^2 - 1) = 5$
$3^3 . 2^{12}(2^4 - 1)$	$v_3\{ 3^3 . 2^{12}(2^4 - 1)\}$	$3 + v_3(2^4 - 1) = 4$
$3^2 . 2^{14} (2^3 - 1)$	$v_3\{ 3^2 . 2^{14} (2^3 - 1)\}$	$2 + v_3(2^3 - 1) = 2$
$3 . 2^{16}(2^2 - 1)$	$v_3\{ 3 . 2^{16}(2^2 - 1)\}$	$1 + v_3(2^2 - 1) = 2$
$2^{18}(2^3 - 1)$	$v_3\{ 2^{18}(2^3 - 1)\}$	$v_3(2^3 - 1) = 0$

It is evident that the term where the first negative perturbed has been introduced caused lesser 3-adic value than $(1 - Z). 3^{10}$.

- **The Contradiction:**
    - $v_3(\text{RHS}) = 10$
    - $v_3(\text{LHS}) < 10$ guaranteed for each term where x is odd in $v_3(2^x - 1)$ i.e. at least in the first negatively perturbed term. This proves the dichotomy that at least one negative perturbation is enough to yield the mismatch, otherwise all terms must contain even x i.e. all terms are positively perturbed which is forbidden by lemma 1B: purely positive perturbation leads to $D_{new} > N_{new}$. This contradiction is inevitable irrespective of 3-adic valuation of $(Z.2^S - 2^{2n})$.

This demonstrates the **fundamental mismatch** proven in Lemma 1C. The specific combination of perturbations creates an actual change in the numerator that is structurally incompatible with the change required to form an integer cycle, as revealed by their different powers of 3. [1]

Section 6.2. In this section, we will validate the modular loop concept and their instability proved in lemma 2A. For this integers are to be classified and transformation rules to be framed.
Lemma 3: **Mod 16 Residual Classification**:
Statement: All odd positive integers can be expressed in the form of 16k +m where m∈{1, 3, 5, 7, 9, 11, 13, 15} and all even positive integers can be expressed as 16k + m´, where m´ ∈{0, 2, 4, 6, 8, 10, 12, 14}.

Proof: When an odd positive integer N_0 is divided by 16, formally, $N_0 \equiv m \pmod{16}$, there can be remainder ‘m’ such that m = 1, 3, 5, 7, 9, 11, 13 and 15.
When an even positive integer N_0' is divided by 16, formally, $N_0' \equiv m' \pmod{16}$ there can be only eight values of remainder m´ such that m´ ∈{0, 2, 4, 6, 8, 10, 12, 14} (Core integer ‘k’ ∈ $\mathbb{Z}^+$).

Table 1: Odd integers:

16k+m; m =	1	3	5	7	9	11	13	15
Defined as ‘Type’	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8

Table 2: Even Integers

16k+ m´; m´ =	0	2	4	6	8	10	12	14
Defined as ‘Ev’	Ev 1	Ev 2	Ev 3	Ev 4	Ev 5	Ev 6	Ev 7	Ev 8

Lemma 4:
Statement: Maximum Divisibility $d = 2^{v2(48k + m +1)}$: Odd integers belonging to each defined odd type on 3x+1 operation transforms into the next odd integers when divided by $d = 2^n$ ($n \in \mathbb{Z}^+$)

Proof:

Type 1: 16k + 1 → 48k + 4 (by 3x +1 operation).

48k + 4 → 12k +1 is an odd integer for any value of k. This gives n = 2 i.e. divisibility 'd' = 2^n, n= v_2(48k+4) = 2

Similarly,

Type 2: 16k +3 → 48k+10 => Gives n = v_2(48k+10) i.e. divisibility 'd' = 2

Type 3: 16k + 5 → 48k + 16 => Gives n = v_2(48k+16) = 4 + v_2(3k+1), i.e. divisibility d ≥ 2^4

Type 4: 16k + 7 → 48k + 22 => Gives n = v_2(48k+22) i.e. divisibility 'd' = 2

Type 5: 16k + 9 → 48k + 28 => Gives n = v_2(48k+28) i.e. divisibility 'd' = 2^2

Type 6: 16k + 11 → 48k + 34 => Gives n = v_2(48k+34) i.e. divisibility 'd' = 2

Type 7: 16k + 13 → 48k + 40 => Gives n = v_2(48k+40) i.e. divisibility 'd' = 2^3

Type 8: 16k + 15 → 48k + 46 => Gives n = v_2(48k+46) i.e. divisibility 'd' = 2

Table 3: Divisibility Factors (d) for Each Odd Type Under the 3x+1 Operation:

Types	1	2	3	4	5	6	7	8
Divisibility(d)	4	2	2^n (n≥ 4)	2	4	2	8	2

Lemma 5:

Statement: Integers transformation rules: On 3x + 1 operation followed by division by d (= 2^n)

1) Type 1 transforms into Types 1, 3, 5, or 7.
2) Type 2 transforms into Types 3 or 7.
3) Type 3 transforms into any odd type.
4) Type 4 transforms into Type 2 or 6.
5) Type 5 transforms into Type 2, 4, 6 or 8.
6) Type 6 transforms into Type 1or 5.
7) Type 7 transforms into any odd types.
8) Type 8 transforms into Type 4 or Type 8 further.

Proof: Type1 transformation:

Step I: 16k +1 → 48k + 4 (by 3x + 1 operation) → 12k + 1 (division by 4)

Step II: If the core integer 'k' belongs to Type1, substituting 'k' by 16x + 1 ($x \in \mathbb{Z}^+$):

12(16x +1) +1 = 192x + 13 = 16(12x) +13 which is a type7 integer. Therefore, Type 1 integers transform into Type 7.

Likewise, if k belongs to Type 2, substituting 'k' by 16x+3:

12(16x + 3) + 1 = 192x + 37 = 16(12x + 2) + 5 => represents a Type 3 integer. Therefore, Type1 integers transform into Type 3 also.

The core integer 'k' when substituted by all even classes (Ev1 to Ev8) and by all odd classes (Type1 to Type 8), summarized results given in the following table:

Table 4: Transformation summary of Type 1:

'k' Substituted by	Ev 1,3,5,7	Type 2, 4,6,8	Ev 2,4,6,8	Type 1,3,5,7
Transforms into	Type 1	Type 3	Type 5	Type 7

With similar treatment on all the rest odd types, the following results are obtained:

Table 5: Transformation rules of Type2:

Type 2	**Type 3**	**Type 7**
Type of core k=>	**EV 1, 2, 3, 4, 5, 6, 7, 8 (all even integers)**	**Type 1, 2, 3, 4, 5, 6, 7, 8 (all odd integers)**

Illustrative Examples: Type2 integer = 16k + 3 → 48k + 10 (by 3x +1) → 24k + 5 (division by 2)

If k = 2n (all even integers), 24k + 5 = 48n + 5 = 16x3n + 5 => a Type 3 integer.

If k = 2n + 1 (all odd integers), 24k + 5 = 48n + 29 = 16(3n+1) +13 => Type7 integer.

Table 6: Transformation rules of Type 3:

Type 3 to	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8
For core integer, k	Ev 1: 16m + 0	Ev 4: 16m + 6	Ev 3: 16m + 4	Ev 2: 16m +2	Ev 5: 16m + 8	Ev 8: 16m+ 14	Ev 3: 16m + 4	Ev 6: 16m + 10
For core integer, k	Type 1: 16m +1 m =4n type even integers	Type 4: 16m +3 m = 1+2n type odd integers	Type 1: 16m +1 m =4n +3 type odd integers	Type 5: 16m + 9 m = 4n type even integers	Type 1: 16m +1 m =4n +2 type even integers	Type 4: 16m +3 m = 2n type even integers	Type 1: 16m +1 m =4n +1 type odd integers	Type 8: 16m+15 m = 2n +1 type odd integers
For core integer, k	Type 3: 16m + 5 m = 16n type even integers	Type 3: 16m + 5 m = 16n+6 type even integers	Type 2: 16m + 3 m = 2n type even integers	Type 3: 16m + 5 m = 16n + 2 type even integers	Type 3: 16m + 5 m = 16n + 8 type even integers	Type 3: 16m + 5 m = 16n + 14 type even integers	Type 2: 16m + 3 m = 2n +1 type odd integers	Type 3: 16m + 5 m = 16n + 10 type even integers
For core integer, k	Type 6: 16m + 11 m = 2n type even integers	Type 5: 16m + 9 m = 4n +1 type odd integers	Type 3: 16m + 5 m = 16n + 12 type even integers	Type 8: 16m+15 m = 2n type even integers	Type 6: 16m + 11 m = 2n+1 type odd integers	Type 5: 16m + 9 m = 4n +3 type odd integers	Type 3: 16m + 5 m = 16n + 4 type even integers	Type 5: 16m + 9 m = 4n +2 type even integers
For core integer, k	Type 7: 16m + 13 m = 8n +2 type even integers	Type 7: 16m + 13 m = 8n +5 type odd integers	Type 7: 16m + 13 m = 8n type even integers	Type 7: 16m + 13 m = 8n +3 type odd integers	Type 7: 16m + 13 m = 8n +6 type even integers	Type 7: 16m + 13 m = 8n +1 type odd integers	Type 7: 16m + 13 m = 8n + 4 type even integers	Type 7: 16m + 13 m = 8n +7 type odd integers

Illustrative Examples:
Type 3 integer = 16k + 5 → 48k +16 (by 3x+1) → 3k +1 (division by 16)
If k = Ev4 = 16m + 6, 3k +1 = 48x + 19 = 16(3x +1) + 3 => a Type 2 integer.
If k = Ev5 = 16m + 8, 3k +1 = 48m + 25= 16(3m +1) + 9 => a Type 5 integer
If k = Type 1 = 16m + 1, and m = 4n +1, 3k + 1 = 3(64n +17) + 1 = 192n + 52 = 48n + 13 = 16x3n + 13 => a Type 7 integer.
If k = Type 8 = 16m + 15 and m = 2n, 3k +1 = 3(32m + 15)+1 = 96n + 46 = 48n + 23 = 16x(3n+1) + 7 => a Type 4 integer.

Table 7: Transformation rules of Type 4:

Type 4	Type 2	Type 6
Core integer, k =	**Type 1, 2, 3, 4, 5, 6, 7, 8 (all odd integers)**	**Ev 1, 2, 3, 4, 5, 6, 7, 8 (all even integers)**

Illustrative Examples: Type 4 integer = 16k + 7→ 48k + 22 (by 3x + 1) → 24k + 11
If k = 2n +1 (odd integers), 24k + 11 = 48n + 35 = 16 x(3n + 2) + 3 => Type 2 integer.
If k = 2n (even integers), 24k + 11 = 48n + 11 = 16x3n + 11 => Type 6 integer.

Table 8: Transformation rules of Type 5:

Type 5	Type 2	Type 4	Type 6	Type 8

Core integer, k =	Type 1, 3, 5. 7	EV 1, 3, 5, 7	Type 2, 4, 6, 8	EV 2, 4, 6, 8

Illustrative Examples: Type 5 integer =16k + 9 → 48k + 28 (by 3x +1) → 12k + 7 (division by 4)
If k = Type 3 = 16m + 5, 12k + 7 = 12(16m+ 5) + 7 = 192m + 67 = 16 (12m + 4) + 3 => Type 2 integer.
If k = Ev 2 = 16m + 2, 12k + 7 = 12(16m + 2) +7 = 192m + 31 = 16(12m+1) + 15 => Type 8 integer.

Table 9: Transformation rules of Type 6:

Type 6	Type 1	Type 5
Core integer, k =	Ev 1, 2, 3, 4, 5, 6, 7, 8 (even) (all even integers)	Type 1, 2, 3, 4, 5, 6, 7, 8 (odd) (all odd integers)

Illustrative Examples: Type 6 integer = 16k + 11 → 48k + 34 (by 3x +1) → 24k + 17 (division by 2)
If k = 2n (even), 24k + 17 = 48k + 17 = 16(3n+1) +1 => Type 1 integer.
If k = 2n +1 (odd), 24k + 17 = 48k + 41 = 16(3n +2) + 9 => Type 5 integer.

Table 10: Transformation rules of Type 7:

Type 7	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8
k =	Ev 2, 6	Type 3, 7	Ev 1, 5	Type 2, 6	Ev 4, 8	Type 1, 5	Ev 3, 7	Type 4, 8

Illustrative Examples: Type 7 integer = 16k + 13 → 48k + 40 (by 3x + 1) → 6k + 5 (division by 8)
If k = Type 2 = 16m + 3, 6k + 5 = 96m + 23 = 16x (6m + 1) + 7 => Type 4 integer.
If k = Ev 7 = 16m + 12, 6k + 5 = 96m + 77 = 16(6m + 4) + 13 => Type 7 integer.
If k = Type 5 = 16m + 9, 6k + 5 = 96m + 59 = 16(6m + 3) + 11 => Type 6 integer.

Table 11: Transformation rules for Type 8:

Type 8	Type 4	Type 8
Core integer, k =	Ev 1, 2, 3, 4, 5, 6, 7, 8 (even) (all even integers)	Type 1, 2, 3, 4, 5, 6, 7, 8 (odd) (all odd integer)

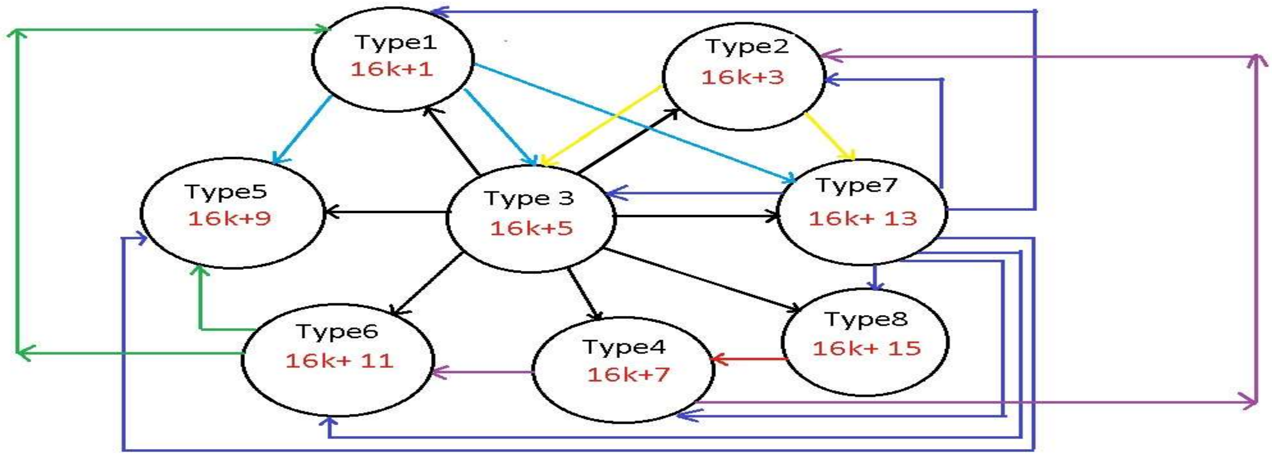
Illustrative Examples: Type 8 integer = 16k + 15 → 48k + 46 (by 3x + 1) → 24k + 23
If k = 2n (even), 24k + 23 = 48n + 23 = 16(3n +1) + 7 => Type 4 integer.
If k = 2n + 1 (odd), 24k + 23 = 48n + 47 = 16(3n +2) + 15 => Type 8 integer.

Table 12: Transformation Rules Summary of All Odd Integers:

Types	Transforms into	Forbidden Transformations	Maximum divisor (d)	Growth Tendency
Type 1	Type 1, 5 (for even core) Type 3, 7 (for odd core)	Type 2, Type 4, Type 6 and Type 8	4	Decreasing
Type 2	Type 3 (for even core), Type 7	Type 1, Type 2, Type 4, Type 5, Type 6 and Type 8	2	Increasing
Type 3	All	None	16, 32 or more	Decreasing
Type 4	Type 2 (odd), Type 6	Type 1, Type 3, Type 4, Type 5, Type 7 and Type 8	2	Increasing

Type 5	Type 2, Type 4, Type 6 and Type 8	Type 1, Type 3, and Type 5 and Type 7	4	Decreasing
Type 6	Type 5 and Type 1	Type 2, Type 3, Type 4, Type 6, Type7 and Type 8	2	Increasing
Type7	All	None	8	Decreasing
Type 8	Type 4 and Type 8	Type 1, Type 2, Type 3, Type 5, Type 6, Type 7	2	Increasing

Visual of Transformation Rules:



Type 3 and 7 are the most connected integers and Type 8 are the least connected integers.

Section 8. Forbidden Transformations: Some transformations, like Type 1 to Type 2, Type 4 to Type 8 or Type 8 to Type 6 etc. are mathematically impossible. These are 'forbidden transformations.' Following table summarizes all forbidden transformations:

Table 13: Forbidden Transformations:

Integer	**Type 1**	**Type 2**	**Type 3**	**Type 4**	**Type 5**	**Type 6**	**Type 7**	**Type 8**
Type 1		**Forbidden**		**Forbidden**		**Forbidden**		**Forbidden**
Type 2	**Forbidden**	**Forbidden**		**Forbidden**	**Forbidden**	**Forbidden**		**Forbidden**
Type 3								
Type 4	**Forbidden**		**Forbidden**	**Forbidden**	**Forbidden**		**Forbidden**	**Forbidden**
Type 5	**Forbidden**		**Forbidden**		**Forbidden**		**Forbidden**	
Type 6		**Forbidden**	**Forbidden**	**Forbidden**		**Forbidden**	**Forbidden**	**Forbidden**
Type 7								
Type 8	**Forbidden**	**Forbidden**	**Forbidden**		**Forbidden**	**Forbidden**	**Forbidden**	

Examples of Valid Transformation Sequences:

Example 1: **Type2** to **Type3** to **Type5** to **Type6** to **Type1** (non-recursive).

Example 2: **Type 8** to **Type 4** to **Type 2** to **Type 7** to **Type 6** to **Type 5** (non-recursive).

Example 3: **Type 6** to **Type 1** to **Type 3** to **Type 5** to **Type 6** (recursive: **Type 6** to **Type 6**).

Example 4: **Type 4** to **Type 2** to **Type 7** to **Type 8** to **Type 4**(recursive: **Type 4** to **Type 4**).

Section 7. Demonstration of Modular loops:

Lemma 7:

Statement: Growth tendency of looping and non-looping Collatz sequences is determined by comparing accumulation of power of 3 in numerator with accumulation of power of 2 in denominator.

Proof: an odd integer $2^v.k_1 + m$, after n transformations forms odd integer =

$2^v.k_1 + m' = \frac{3^n 2^v.k1+3^n.m + 3^{n-1}+ 3^{n-2}.2^{a_1}+3^{n-3}.2^{a_1+a_2}+\dots+2^{a_1+a_2+\dots+a_{n-1}}}{2^{a_1+a_2+a_3+\dots\dots\dots+a_n}} = \frac{3^n.x+T}{2^p}$

Every such sequence is attributed by a linear Diophantine equation with integer solutions of k. For any sequence length, $n \geq 2$ and hence $2^v.k_1 + m$ cannot be insignificant. For large values of x, T is negligible. Therefore, if $3^n > 2^p$ then the sequence is increasing and if $3^n < 2^p$, it is decreasing.

Empirical Evidence: Tracking of Transformation Paths: In this finite state transition system with deterministic transformation rules (lemma 7), using a DFS (python) code, **911 looping sequences** are exhaustively tracked. The DFS code is attached in the appendix section. Out of the **911 looping** sequences, it is found only **49** looping sequences are of increasing or diverging growth tendency, by lemma 7.

Demonstrative Examples:

A) Looping sequence: **Type 2** to **Type 7** to **Type 8** to **Type 4** to **Type 2**:

Step I: Starting Type 2 = 16k + 3 → 24k + 5 (3x + 1, followed by division by 2).

Step II: 24k + 5 (Type 7, according to sequence) → 9k + 2 (3x + 1, followed by division by 8).

Step III: 9k + 2 (Type 8) → $\frac{27k+7}{2}$ (3x + 1, followed by division by 2).

Step IV: $\frac{27k+7}{2}$ (Type 4) → $\frac{81k+23}{4}$ (3x + 1, followed by division by 2)

The filial modulo class of the loop = Type 2 (say, 16k' +3) = $\frac{81k+23}{4}$.

$k' = \frac{81k+11}{64}$ => This linear Diophantine equation has unique integer solution set for k = 37 + 64.n yielding k' = 47 + 81.n

The integer 16k +3 survives for a single cycle if initiated by the core integer k = 37 + 64.n ($n \in \mathbb{Z}^+$)

Growth tendency = (numerator's power of 3) / (denominator's power of 2) = $3^4 / 2^6 > 1$ = increasing.

Section 8. Samples of Looping Sequences and Validation of Lemma 2A:

Setting up Diophantine equation:

Case 1: **Type 4** to **type 2** to **type 7** to **type 8** to **type 4:** Let the parent integer be 16k+7.

1st Cycle: 16k+7 → 24k+ 11 (by 3x+1 operation, followed by division by 2 as divisibility of type4 is 2)

24k+11 → 36k+17 (by 3x+1 operation, followed by division by 2 as divisibility of type2 is 2).

36k+17 → $\frac{27k+13}{2}$ (by 3x+1 operation, followed by division by 2 as divisibility of type7 is 8).

$\frac{27k+13}{2}$ → $\frac{81k+41}{4}$ (by 3x+1 operation, followed by division by 2 as divisibility of type8 is 2).

2nd Cycle: $\frac{81k+41}{4} \rightarrow \frac{243k+127}{8} \rightarrow \frac{729k+389}{16} \rightarrow \frac{2187k+1183}{128} \rightarrow \frac{6561k+3677}{256}$.

3rd Cycle: $\frac{6561k+3677}{256} \rightarrow \frac{19683k+11287}{512} \rightarrow \frac{59049k+34373}{1024} \rightarrow \frac{177147k+10414}{8192} \rightarrow \frac{531441k+32062}{16384}$.

The last term of the loop represents filial Type 4 integer and may be represented as 16m +7

Therefore, $16k'+7 = \frac{531441k+320}{16384}$

$k' = \frac{531441k+20599}{262144}$ This Diophantine equation has integer solutions of k = 246723 + 262144.n which yields k' = 500179 + 531441.n ($n \in \mathbb{Z}^+$)

The parent integer should be $16k+7 = 3947575 + 2^{22}.n$

The calculations show that the said looping sequence initiated by **3947575** $+2^{22}.n$ type of integer will continue for 3 full cycles and will reach to 16m+7 = **8002871** + 8503056.n. Let us verify with n = 0 :

1st Cycle: 3947575(type4) → 5921363(type2) → 8882045(type7) → 3330767(type8) → 4996151(type4) → 7494227(type2) → 11241341(type7) → 4215503(type8) → 6323255(type4)

2nd Cycle: 6323255(type4)→ 9484883(type2) → 14227325(type7) → 5335247(type8) → **8002871(= 16 x 500179 +7 => type4)**

And then, the 3rd cycle: 8002871(type4) → 12004307 (type2) → 18006461(type7) → 6752423 (type4)

So, the loop terminates after 3rd cycle due to residual mismatch.

Case 2: **Type 2** – **type 7** – **type 8** – **type 4** – **type 6** – **type 1** – **type 5** – **type 2**:

Parent integer = 16k+3

1st Cycle: 16k+3 → 24k+5 → 9k+2 → $\frac{27k+7}{2}$ → $\frac{81k+2}{4}$ → $\frac{243k+}{8}$ → $\frac{729k+227}{32}$ → $\frac{2187k+713}{128}$

2nd Cycle: $\frac{2187k+713}{128}$ → $\frac{6561k+2267}{256}$ → $\frac{19683k+705}{2048}$ → $\frac{59049k+23219}{4096}$ → $\frac{177147k+7375}{8192}$ →

$\frac{531441k+2}{16384}$ → $\frac{1594323k+70}{65536}$ → $\frac{4782969k+21797}{262144}$

After 2nd cycle, the filial integer, say, 16k'+3 = $\frac{4782969k+2179747}{262144}$ => k' = $\frac{4782969k+1393315}{262144x\ 16}$

The Diophantine equation is satisfied by k = 232346 + 2^{22}.n yielding m = 2649556 + 3^{14}.n (n Є $\mathbb{Z}^+$)

Therefore, this loop with two cycles is initiated by the parent integer = 16k+7 = 37175379 + 2^{26}.n and terminates with the filial integer is 16k'+7 = 42392899 + 76527504.n due to modular mismatch.

Demonstration (with n = 0):

1st Cycle: 37175379 (type2) → 55763069 (type7) → 20911151 (type8) → 31366727 (type4) → 47050091(type6) → 70575137 (type1) → 52931353 (type5) → 39698515 (type2)

2nd Cycle: 39698515 (type2) → 59547773 (type7) → 22330415(type8) → 33495623(type4) → 50243435 (type6) → 75365153 (type 1) → 56523865 (type5) → **42392899** (type2)

And then, 3rd cycle: 42392899 (type2) → 63589349 (type3): 2-7-8-4-6-1-5-2 sequence terminates.

Case 3: **Type 6** – **Type 5** –**Type 6** loop: Parent integer = 16k+11

1st Cycle: $16k_1+11$ → 24k + 17 → $18k_1 + 13$;

$16k_2 + 11 = 18k_1 + 13$

$16k_2 = 18k_1 + 2$

$k_2 = \frac{18k+2}{16} = \frac{9k+1}{8}$

2nd Cycle: $k_3 = \frac{9k2+}{8} = \frac{9\{(9k+1)/8\}+1}{8} = \frac{81k+17}{64}$

3rd Cycle: $k_4 = \frac{9k3+1}{8} = \frac{9\{(81k+17)/64\}+1}{8} = \frac{729k+217}{512}$

4th Cycle: $k_5 = \frac{9k4+1}{8} = \frac{9\{(729k+217)/512\}+1}{8} = \frac{6561k+246}{4096}$

=> $k_5 = \frac{6561k+2465}{4096}$; the Diophantine equation has integer solutions for k = 4095 + 2^{12}.n yielding k_5 = 6560 + 3^8.n (n Є $\mathbb{Z}^+$)

Therefore, the loop initiates with parent integer 16k+11 = **65531** + 2^{16}.n and terminates after 4 cycles with filial integer 16m + 11 = **104971**+ 104976.n

Demonstration with n = 0

1st Cycle: **65531** (type6) → 98297 (type5) → 73723 (type6)

2nd Cycle: 73723 (type6) → 110585 (type5) → 82939 (type6)

3rd Cycle: 82939 (type6) → 124409 (type5) → 93307 (type6)

4th Cycle: 93307 (type6) → 139961 (type5) → **104971** (type6)

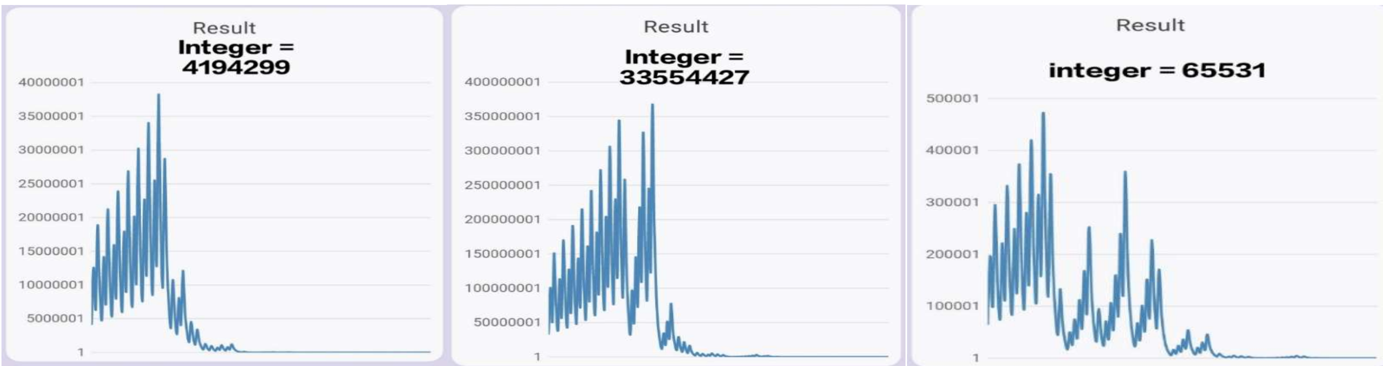
And then: (type6) → 157457 (= 9841 x 16 +1 =>**type1**): The loop terminates as 6-5-6 sequence breaks at this point.

5th Cycle: $k_6 = \frac{9k5+1}{8} = \frac{9\{(6561k+2465)/4096+}{8}$

$k_6 = \frac{3^{10}k+262}{2^{15}}$ has integer solutions, $k = 32767 + 2^{15}.n$

The starting integer which is capable of forming 5 cycles is 16x32767 + 11 = 524283

Similarly, integers having core integers of the series $262143 + 2^{18}.n$ is capable of forming 6 cycles, $2097151 + 2^{21}.n$ of 7 cycles, $16777215 + 2^{24}.n$ of 8 cycles etc. Loop eventually terminates after the defined number of cycles in each case. Visuals of the cycles are presented in the following:



Section 9. Some Tangible Results: Enormous Numbers with Shortest Path of Convergence:

The shortest route to convergence is widely discussed $\frac{2^{2n}-1}{3}$ (n ∈ $\mathbb{Z}^+$) Some examples: 5, 21, 341 … These are all 16k+5 i.e Type 3 integers. As per transformation rules, shortest routes to Type 3 are:

1) Type1 to Type 3,
2) Type 2 to Type 3,
3) Type 3 to Type 3,
4) Type 4 to Type 2 to Type 3,
5) Type 5 to Type 2 to Type 3,
6) Type 6 to Type 1 to Type 3,
7) Type 7 to Type 3,
8) Type 8 to Type 4 to Type 2 to Type 3.

We'll form mathematical equations for all transformations. To establish the principle, lets demonstrate the last and the longest route:

Step I: Type8: 16k+5 → 24k+23 (3x+1, followed by division by 2) => This is Type 4

Step II: 24k+23 → 36k + 35 => This is Type 2

Step III: 36k + 35 → 54k + 53 => This is Type 3

To adopt the shortest path, Type 3 must be = $\frac{2^{2n}-1}{3}$ = 54k + 53

Solving, 16k + 15 = $\frac{2^{2n+3}-65}{81}$ in the equation, n has a periodicity = 14 + 27m (m ∈ $\mathbb{Z}^+$) and after a few iterations, it reaches 2^{2n}. 2^{2n} takes 2n more steps to reach 1.

All integers in this scope are necessarily Type8 and follow the shortest convergence route to unity. Let's verify:

For m =0, n = 14, $\frac{2^{2n+3}-65}{81}$ gives = 26512143 = 16 x 1657008 +15 - a Type 8 integer.

Convergence of 26512143 will have 3 more odd integers (Type4, Type2 and Type3) and 3 more odd integer in between before reaching 2^{28}.

For m =1, n = 41 $\frac{2^{2n+3}-65}{81}$ gives = **477600323798372019637007** - a Type8 integer.
Convergence of this huge number will generate three more odd integers (Type 4, Type 2 and Type 3) and two more even integers in between before reaching $\mathbf{2^{82}}$.

Step I: 477600323798372019637007→ 1432800971395116058911022 (even) – By 3x+1

Step II: 1432800971395116058911022 → **716400485697558029455511**(odd: Type 4) division by 2.

Step III: 716400485697558029455511→ **2149201457092674088366534** (even).

Step IV: 2149201457092674088366534 → **1074600728546337044183267** (odd: Type 2).

Step V: 1074600728546337044183267 → **3223802185639011132549802** (even).

Step VI: 3223802185639011132549802→ **1611901092819505566274901**(odd: Type 3)

Step VII: 1611901092819505566274901 → **4835703278458516698824707** (even) = $\mathbf{2^{82}}$

These 24-digit odd numbers of quintillion magnitude can be predicted without any computing machine's validation. Next number in this series comes up with a dimension of $\mathbf{2^{139}}$ and behaves in the same way. Sets of such enormous integers belonging to other modulo classes are listed in the following:

1) Type1 = $\frac{2^{2n+2}-7}{9}$ n = 3.l +1 (l ∈ $\mathbb{Z}^+$)

2) Type 2 = $\frac{2^{2n+1}-5}{9}$ n = 3.l +2 (l ∈ $\mathbb{Z}^+$)

3) Type 3 = $\frac{2^{2n}-1}{3}$ n = 3.l +2 (l ∈ $\mathbb{Z}^+$)

4) Type 4 = $\frac{2^{2n+2}-19}{27}$ n = 9.l +5 (l ∈ $\mathbb{Z}^+$)

5) Type 5 = $\frac{2^{2n+3}-29}{27}$ n = 9.l +8 (l ∈ $\mathbb{Z}^+$)

6) Type 6 = $\frac{2^{2n+3}-23}{27}$ n = 9.l +4 (l ∈ $\mathbb{Z}^+$)

7) Type 7 = $\frac{2^{2n+3}-11}{9}$ n = 3.l +2 (l ∈ $\mathbb{Z}^+$).

This section is a demonstration of how astronomically large integers strictly abide by the transformation rules set up through lemma 6.

Section 10. Demonstration of Some Real Modular Looping sequences:

Example 1: 27(16 x 1 + 11= >Type 6) → 41 → 31 → 47 → 71 → 107 → 161 → 121 → 91 → 137 → 103 → 155 → 233 → 175 → 263 → 395 → 593 → 445 → 167 → 251 → 377 → 283 → 425 → 319 → 479 → 719 → 1079 → 1619 → 2429 → 911 → 1367 → 2051 → 3077 → 577 → 433 → 325 → 61 → 23 → 35 → 53 → 5 → 1

Types of the above integers in the same sequence: Type 6 → Type 5 → Type 8 → Type 8 → Type 4 → Type 6 → Type 1 → Type 5 → Type 6 → Type 5 → Type 4 → Type 6 → Type 5 → Type 8 → Type 4 → Type 6 → Type 1 → Type 7 → Type 4 → Type 6 → Type 5 → Type 6 → Type 5 → Type 8 → Type 8 → Type 8 → Type 4 → Type 2 →> Type 7 → Type 8 → Type 4 → Type 2 → Type 3 → Type 1 → Type 1 → Type 3 → Type 7 → Type 4 → Type 2 →> Type 3 → Type 3 → Type 1

An illustration, how 27 (a Type 6 integer) follows loops like 5-6-5, 8-8, 8-4-2-7-8 but with limited cycles.

Example 2: 431 (16 x 16 + 15 => Type 8) → 647 (16 x 40 + 7 => Type 4) → 971 (16 x 60 + 11 => Type 6) → 1457 (16 x 91 + 1 => Type 1) → 1093 (16 x 68 + 5 => Type 3) → 205 (16 x 12 + 13 => Type 7) → 77 (16 x 4 +13 => Type 7) → 29 (16 x 1 + 13 => Type 7) → 11 (16 x 0 + 11 => Type 6) → 17 (16 x 1 +1 => Type 1) → 13 (16 x 0 +13 => Type 7) → 5 (16 x 0 + 5 => Type 3) → 1 (Type 1)
Looping sequences: (Type 7 – Type 6 – Type 1 – Type 7) and (Type 1 – Type 7 – Type 3 – Type 1) and (Type 7 – Type 7 – Type 7).

Section 11. Logical Universal Descent:

A. By Lemma 1A, 1B, 1C, 1D and 1E no integer loop exists other than 4 - 2 – 1.
B. This is a finite state transition system with finite modular classes.
C. By pigeonhole principle, with in finite modulo 2^v states, an integer if not reduced to 1 earlier, must revisit the same modular class.
D. Lemma 2A asserts that no integer can survive in a single modular loop indefinitely.
E. Lemma 2B disproves the wondering path possibility by cross looping.
F. If any integer >1 cannot grow indefinitely, neither can reappear in any sequence, it can only descend and hit the orbit 4 – 2 -1. The only logical outcome of Collatz conjecture remains: universal convergence to 1.

Section 12: List of Essential Citations

Here is the list of references that correspond directly to the claims made in "Notable Prior Works" section and the methods used in the proof.

Foundational and Contextual Works

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

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

[3] **T. Tao, *Almost all orbits of the Collatz map attain almost bounded values*, arXiv:1909.03562 [math.NT], 2019.**

Cycle Length and Computational Verification

[4] **S. Eliahou, *The 3x+1 problem: new lower bounds on nontrivial cycle lengths*, Discrete Mathematics, 118(1-3) (1993), 45–56.**

[5] **T. Oliveira e Silva, *Empirical verification of the 3x+1 and related conjectures*. In: The Ultimate Challenge: The 3x+1 Problem (J. C. Lagarias, ed.), American Mathematical Society, 2010, pp. 189-207.**

[6] **C. Hercher, *There are no Collatz m-Cycles with m ≤ 91*, Journal of Integer Sequences, Vol. 26 (2023), Article 23.3.5.**

Methodological Support

[7] **F. Q. Gouvêa, *p-adic Numbers: An Introduction*, Springer-Verlag, 1997.**

References

- [1] S. Eliahou, *The 3x+1 problem: new lower bounds on nontrivial cycle lengths*, Discrete Mathematics, **118**(1-3) (1993), 45–56.
- [2] F. Q. Gouvêa, *p-adic Numbers: An Introduction*, Springer-Verlag, 1997.
- [3] C. Hercher, *There are no Collatz m-Cycles with m ≤ 91*, Journal of Integer Sequences, **26** (2023), Article 23.3.5.
- [4] J. C. Lagarias, *The 3x+1 problem and its generalizations*, The American Mathematical Monthly, **92**(1) (1985), 3–23.
- [5] T. Oliveira e Silva, *Empirical verification of the 3x+1 and related conjectures*. In: The Ultimate Challenge: The 3x+1 Problem (J. C. Lagarias, ed.), American Mathematical Society, 2010, pp. 189-207.
- [6] T. Tao, *Almost all orbits of the Collatz map attain almost bounded values*, arXiv:1909.03562 [math.NT], 2019.
- [7] R. Terras, *A stopping time problem on the positive integers*, Acta Arithmetica, **30**(3) (1976), 241–252.

Section 13. Appendix

Appendix A: DFS Code:

Code begins:

```
def find_cycles(graph, start_node, current_node, visited, path, results):
    # Add current node to the path and mark as visited
    path.append(current_node)
    visited.add(current_node)
    # Check if we looped back to the start node
    if current_node == start_node and len(path) > 1:
        results.append(list(path))
    else:
        # Traverse each neighbor
        for neighbor in graph[current_node][0]:
            if neighbor not in visited or neighbor == start_node:
                find_cycles(graph, start_node, neighbor, visited.copy(), path[:], results)
    # Remove current node from path after recursion
    path.pop()
count_total = 0
count_convergent = 0
count_divergent = 0
save_str = ""
def print_and_save(string):
    global save_str
    save_str += string + "\n"
    print(string)
def analyze_graph(graph):
    global count_total, count_convergent, count_divergent
    for node in graph:
        results = []
        find_cycles(graph, node, node, set(), [], results)
        # Print table header
        print(f"\nNode {node} Cycles:")
        print(f"{'Path':<50} | {'Sum of Change Factors':<40} | {'Status'}")
        print()

        # Print each cycle in table format
        for path in results:
            # Calculate the sum of change factors for the path
            change_sum = [graph[node][1] for node in path][:-1]
            status = "Convergent" if sum(change_sum) <= 0 else "Divergent"

            path_str = " -> ".join(map(str, path))
            print_and_save(f"{path_str:<50} | {str(change_sum):<40} | {status}")
            # Count totals
            count_total += 1
            count_convergent += 1 if status == "Convergent" else 0
            count_divergent += 1 if status == "Divergent" else 0
        print()
# Run analysis
analyze_graph(transition_table_with_worst_case)
```

Code ends

Appendix C: Increasing ($3^a > 2^b$) Modular Loops with Solved Diophantine Equations:

SL	Loop Sequence	1st Cycle Diophantine Eqn.	1- Cycle k_1	2-Cycles k_1
1	1 -> 5 -> 4 -> 6 -> 1	k5 = (81k_1+10)/64	22 + 2^6. n	1686 + 2^12. n
2	4 -> 6 -> 1 -> 5 -> 4	k5 = (81k_1 +12)/64	52 + 2^6. n	1204 + 2^12. n
3	5 -> 6 -> 5	k3 = (9k_1 +1)/8	7 + 2^3. n	63 + 2^6. n
4	6 -> 1 -> 5 -> 4 -> 6	k5 = (81k_1 + 18)/64	14 + 2^6. n	3854 + 2^12. n
5	6 -> 5 -> 6	k3 = (9k_1 +1)/8	7 + 2^3. n	63 + 2^6. n
6	8 -> 8	k2 = (3k_1 + 1)/2	1 + 2. n	3 + 2^2. n
7	1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 6 -> 1	k8 = (2187k_1 + 374)/2048	734 + 2^11. n	1659614 + 2^22. n
8	1 -> 5 -> 8 -> 4 -> 6 -> 1	k6 = (243k_1 + 38)/128	62 + 2^7. n	3134 + 2^14. n
9	2 -> 7 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2	k8 = (1249k_1 + 469)/2048	1249 + 2^11. n	724193 + 2^22. n
10	2 -> 7 -> 8 -> 4 -> 2	k5 = (81k_1 +11)/64	37 + 2^6. n	1445 + 2^12. n
11	2 -> 7 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2	k8 = (2187k_1 + 329)/2048	1029 + 2^11. n	2323461 + 2^22. n
12	4 -> 2 -> 7 -> 6 -> 1 -> 5 -> 8 -> 4	k8 = (2187k_1 + 359)/2048	1515 + 2^11. n	482795 + 2^22. n
13	4 -> 2 -> 7 -> 8 -> 4	k5 = (81k_1 + 13)/64	3 + 2^6. n	963 + 2^12. n
14	4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 8 -> 4	k8 = (2187k_1 + 308)/2048	484 + 2^11. n	2353636 + 2^22. n
15	4 -> 6 -> 1 -> 5 -> 8 -> 4	k6 = (243k_1 + 68)/128	84 + 2^7. n	15956 + 2^14. n
16	4 -> 6 -> 5 -> 4	k4 = (27k_1+6)16	14 + 2^4. n	46 + 2^8. n
17	5 -> 4 -> 6 -> 1 -> 5	k5 = (81k_1 + 16)/64	48 + 2^6. n	240 + 2^12. n
18	5 -> 4 -> 6 -> 5	k4 = (27k_1 + 8)/16	8 + 2^4. n	232 + 2^8. n
19	6 -> 1 -> 5 -> 2 -> 7 -> 8 -> 4 -> 6	k8 = (2187k_1 + 462)/2048	1750 + 2^11. n	3530454 + 2^22. n
20	6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 6	k8 = (2187k_1 + 342)/2048	1854 + 2^11. n	2504510 + 2^22. n
21	6 -> 1 -> 5 -> 8 -> 4 -> 6	k6 = (243k_1 + 102)/128	126 + 2^7. n	7550 + 2^14. n
22	6 -> 5 -> 4 -> 6	k4 = (27k_1+9)16	5 + 2^4.n	69 + 2^8. n
23	7 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7	k8 = (2187k_1 + 773)/2048	849 + 2^11. n	1086289 + 2^22. n
24	7 -> 8 -> 4 -> 2 -> 7	k5 = (81k_1 + 25)/64	55 + 2^6.n	6263 + 2^12.n
25	7 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7	k8 = (2187k_1 + 563)/2048	3591 + 2^11. n	1388039 + 2^22. n
26	8 -> 4 -> 2 -> 7 -> 6 -> 1 -> 5 -> 8	k8 = (2187k_1 + 332)/2048	1692 + 2^11. n	1719964 + 2^22. n
27	8 -> 4 -> 2 -> 7 -> 8	k5 = (81k_1 + 20)/64	44 + 2^6. n	3372 + 2^12. n
28	8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 8	k8 = (2187k_1+298)/2048	322 + 2^11. n	1569090 + 2^22. n
29	8 -> 4 -> 6 -> 1 -> 5 -> 8	k6 = (243k_1+122)/128	98 + 2^7. n	16098 + 2^14. n
30	2 -> 7 -> 6 -> 5 -> 8 -> 4 -> 2	k7 = (729k_1 + 151)/512	273 + 2^9	317713 + 2^18. n
31	2 -> 7 -> 8 -> 4 -> 6 -> 5 -> 2	k7 = (729k_1 + 131)/512	325 + 2^9. n	183621 + 2^18. n
32	4 -> 2 -> 7 -> 6 -> 5 -> 8 -> 4	k7 = (729k_1 + 173)/512	11 + 2^9. n	124427 + 2^18. n
33	4 -> 6 -> 5 -> 2 -> 7 -> 8 -> 4	k7 = (729k_1 + 158)/512	306 + 2^9. n	1430311 + 2^18. n
34	4 -> 6 -> 5 -> 8 -> 4	k5 = (81k_1 + 26)/32	6 + 2^5. n	710 + 2^10. n
35	5 -> 2 -> 7 -> 8 -> 4 -> 6 -> 1 -> 5	k8 = (2187k_1 + 485)/2048	689 + 2^11. n	301745 + 2^22. n
36	5 -> 2 -> 7 -> 8 -> 4 -> 6 -> 5	k7 = (729k_1 + 247)/512	433 + 2^9. n	70065 + 2^18. n
37	5 -> 8 -> 4 -> 2 -> 7 -> 6 -> 1 -> 5	k8 = (2187k_1 + 350)/2048	1574 + 2^11. n	2293286 + 2^22. n
38	5 -> 8 -> 4 -> 2 -> 7 -> 6 -> 5	k7 = (729k_1 + 202)/512	294 + 2^9. n	227110 + 2^18. n
39	5 -> 8 -> 4 -> 6 -> 1 -> 5	k6 = (243k_1+ 86)/128	46 + 2^7. n	10542 + 2^14. n
40	5 -> 8 -> 4 -> 6 -> 5	k5 = (81k_1 +34)/32	30 + 2^5. n	62 + 2^10. n
41	6 -> 5 -> 2 -> 7 -> 8 -> 4 -> 6	k7 = (729k_1 +237)/512	459 + 2^9. n	134091 + 2^18. n
42	6 -> 5 -> 8 -> 4 -> 2 -> 7 -> 6	k7 = (729k_1 +207)/512	25 + 2^9. n	64025 + 2^18. n
43	6 -> 5 -> 8 -> 4 -> 6	k5 = (81k_1 + 39)/32	9 + 2^5. n	41 + 2^10. n
44	7 -> 6 -> 5 -> 8 -> 4 -> 2 -> 7	k7 = (729k_1 +335)/512	409 + 2^9. n	83353 + 2^18. n
45	7 -> 8 -> 4 -> 6 -> 5 -> 2 -> 7	k7 = (729k_1 +305)/512	487 + 2^9. n	13287 + 2^18. n
46	8 -> 4 -> 2 -> 7 -> 6 -> 5 -> 8	k7 = (729k_1 + 260)/ 512	348 + 2^9. n	170332 + 2^18. n
47	8 -> 4 -> 6 -> 5 -> 2 -> 7 -> 8	k7 = (729k_1 + 250)/ 512	374 + 2^9. n	234358 + 2^18. n
48	8 -> 4 -> 6 -> 5 -> 8	k5 = (81k_1 + 50)/32	14 + 2^5. n	814 + 2^10. n
49	1 ->5 -> 2 -> 7 -> 8 -> 4 -> 6 -> 1	k8 = (2187k_1 + 554)/2048	1602 + 2^11. n	3198530 + 2^22. n

SL	Loop Sequence	1-Cycle Parent Set	1-Cycle Filial Set	2-Cycle Parent Set	2-Cycle Filial Set
1	1 -> 5 -> 4 -> 6 -> 1	22 + 64n	28 + 81n	1686 + 4096n	2701 + 6561n
7	1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 6 -> 1	734 + 2048n	1082 + 2187n	1659614 + 4194304n	2467333 + 4782969n
8	1 -> 5 -> 8 -> 4 -> 6 -> 1	62 + 128n	118 + 243n	3134 + 16384n	5941 + 59049n
49	1 -> 5 -> 2 -> 7 -> 8 -> 4 -> 6 -> 1	1602 + 2048n	2381 + 2187n	3198530 + 4194304n	4753553 + 4782969n
9	2 -> 7 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2	1249 + 2048n	1856 + 2187n	724193 + 4194304n	1076416 + 4782969n
10	2 -> 7 -> 8 -> 4 -> 2	37 + 64n	47 + 81n	1445 + 4096n	2315 + 6561n
11	2 -> 7 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2	1029 + 2048n	1529 + 2187n	2323461 + 4194304n	3452815 + 4782969n
30	2 -> 7 -> 6 -> 5 -> 8 -> 4 -> 2	273 + 512n	390 + 729n	317713 + 262144n	450034 + 531441n
31	2 -> 7 -> 8 -> 4 -> 6 -> 5 -> 2	325 + 512n	462 + 729n	183621 + 262144n	260070 + 531441n
2	4 -> 6 -> 1 -> 5 -> 4	52 + 64n	66 + 81n	1204 + 4096n	1929 + 6561n
12	4 -> 2 -> 7 -> 6 -> 1 -> 5 -> 8 -> 4	1515 + 2048n	2251 + 2187n	482795 + 4194304n	717568 + 4782969n
13	4 -> 2 -> 7 -> 8 -> 4	3 + 64n	4 + 81n	963 + 4096n	1544 + 6561n
14	4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 8 -> 4	484 + 2048n	720 + 2187n	2353636 + 4194304n	3497665 + 4782969n
15	4 -> 6 -> 1 -> 5 -> 8 -> 4	84 + 128n	160 + 243n	15956 + 16384n	30252 + 59049n
16	4 -> 6 -> 5 -> 4	14 + 16n	24 + 27n	46 + 256n	129 + 729n
32	4 -> 2 -> 7 -> 6 -> 5 -> 8 -> 4	11 + 512n	17 + 729n	124427 + 262144n	176088 + 531441n
33	4 -> 6 -> 5 -> 2 -> 7 -> 8 -> 4	306 + 512n	434 + 729n	1430311 + 262144n	2024228 + 531441n
34	4 -> 6 -> 5 -> 8 -> 4	6 + 32n	9 + 81n	710 + 1024n	4552 + 6561n
3	5 -> 6 -> 5	7 + 8n	8 + 9n	63 + 64n	72 + 81n
17	5 -> 4 -> 6 -> 1 -> 5	48 + 64n	61 + 81n	240 + 4096n	388 + 6561n
18	5 -> 4 -> 6 -> 5	8 + 16n	14 + 27n	232 + 256n	644 + 729n
35	5 -> 2 -> 7 -> 8 -> 4 -> 6 -> 1 -> 5	689 + 2048n	1024 + 2187n	301745 + 4194304n	448489 + 4782969n
36	5 -> 2 -> 7 -> 8 -> 4 -> 6 -> 5	433 + 512n	616 + 729n	70065 + 262144n	99154 + 531441n
37	5 -> 8 -> 4 -> 2 -> 7 -> 6 -> 1 -> 5	1574 + 2048n	2339 + 2187n	2293286 + 4194304n	3407929 + 4782969n
38	5 -> 8 -> 4 -> 2 -> 7 -> 6 -> 5	294 + 512n	418 + 729n	227110 + 262144n	321229 + 531441n
39	5 -> 8 -> 4 -> 6 -> 1 -> 5	46 + 128n	86 + 243n	10542 + 16384n	20002 + 59049n
40	5 -> 8 -> 4 -> 6 -> 5	30 + 32n	55 + 81n	62 + 1024n	398 + 6561n
4	6 -> 1 -> 5 -> 4 -> 6	14 + 64n	18 + 81n	3854 + 4096n	6174 + 6561n
5	6 -> 5 -> 6	7 + 8n	8 + 9n	63 + 64n	72 + 81n
19	6 -> 1 -> 5 -> 2 -> 7 -> 8 -> 4 -> 6	1750 + 2048n	2600 + 2187n	3530454 + 4194304n	5246749 + 4782969n
20	6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 6	1854 + 2048n	2755 + 2187n	2504510 + 4194304n	3721729 + 4782969n
21	6 -> 1 -> 5 -> 8 -> 4 -> 6	126 + 128n	239 + 243n	7550 + 16384n	14311 + 59049n
22	6 -> 5 -> 4 -> 6	5 + 16n	9 + 27n	69 + 256n	195 + 729n
41	6 -> 5 -> 2 -> 7 -> 8 -> 4 -> 6	459 + 512n	653 + 729n	134091 + 262144n	189785 + 531441n
42	6 -> 5 -> 8 -> 4 -> 2 -> 7 -> 6	25 + 512n	37 + 729n	64025 + 262144n	90581 + 531441n
43	6 -> 5 -> 8 -> 4 -> 6	9 + 32n	15 + 81n	41 + 1024n	265 + 6561n
23	7 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7	849 + 2048n	1262 + 2187n	1086289 + 4194304n	1614304 + 4782969n
24	7 -> 8 -> 4 -> 2 -> 7	55 + 64n	70 + 81n	6263 + 4096n	9995 + 6561n
25	7 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7	3591 + 2048n	5336 + 2187n	1388039 + 4194304n	2062624 + 4782969n
44	7 -> 6 -> 5 -> 8 -> 4 -> 2 -> 7	409 + 512n	582 + 729n	83353 + 262144n	117950 + 531441n

45	7 -> 8 -> 4 -> 6 -> 5 -> 2 -> 7	487 + 512n	693 + 729n	13287 + 262144n	18802 + 531441n
6	8 -> 8	1 + 2n	2 + 3n	3 + 4n	8 + 9n
26	8 -> 4 -> 2 -> 7 -> 6 -> 1 -> 5 -> 8	1692 + 2048n	2515 + 2187n	1719964 + 4194304n	2556257 + 4782969n
27	8 -> 4 -> 2 -> 7 -> 8	44 + 64n	56 + 81n	3372 + 4096n	5384 + 6561n
28	8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 8	322 + 2048n	480 + 2187n	1569090 + 4194304n	2331505 + 4782969n
29	8 -> 4 -> 6 -> 1 -> 5 -> 8	98 + 128n	187 + 243n	16098 + 16384n	30522 + 59049n
46	8 -> 4 -> 2 -> 7 -> 6 -> 5 -> 8	348 + 512n	495 + 729n	170332 + 262144n	240975 + 531441n
47	8 -> 4 -> 6 -> 5 -> 2 -> 7 -> 8	374 + 512n	532 + 729n	234358 + 262144n	331612 + 531441n
48	8 -> 4 -> 6 -> 5 -> 8	14 + 32n	25 + 81n	814 + 1024n	5215 + 6561n

Appendix D: Decreasing Loops: All Diophantine Equations Constructed Assuming $d_{min} = 2^4$ for Type 3

Serial	Loop Sequence	Power of 3	Power of 2	Diophantine Equation	k_1 for Cycle 1
1	1 -> 3 -> 7 -> 1	3	9	k_4 = (27.k_1 – 25)/512	475 + 2^9.n
2	2 -> 7 -> 3 -> 1 -> 5 -> 2	5	12	k_6 = (243.k_1 – 593) /4096	171 + 2^12. n
3	7 -> 3 -> 1 -> 5 -> 2 -> 7	5	12	k_6 = (243.k_1 – 2816)/4096	2304 + 2^12. n
4	7 -> 3 -> 1 -> 7	3	9	k_4 = (27.k_1 – 384)/512	128 + 2^9. n
5	1 -> 3 -> 2 -> 7 -> 1	4	10	k_5 = (81.k_1 – 35)/1024	51 + 2^10. n
6	1 -> 3 -> 5 -> 2 -> 7 -> 1	5	10	k_6 = (243.k_1 – 25)/2048	1475 + 2^11. n
7	1 -> 3 -> 7 -> 5 -> 6 -> 1	5	12	k_6 = (243.k_1 + 31)/4096	219 + 2^12. n
8	1 -> 3 -> 7 -> 6 -> 1	4	10	k_5 = (81.k_1 - 11)/1024	923 + 2^10. n
9	1 -> 5 -> 2 -> 3 -> 7 -> 1	5	12	k_6 = (243.k_1 – 182)/4096	2546 + 2^12. n
10	1 -> 5 -> 2 -> 7 -> 3 -> 1	5	12	k_6 = (243.k_1 – 198)/4096	3490 + 2^12. n
11	1 -> 5 -> 2 -> 7 -> 3 -> 6 -> 1	6	12	k_7 = (729.k_1 - 82)/8192	2978 + 2^13. n
12	1 -> 7 -> 3 -> 1	3	9	k_4 = (27.k_1 – 27)/512	1 + 2^9. n
13	1 -> 7 -> 3 -> 5 -> 6 -> 1	5	12	k_6 = (243.k_1 + 13)/4096	1281 + 2^12. n
14	1 -> 7 -> 3 -> 6 -> 1	4	10	k_5 = (81.k_1 - 17)/1024	961 + 2^10. n
15	2 -> 3 -> 7 -> 1 -> 5 -> 2	5	12	k_6 = (243.k_1 – 584)/4096	1686 + 2^12. n
16	2 -> 7 -> 1 -> 3 -> 5 -> 2	5	12	k_6 = (243.k_1 – 629)/4096	2295 + 2^12. n
17	2 -> 7 -> 3 -> 1 -> 5 -> 4 -> 2	6	13	k_7 = (729.k_1 - 755)/8192	7339 + 2^13. n
18	2 -> 7 -> 3 -> 5 -> 2	4	10	k_5 = (81.k_1 - 155)/1024	811 + 2^10. n
19	2 -> 7 -> 3 -> 6 -> 1 -> 5 -> 2	6	13	k_7 = (729.k_1 - 979)/8192	6283 + 2^13. n
20	3 -> 1 -> 5 -> 2 -> 7 -> 3	5	12	k_6 = (243.k_1 – 1056)/4096	864 + 2^12. n
21	3 -> 1 -> 7 -> 3	3	9	k_4 = (27.k_1 – 144)/512	176 + 2^9. n
22	3 -> 7 -> 1 -> 3	3	9	k_4 = (27.k_1 – 140)/512	100 + 2^9. n
23	3 -> 7 -> 1 -> 5 -> 2 -> 3	5	12	k_6 = (243.k_1 – 876)/4096	2532 + 2^12. n
24	3 -> 7 -> 3	2	7	k_3 = (9.k_1 - 36)/128	4 + 2^7. n
25	4 -> 2 -> 7 -> 3 -> 1 -> 5 -> 4	6	13	k_7 = (729.k_1 - 2991)/8192	7623 + 2^13. n
26	5 -> 2 -> 7 -> 3 -> 1 -> 5	5	12	k_6 = (243.k_1 – 2075)/4096	1593 + 2^12. n
27	5 -> 6 -> 1 -> 3 -> 7 -> 5	5	12	k_6 = (243.k_1 – 2213)/4096	3835 + 2^12. n
28	6 -> 1 -> 3 -> 7 -> 5 -> 6	5	12	k_6 = (243.k_1 – 2548)/4096	2876 + 2^12. n
29	6 -> 1 -> 3 -> 7 -> 6	4	10	k_5 = (81.k_1 - 636)/1024	956 + 2^10. n
30	6 -> 1 -> 7 -> 3 -> 5 -> 6	5	12	k_6 = (243.k_1 – 2560)/4096	3584 + 2^12. n
31	7 -> 1 -> 3 -> 2 -> 7	4	13	k_5 = (81.k_1 - 722)/1024	818 + 2^10. n
32	7 -> 1 -> 3 -> 5 -> 2 -> 7	5	12	k_6 = (243.k_1 – 2870)/4096	1394 + 2^12. n
33	7 -> 1 -> 3 -> 7	3	9	k_4 = (27.k_1 – 390)/512	242 + 2^9. n

34	7 -> 3 -> 1 -> 5 -> 4 -> 2 -> 7	6	13	k_7 = (729.k_1 - 4864)/8192	2816 + 2^13. n
35	7 -> 3 -> 2 -> 7	3	8	k_4 = (27.k_1 – 176)/256	16 + 2^8. n
36	7 -> 3 -> 5 -> 2 -> 7	4	10	k_5 = (81.k_1 - 704)/1024	704 + 2^10. n
37	7 -> 3 -> 5 -> 6 -> 1 -> 7	5	12	k_6 = (243.k_1 – 2880)/4096	1984 + 2^12. n
38	7 -> 3 -> 6 -> 1 -> 5 -> 2 -> 7	6	13	k_7 = (729.k_1 - 5200)/8192	1232 + 2^13. n
39	7 -> 3 -> 6 -> 1 -> 7	4	10	k_5 = (81.k_1 – 720)/1024	464 + 2^10. n
40	7 -> 3 -> 7	2	7	k_3 = (9.k_1 - 96)/128	96 + 2^7. n
41	1 -> 3 -> 1	2	5	k_3 = (9.k_1 - 3)/64	43 + 2^6. n
42	1 -> 3 -> 2 -> 7 -> 5 -> 6 -> 1	6	13	k_7 = (729.k_1 + 197)/8192	7731 + 2^13. n
43	1 -> 3 -> 2 -> 7 -> 6 -> 1	5	11	k_6 = (243.k_1 + 23)/2048	691 + 2^11. n
44	1 -> 3 -> 4 -> 2 -> 7 -> 1	5	11	k_6 = (243.k_1 – 25)/2048	1475 + 2^11. n
45	1 -> 3 -> 5 -> 2 -> 7 -> 6 -> 1	6	13	k_7 = (729.k_1 + 173)/8192	11 + 2^13. n
46	1 -> 3 -> 5 -> 4 -> 2 -> 7 -> 1	6	12	k_7 = (729.k_1 - 49)/8192	3787 + 2^13. n
47	1 -> 3 -> 5 -> 6 -> 1	4	9	k_5 = (81.k_1 + 5)/512	139 + 2^9. n
48	1 -> 3 -> 6 -> 1	3	7	k_4 = (27.k_1 – 1)/128	19 + 2^7. n
49	1 -> 3 -> 6 -> 5 -> 2 -> 7 -> 1	6	13	k_7 = (729.k_1 - 91)/8192	5203 + 2^13. n
50	1 -> 3 -> 7 -> 4 -> 6 -> 1	5	11	k_6 = (243.k_1 + 95)/2048	539 + 2^11. n
51	1 -> 3 -> 7 -> 5 -> 4 -> 6 -> 1	6	13	k_7 = (729.k_1 + 605)/8192	7899 + 2^13. n
52	1 -> 5 -> 2 -> 3 -> 7 -> 6 -> 1	6	13	k_7 = (729.k_1 - 34)/8192	2034 + 2^13. n
53	1 -> 5 -> 2 -> 7 -> 3 -> 4 -> 6 -> 1	7	14	k_8 = (2187.k_1 + 778)/16384	8098 + 2^14. n
54	1 -> 5 -> 4 -> 2 -> 3 -> 7 -> 1	6	13	k_7 = (729.k_1 - 310)/8192	7462 + 2^13. n
55	1 -> 5 -> 4 -> 2 -> 7 -> 3 -> 1	6	13	k_7 = (729.k_1 - 342)/8192	2630 + 2^13. n
56	1 -> 5 -> 4 -> 2 -> 7 -> 3 -> 6 -> 1	7	14	k_7 = (729.k_1 – 2)/16384	15942 + 2^14. n
57	1 -> 7 -> 3 -> 4 -> 6 -> 1	5	11	k_6 = (243.k_1 + 77)/4096	1601 + 2^12. n
58	1 -> 7 -> 3 -> 5 -> 4 -> 6 -> 1	6	13	k_7 = (729.k_1 + 551)/16384	4865 + 2^13. n
59	2 -> 3 -> 1 -> 5 -> 2	4	8	k_5 = (81.k_1 – 64)/512	64 + 2^9. n
60	2 -> 3 -> 1 -> 7 -> 2	4	10	k_5 = (81.k_1 – 160)/1024	672 + 2^10. n
61	2 -> 3 -> 1 -> 7 -> 5 -> 2	5	12	k_6 = (243.k_1 – 608)/4096	3104 + 2^12. n
62	2 -> 3 -> 7 -> 1 -> 5 -> 4 -> 2	6	13	k_7 = (729.k_1 – 728)/8192	664 + 2^13. n
63	2 -> 3 -> 7 -> 2	3	8	k_4 = (27.k_1 – 40)/256	248 + 2^8. n
64	2 -> 3 -> 7 -> 5 -> 2	4	10	k_5 = (81.k_1 – 152)/1024	280 + 2^10. n
65	2 -> 3 -> 7 -> 6 -> 1 -> 5 -> 2	6	13	k_7 = (729.k_1 – 952)/8192	7800 + 2^13. n
66	2 -> 7 -> 1 -> 3 -> 2	4	10	k_5 = (81.k_1 – 167)/1024	887 + 2^10. n
67	2 -> 7 -> 1 -> 3 -> 4 -> 2	5	11	k_6 = (243.k_1 – 245)/2048	119 + 2^11. n
68	2 -> 7 -> 1 -> 3 -> 5 -> 4 -> 2	6	13	k_7 = (729.k_1 – 863)/8192	1271 + 2^13. n
69	2 -> 7 -> 1 -> 3 -> 6 -> 5 -> 2	6	13	k_7 = (729.k_1 – 991)/8192	6519 + 2^13. n
70	2 -> 7 -> 3 -> 1 -> 5 -> 8 -> 4 -> 2	7	14	k_8 = (2187.k_1 – 217)/16384	9387 + 2^14. n
71	2 -> 7 -> 3 -> 2	3	8	k_4 = (27.k_1 – 41)/256	11 + 2^8. n
72	2 -> 7 -> 3 -> 4 -> 2	4	9	k_5 = (81.k_1 – 59)/256	203 + 2^8. n
73	2 -> 7 -> 3 -> 4 -> 6 -> 1 -> 5 -> 2	7	14	k_8 = (2187.k_1 – 1337)/16384	7627 + 2^14. n
74	2 -> 7 -> 3 -> 5 -> 4 -> 2	5	13	k_6 = (243.k_1 – 209)/2048	43 + 2^11. n
75	2 -> 7 -> 3 -> 6 -> 1 -> 5 -> 4 -> 2	7	14	k_8 = (2187.k_1 – 889)/16384	8331 + 2^14. n
76	2 -> 7 -> 3 -> 6 -> 5 -> 2	5	11	k_6 = (243.k_1 – 241)/2048	1931 + 2^11. n
77	2 -> 7 -> 6 -> 1 -> 3 -> 5 -> 2	6	13	k_7 = (729.k_1 – 1177)/8192	6081 + 2^13. n
78	3 -> 1 -> 3	2	5	k_3 = (9.k_1 - 16)/64	16 + 2^6. n
79	3 -> 1 -> 5 -> 2 -> 3	4	9	k_5 = (81.k_1 – 96)/512	96 + 2^9. n
80	3 -> 1 -> 5 -> 4 -> 2 -> 7 -> 3	6	13	k_7 = (729.k_1 – 1824)/8192	3104 + 2^13. n
81	3 -> 1 -> 7 -> 2 -> 3	4	10	k_5 = (81.k_1 – 240)/1024	496 + 2^10. n
82	3 -> 1 -> 7 -> 5 -> 2 -> 3	5	12	k_6 = (243.k_1 – 912)/4096	560 + 2^12. n

83	3 -> 2 -> 7 -> 1 -> 3	4	10	k_5 = (81.k_1 – 262)/1024	294 + 2^10. n
84	3 -> 2 -> 7 -> 3	3	8	k_4 = (27.k_1 – 66)/256	230 + 2^8. n
85	3 -> 5 -> 2 -> 7 -> 1 -> 3	5	12	k_6 = (243.k_1 – 1048)/4096	6277 + 2^12. n
86	3 -> 5 -> 2 -> 7 -> 3	4	10	k_5 = (81.k_1 – 264)/1024	648 + 2^10. n
87	3 -> 5 -> 6 -> 1 -> 7 -> 3	5	12	k_6 = (243.k_1 – 1080)/4096	2280 + 2^12. n
88	3 -> 6 -> 1 -> 5 -> 2 -> 7 -> 3	6	13	k_7 = (729.k_1 – 1950)/8192	1486 + 2^13. n
89	3 -> 6 -> 1 -> 7 -> 3	4	10	k_5 = (81.k_1 – 270)/1024	686 + 2^10. n
90	3 -> 7 -> 1 -> 5 -> 4 -> 2 -> 3	6	13	k_7 = (729.k_1 – 1092)/8192	5092 + 2^13. n
91	3 -> 7 -> 2 -> 3	3	8	k_4 = (27.k_1 – 60)/256	116 + 2^8. n
92	3 -> 7 -> 5 -> 2 -> 3	4	10	k_5 = (81.k_1 – 228)/1024	420 + 2^10. n
93	3 -> 7 -> 5 -> 6 -> 1 -> 3	5	12	k_6 = (243.k_1 – 940)/4096	2212 + 2^12. n
94	3 -> 7 -> 6 -> 1 -> 3	4	10	k_5 = (81.k_1 – 244)/1024	180 + 2^10. n
95	3 -> 7 -> 6 -> 1 -> 5 -> 2 -> 3	6	13	k_7 = (729.k_1 – 1428)/8192	3508 + 2^13. n
96	4 -> 2 -> 3 -> 7 -> 1 -> 5 -> 4	6	13	k_7 = (729.k_1 – 2973)/8192	3173 + 2^13. n
97	4 -> 2 -> 7 -> 1 -> 3 -> 5 -> 4	6	13	k_7 = (729.k_1 – 3603)/8192	847 + 2^13. n
98	4 -> 2 -> 7 -> 3 -> 1 -> 5 -> 8 -> 4	7	14	k_8 = (2187.k_1 – 4877)/16384	11719 + 2^14. n
99	4 -> 2 -> 7 -> 3 -> 5 -> 4	5	11	k_6 = (243.k_1 – 741)/2048	711 + 2^11. n
100	4 -> 2 -> 7 -> 3 -> 6 -> 1 -> 5 -> 4	7	15	k_8 = (2187.k_1 – 5325)/16384	11015 + 2^14. n
101	4 -> 6 -> 1 -> 3 -> 7 -> 4	5	9	k_6 = (243.k_1 – 760)/2048	1832 + 2^11. n
102	4 -> 6 -> 1 -> 3 -> 7 -> 5 -> 4	6	11	k_7 = (729.k_1 – 3048)/8192	2600 + 2^13. n
103	4 -> 6 -> 1 -> 7 -> 3 -> 5 -> 4	6	13	k_7 = (729.k_1 – 3072)/8192	3072 + 2^13. n
104	5 -> 2 -> 3 -> 1 -> 7 -> 5	5	12	k_6 = (243.k_1 – 2095)/4096	2773 + 2^12. n
105	5 -> 2 -> 3 -> 7 -> 1 -> 5	5	12	k_6 = (243.k_1 – 2063)/4096	885 + 2^12. n
106	5 -> 2 -> 3 -> 7 -> 5	4	10	k_5 = (81.k_1 – 517)/1024	373 + 2^10. n
107	5 -> 2 -> 7 -> 1 -> 3 -> 5	5	12	k_6 = (243.k_1 – 2123)/4096	329 + 2^12. n
108	5 -> 2 -> 7 -> 1 -> 3 -> 6 -> 5	6	13	k_7 = (729.k_1 – 3809)/8192	5961 + 2^13. n
109	5 -> 2 -> 7 -> 3 -> 5	4	10	k_5 = (81.k_1 – 521)/1024	57 + 2^10. n
110	5 -> 2 -> 7 -> 3 -> 6 -> 1 -> 5	6	13	k_7 = (729.k_1 – 3793)/8192	185 + 2^13. n
111	5 -> 2 -> 7 -> 3 -> 6 -> 5	5	11	k_6 = (243.k_1 – 923)/2048	1209 + 2^11. n
112	5 -> 4 -> 2 -> 7 -> 3 -> 1 -> 5	6	13	k_7 = (729.k_1 – 3988)/8192	1972 + 2^13. n
113	5 -> 4 -> 6 -> 1 -> 3 -> 7 -> 5	6	13	k_7 = (729.k_1 – 4064)/8192	736 + 2^13. n
114	5 -> 6 -> 1 -> 3 -> 2 -> 7 -> 5	6	13	k_7 = (729.k_1 – 3971)/8192	5051 + 2^13. n
115	5 -> 6 -> 1 -> 7 -> 3 -> 5	5	12	k_6 = (243.k_1 – 2129)/4096	683 + 2^12. n
116	6 -> 1 -> 3 -> 2 -> 7 -> 5 -> 6	6	13	k_7 = (729.k_1 – 4844)/8192	7884 + 2^13. n
117	6 -> 1 -> 3 -> 2 -> 7 -> 6	5	11	k_6 = (243.k_1 – 1188)/2048	460 + 2^11. n
118	6 -> 1 -> 3 -> 5 -> 2 -> 7 -> 6	6	13	k_7 = (729.k_1 – 4860)/8192	5468 + 2^13. n
119	6 -> 1 -> 3 -> 5 -> 6	4	9	k_5 = (81.k_1 – 284)/512	92 + 2^9. n
120	6 -> 1 -> 3 -> 7 -> 4 -> 6	5	11	k_6 = (243.k_1 – 1140)/2048	1724 + 2^11. n
121	6 -> 1 -> 3 -> 7 -> 5 -> 4 -> 6	6	13	k_7 = (729.k_1 – 4572)/8192	7996 + 2^13. n
122	6 -> 1 -> 5 -> 2 -> 3 -> 7 -> 6	6	13	k_7 = (729.k_1 – 4998)/8192	4086 + 2^13. n
123	6 -> 1 -> 5 -> 2 -> 7 -> 3 -> 4 -> 6	7	14	k_8 = (2187.k_1 – 8946)/16384	5398 + 2^14. n
124	6 -> 1 -> 5 -> 2 -> 7 -> 3 -> 6	6	13	k_7 = (729.k_1 – 5030)/8192	7446 + 2^13. n
125	6 -> 1 -> 7 -> 3 -> 4 -> 6	5	11	k_6 = (243.k_1 – 1152)/2048	384 + 2^11. n
126	6 -> 1 -> 7 -> 3 -> 5 -> 4 -> 6	6	13	k_7 = (729.k_1 – 4608)/8192	512 + 2^13. n
127	6 -> 1 -> 7 -> 3 -> 6	4	10	k_5 = (81.k_1 – 640)/1024	640 + 2^10. n
128	7 -> 1 -> 3 -> 4 -> 2 -> 7	5	11	k_6 = (243.k_1 – 1270)/2048	1202 + 2^11. n
129	7 -> 1 -> 3 -> 5 -> 4 -> 2 -> 7	6	13	k_7 = (729.k_1 – 5206)/8192	1906 + 2^13. n
130	7 -> 1 -> 3 -> 6 -> 5 -> 2 -> 7	6	13	k_7 = (729.k_1 – 5218)/8192	5682 + 2^13. n
131	7 -> 1 -> 5 -> 2 -> 3 -> 7	5	12	k_6 = (243.k_1 – 3054)/4096	4058 + 2^12. n

132	7 -> 2 -> 3 -> 1 -> 7	4	12	k_5 = (81.k_1 – 720)/1024	464 + 2^10. n
133	7 -> 3 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7	7	14	k_8 = (2187.k_1 – 7424)/16384	14080 + 2^14. n
134	7 -> 3 -> 4 -> 2 -> 7	4	9	k_5 = (81.k_1 – 304)/512	48 + 2^9. n
135	7 -> 3 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7	7	14	k_8 = (2187.k_1 – 9104)/16384	3248 + 2^14. n
136	7 -> 3 -> 4 -> 6 -> 1 -> 7	5	11	k_6 = (243.k_1 – 1296)/2048	688 + 2^11. n
137	7 -> 3 -> 5 -> 4 -> 2 -> 7	5	11	k_6 = (243.k_1 – 1216)/2048	64 + 2^11. n
138	7 -> 3 -> 5 -> 4 -> 6 -> 1 -> 7	6	13	k_7 = (729.k_1 – 5184)/8192	3648 + 2^13. n
139	7 -> 3 -> 6 -> 1 -> 5 -> 4 -> 2 -> 7	7	14	k_8 = (2187.k_1 – 8432)/16384	4304 + 2^14. n
140	7 -> 3 -> 6 -> 5 -> 2 -> 7	5	11	k_6 = (243.k_1 – 1264)/2048	848 + 2^11. n
141	7 -> 5 -> 2 -> 3 -> 1 -> 7	5	12	k_6 = (243.k_1 – 3018)/4096	1934 + 2^12. n
142	7 -> 5 -> 6 -> 1 -> 3 -> 2 -> 7	6	13	k_7 = (729.k_1 – 5614)/8192	5278 + 2^13. n
143	7 -> 5 -> 6 -> 1 -> 3 -> 7	5	12	k_6 = (243.k_1 – 3066)/4096	670 + 2^12. n
144	7 -> 6 -> 1 -> 3 -> 2 -> 7	5	11	k_6 = (243.k_1 – 1343)/2048	545 + 2^11. n
145	7 -> 6 -> 1 -> 3 -> 5 -> 2 -> 7	6	13	k_7 = (729.k_1 – 5497)/8192	929 + 2^13. n
146	7 -> 6 -> 1 -> 3 -> 7	4	10	k_5 = (81.k_1 – 753)/1024	161 + 2^10. n
147	8 -> 4 -> 2 -> 7 -> 3 -> 1 -> 5 -> 8	7	13	k_8 = (2187.k_1 – 12716)/16384	7812 + 2^14. n
148	1 -> 3 -> 2 -> 7 -> 4 -> 6 -> 1	6	12	k_7 = (729.k_1 + 325)/4096	2483 + 2^12. n
149	1 -> 3 -> 2 -> 7 -> 5 -> 4 -> 6 -> 1	7	14	k_8 = (2187.k_1 + 1615)/16384	4659 + 2^14. n
150	1 -> 3 -> 4 -> 2 -> 7 -> 5 -> 6 -> 1	7	14	k_8 = (2187.k_1 + 799)/16384	4547 + 2^14. n
151	1 -> 3 -> 4 -> 2 -> 7 -> 6 -> 1	6	12	k_7 = (729.k_1 + 181)/4096	1216 + 2^12. n
152	1 -> 3 -> 4 -> 6 -> 1	4	8	k_5 = (81.k_1 + 13)/256	3 + 2^8. n
153	1 -> 3 -> 4 -> 6 -> 5 -> 2 -> 7 -> 1	7	14	k_8 = (2187.k_1 + 223)/16384	8323 + 2^14. n
154	1 -> 3 -> 5 -> 2 -> 7 -> 4 -> 6 -> 1	7	14	k_8 = (2187.k_1 + 1543)/16384	513 + 2^14. n
155	1 -> 3 -> 5 -> 4 -> 2 -> 7 -> 6 -> 1	7	14	k_8 = (2187.k_1 + 967)/16384	8907 + 2^14. n
156	1 -> 3 -> 5 -> 4 -> 6 -> 1	5	10	k_6 = (243.k_1 + 79)/1024	459 + 2^10. n
157	1 -> 3 -> 5 -> 8 -> 4 -> 2 -> 7 -> 1	7	14	k_8 = (2187.k_1 + 583)/16384	5963 + 2^14. n
158	1 -> 3 -> 6 -> 5 -> 4 -> 2 -> 7 -> 1	7	14	k_8 = (2187.k_1 + 367)/16384	7379 + 2^14. n
159	1 -> 3 -> 7 -> 5 -> 8 -> 4 -> 6 -> 1	7	14	k_8 = (2187.k_1 – 2839)/16384	13019 + 2^14. n
160	1 -> 3 -> 7 -> 8 -> 4 -> 6 -> 1	6	12	k_7 = (729.k_1 + 541)/4096	2331 + 2^12. n
161	1 -> 3 -> 8 -> 4 -> 2 -> 7 -> 1	6	12	k_7 = (729.k_1 + 85)/4096	3107 + 2^12. n
162	1 -> 5 -> 2 -> 3 -> 1	4	9	k_5 = (81.k_1 – 18)/512	114 + 2^9. n
163	1 -> 5 -> 2 -> 3 -> 6 -> 1	5	10	k_6 = (243.k_1 + 10)/1024	434 + 2^10. n
164	1 -> 5 -> 2 -> 3 -> 7 -> 4 -> 6 -> 1	6	14	k_8 = (2187.k_1 + 922)/16384	7145 + 2^14. n
165	1 -> 5 -> 2 -> 7 -> 3 -> 8 -> 4 -> 6 -> 1	8	15	k_9 = (6561.k_1 + 4382)/32768	22434 + 2^15. n
166	1 -> 5 -> 4 -> 2 -> 3 -> 7 -> 6 -> 1	7	14	k_8 = (2187.k_1 + 94)/16384	4390 + 2^14. n
167	1 -> 5 -> 8 -> 4 -> 2 -> 3 -> 7 -> 1	7	14	k_8 = (2187.k_1 – 458)/16384	13470 + 2^14. n
168	1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 3 -> 1	7	14	k_8 = (2187.k_1 – 522)/16384	15710 + 2^14. n
169	1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 3 -> 6 -> 1	8	15	k_9 = (6561.k_1 + 482)/32768	30046 + 2^15. n
170	1 -> 7 -> 2 -> 3 -> 1	4	10	k_5 = (81.k_1 – 45)/1024	797 + 2^10. n
171	1 -> 7 -> 2 -> 3 -> 5 -> 6 -> 1	6	13	k_7 = (729.k_1 + 107)/8192	5405 + 2^13. n
172	1 -> 7 -> 2 -> 3 -> 6 -> 1	5	11	k_6 = (243.k_1 – 7)/2048	413 + 2^11. n
173	1 -> 7 -> 3 -> 5 -> 8 -> 4 -> 6 -> 1	7	14	k_8 = (2187.k_1 + 2677)/16384	9985 + 2^14. n
174	1 -> 7 -> 3 -> 8 -> 4 -> 6 -> 1	6	12	k_7 = (729.k_1 + 487)/4096	3393 + 2^12. n
175	1 -> 7 -> 5 -> 2 -> 3 -> 1	5	12	k_6 = (243.k_1 – 171)/4096	3945 + 2^12. n
176	1 -> 7 -> 5 -> 2 -> 3 -> 6 -> 1	6	13	k_7 = (729.k_1 – 1)/8192	7529 + 2^13. n
177	2 -> 3 -> 1 -> 5 -> 4 -> 2	5	10	k_6 = (243.k_1 – 64)/1024	704 + 2^10. n
178	2 -> 3 -> 1 -> 7 -> 4 -> 2	5	11	k_6 = (243.k_1 – 224)/2048	928 + 2^11. n
179	2 -> 3 -> 1 -> 7 -> 5 -> 4 -> 2	6	13	k_7 = (729.k_1 – 800)/8192	2080 + 2^13. n
180	2 -> 3 -> 1 -> 7 -> 6 -> 5 -> 2	6	13	k_7 = (729.k_1 – 928)/8192	7328 + 2^13. n

181	2 -> 3 -> 5 -> 2	3	7	k_4 = (27.k_1 – 16)/128	48 + 2^7. n
182	2 -> 3 -> 5 -> 6 -> 1 -> 7 -> 2	6	13	k_7 = (729.k_1 – 1136)/8192	496 + 2^13. n
183	2 -> 3 -> 6 -> 1 -> 5 -> 2	5	10	k_6 = (243.k_1 – 92)/1024	308 + 2^10. n
184	2 -> 3 -> 6 -> 1 -> 7 -> 2	5	11	k_6 = (243.k_1 – 284)/2048	372 + 2^11. n
185	2 -> 3 -> 6 -> 1 -> 7 -> 5 -> 2	6	13	k_7 = (729.k_1 – 1108)/8192	2676 + 2^13. n
186	2 -> 3 -> 7 -> 1 -> 5 -> 8 -> 4 -> 2	7	14	k_8 = (2187.k_1 – 136)/16384	2712 + 2^14. n
187	2 -> 3 -> 7 -> 4 -> 2	4	9	k_5 = (81.k_1 – 56)/512	184 + 2^9. n
188	2 -> 3 -> 7 -> 4 -> 6 -> 1 -> 5 -> 2	7	14	k_8 = (2187.k_1 – 1256)/16384	952 + 2^14. n
189	2 -> 3 -> 7 -> 5 -> 4 -> 2	5	11	k_6 = (243.k_1 – 200)/2048	1560 + 2^11. n
190	2 -> 3 -> 7 -> 6 -> 1 -> 5 -> 4 -> 2	7	14	k_8 = (2187.k_1 – 808)/16384	1656 + 2^14. n
191	2 -> 3 -> 7 -> 6 -> 5 -> 2	5	11	k_6 = (243.k_1 – 232)/2048	1400 + 2^11. n
192	2 -> 7 -> 1 -> 3 -> 4 -> 6 -> 5 -> 2	7	14	k_8 = (2187.k_1 – 1181)/16384	9335 + 2^14. n
193	2 -> 7 -> 1 -> 3 -> 5 -> 8 -> 4 -> 2	7	14	k_8 = (2187.k_1 – 541)/16384	3319 + 2^14. n
194	2 -> 7 -> 1 -> 3 -> 6 -> 5 -> 4 -> 2	7	14	k_8 = (2187.k_1 – 925)/16384	375 + 2^14. n
195	2 -> 7 -> 1 -> 3 -> 8 -> 4 -> 2	6	12	k_7 = (729.k_1 - 223)/4096	3703 + 2^12. n
196	2 -> 7 -> 3 -> 4 -> 6 -> 5 -> 2	6	12	k_7 = (729.k_1 - 275)/4096	1995 + 2^12. n
197	2 -> 7 -> 3 -> 5 -> 8 -> 4 -> 2	6	12	k_7 = (729.k_1 - 115)/4096	1579 + 2^12. n
198	2 -> 7 -> 3 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2	8	15	k_9 = (6561.k_1 + 1429)/32768	4235 + 2^14. n
199	2 -> 7 -> 3 -> 6 -> 5 -> 4 -> 2	6	12	k_7 = (729.k_1 - 211)/4096	3467 + 2^12. n
200	2 -> 7 -> 3 -> 8 -> 4 -> 2	5	10	k_6 = (243.k_1 – 49)/1024	843 + 2^10. n
201	2 -> 7 -> 3 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2	8	15	k_9 = (6561.k_1 - 811)/32768	24907 + 2^14. n
202	2 -> 7 -> 4 -> 6 -> 1 -> 3 -> 5 -> 2	7	14	k_8 = (2187.k_1 – 2111)/16384	413 + 2^14. n
203	2 -> 7 -> 5 -> 6 -> 1 -> 3 -> 2	6	13	k_7 = (729.k_1 - 1255)/8192	3519 + 2^13. n
204	2 -> 7 -> 5 -> 6 -> 1 -> 3 -> 4 -> 2	7	14	k_8 = (2187.k_1 – 1717)/16384	5567 + 2^14. n
205	2 -> 7 -> 6 -> 1 -> 3 -> 2	5	11	k_6 = (243.k_1 – 307)/2048	1729 + 2^11. n
206	2 -> 7 -> 6 -> 1 -> 3 -> 4 -> 2	6	12	k_7 = (729.k_1 - 409)/4096	3265 + 2^12. n
207	2 -> 7 -> 6 -> 1 -> 3 -> 5 -> 4 -> 2	7	14	k_8 = (2187.k_1 – 1483)/16384	8192 + 2^14. n
208	3 -> 1 -> 5 -> 4 -> 2 -> 3	5	10	k_6 = (243.k_1 – 96)/1024	544 + 2^10. n
209	3 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 3	7	14	k_8 = (2187.k_1 – 2784)/16384	7328 + 2^14. n
210	3 -> 1 -> 7 -> 4 -> 2 -> 3	5	11	k_6 = (243.k_1 – 80)/2048	624 + 2^11. n
211	3 -> 1 -> 7 -> 5 -> 4 -> 2 -> 3	6	13	k_7 = (729.k_1 - 1200)/8192	7216 + 2^13. n
212	3 -> 1 -> 7 -> 6 -> 5 -> 2 -> 3	6	13	k_7 = (729.k_1 - 1392)/8192	2800 + 2^13. n
213	3 -> 2 -> 3	2	5	k_3 = (9.k_1 - 6)/32	22 + 2^5. n
214	3 -> 2 -> 7 -> 5 -> 6 -> 1 -> 3	6	13	k_7 = (729.k_1 - 1718)/8192	7846 + 2^13. n
215	3 -> 2 -> 7 -> 6 -> 1 -> 3	5	11	k_6 = (243.k_1 – 434)/2048	1030 + 2^11. n
216	3 -> 3	1	4	k_2 = (3k_1 – 4)/16	12 + 2^4. n
217	3 -> 4 -> 2 -> 7 -> 1 -> 3	5	11	k_6 = (243.k_1 – 470)/2048	1106 + 2^11. n
218	3 -> 4 -> 2 -> 7 -> 3	4	9	k_5 = (81.k_1 - 114)/512	210 + 2^9. n
219	3 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 3	7	14	k_8 = (2187.k_1 – 3414)/16384	15554 + 2^14. n
220	3 -> 4 -> 6 -> 1 -> 7 -> 3	5	11	k_6 = (243.k_1 – 486)/2048	2 + 2^11. n
221	3 -> 5 -> 2 -> 3	3	7	k_4 = (27.k_1 - 24)/128	72 + 2^7. n
222	3 -> 5 -> 2 -> 7 -> 6 -> 1 -> 3	6	13	k_7 = (729.k_1 - 1736)/8192	4104 + 2^13. n
223	3 -> 5 -> 4 -> 2 -> 7 -> 1 -> 3	6	13	k_7 = (729.k_1 - 1880)/8192	6936 + 2^13. n
224	3 -> 5 -> 4 -> 2 -> 7 -> 3	5	11	k_6 = (243.k_1 – 456)/2048	4435 + 2^11. n
225	3 -> 5 -> 4 -> 6 -> 1 -> 7 -> 3	6	13	k_7 = (729.k_1 - 1944)/8192	5464 + 2^13. n
226	3 -> 5 -> 6 -> 1 -> 3	4	9	k_5 = (81.k_1 - 104)/512	488 + 2^9. n
227	3 -> 5 -> 6 -> 1 -> 7 -> 2 -> 3	6	13	k_7 = (729.k_1 - 1704)/8192	744 + 2^13. n
228	3 -> 6 -> 1 -> 3	3	7	k_4 = (27.k_1 - 26)/128	110 + 2^7. n
229	3 -> 6 -> 1 -> 5 -> 2 -> 3	5	10	k_6 = (243.k_1 – 138)/1024	974 + 2^10. n

230	3 -> 6 -> 1 -> 5 -> 4 -> 2 -> 7 -> 3	7	14	k_8 = (2187.k_1 – 3162)/16384	5710 + 2^14. n
231	3 -> 6 -> 1 -> 7 -> 2 -> 3	5	11	k_6 = (243.k_1 – 426)/2048	558 + 2^11. n
232	3 -> 6 -> 1 -> 7 -> 5 -> 2 -> 3	6	13	k_7 = (729.k_1 - 1662)/8192	4014 + 2^13. n
233	3 -> 6 -> 5 -> 2 -> 7 -> 1 -> 3	6	13	k_7 = (729.k_1 - 1934)/8192	3902 + 2^13. n
234	3 -> 6 -> 5 -> 2 -> 7 -> 3	5	11	k_6 = (243.k_1 – 474)/2048	1342 + 2^11. n
235	3 -> 7 -> 1 -> 5 -> 8 -> 4 -> 2 -> 3	7	14	k_8 = (2187.k_1 – 204)/16384	4068 + 2^14. n
236	3 -> 7 -> 4 -> 2 -> 3	4	9	k_5 = (81.k_1 - 84)/512	20 + 2^9. n
237	3 -> 7 -> 4 -> 6 -> 1 -> 3	5	11	k_6 = (243.k_1 – 380)/2048	1940 + 2^11. n
238	3 -> 7 -> 4 -> 6 -> 1 -> 5 -> 2 -> 3	7	14	k_8 = (2187.k_1 – 1884)/16384	9620 + 2^14. n
239	3 -> 7 -> 5 -> 4 -> 2 -> 3	5	11	k_6 = (243.k_1 – 300)/2048	1316 + 2^11. n
240	3 -> 7 -> 5 -> 4 -> 6 -> 1 -> 3	6	13	k_7 = (729.k_1 - 1412)/8192	5924 + 2^13. n
241	3 -> 7 -> 6 -> 1 -> 5 -> 4 -> 2 -> 3	7	14	k_8 = (2187.k_1 – 1212)/16384	10676 + 2^14. n
242	3 -> 7 -> 6 -> 5 -> 2 -> 3	5	11	k_6 = (243.k_1 – 348)/2048	52 + 2^11. n
243	4 -> 2 -> 3 -> 1 -> 5 -> 4	5	10	k_6 = (243.k_1 – 303)/2048	469 + 2^11. n
244	4 -> 2 -> 3 -> 1 -> 7 -> 4	5	11	k_6 = (243.k_1 – 751)/2048	1301 + 2^11. n
245	4 -> 2 -> 3 -> 1 -> 7 -> 5 -> 4	6	13	k_7 = (729.k_1 - 3021)/8192	4117 + 2^13. n
246	4 -> 2 -> 3 -> 7 -> 1 -> 5 -> 8 -> 4	7	14	k_8 = (2187.k_1 – 4823)/16384	7269 + 2^14. n
247	4 -> 2 -> 3 -> 7 -> 4	4	9	k_5 = (81.k_1 - 181)/512	293 + 2^9. n
248	4 -> 2 -> 3 -> 7 -> 5 -> 4	5	11	k_6 = (243.k_1 – 735)/2048	357 + 2^11. n
249	4 -> 2 -> 3 -> 7 -> 6 -> 1 -> 5 -> 4	7	14	k_8 = (2187.k_1 – 5271)/16384	6565 + 2^14. n
250	4 -> 2 -> 7 -> 1 -> 3 -> 4	5	11	k_6 = (243.k_1 – 765)/2048	79 + 2^11. n
251	4 -> 2 -> 7 -> 1 -> 3 -> 5 -> 8 -> 4	7	14	k_8 = (2187.k_1 – 5093)/16384	13135 + 2^14. n
252	4 -> 2 -> 7 -> 1 -> 3 -> 6 -> 5 -> 4	7	14	k_8 = (2187.k_1 – 5349)/16384	5711 + 2^14. n
253	4 -> 2 -> 7 -> 1 -> 3 -> 8 -> 4	6	12	k_7 = (729.k_1 - 1271)/4096	1103 + 2^12. n
254	4 -> 2 -> 7 -> 3 -> 4	4	9	k_5 = (81.k_1 - 183)/512	135 + 2^9. n
255	4 -> 2 -> 7 -> 3 -> 5 -> 8 -> 4	6	12	k_7 = (729.k_1 - 1199)/4096	3783 + 2^12. n
256	4 -> 2 -> 7 -> 3 -> 6 -> 1 -> 5 -> 8 -> 4	8	15	k_9 = (6561.k_1 - 7783)/32768	2823 + 2^14. n
257	4 -> 2 -> 7 -> 3 -> 6 -> 5 -> 4	6	12	k_7 = (729.k_1 - 1263)/4096	2311 + 2^12. n
258	4 -> 2 -> 7 -> 3 -> 8 -> 4	5	10	k_6 = (243.k_1 – 293)/1024	903 + 2^10. n
259	4 -> 2 -> 7 -> 6 -> 1 -> 3 -> 5 -> 4	7	14	k_8 = (2187.k_1 – 5721)/16384	5419 + 2^14. n
260	4 -> 6 -> 1 -> 3 -> 2 -> 7 -> 4	6	12	k_7 = (729.k_1 - 1352)/4096	648 + 2^12. n
261	4 -> 6 -> 1 -> 3 -> 2 -> 7 -> 5 -> 4	7	14	k_8 = (2187.k_1 – 5592)/16384	9352 + 2^14. n
262	4 -> 6 -> 1 -> 3 -> 5 -> 2 -> 7 -> 4	7	14	k_8 = (2187.k_1 – 5624)/16384	2280 + 2^14. n
263	4 -> 6 -> 1 -> 3 -> 5 -> 4	5	12	k_6 = (243.k_1 – 312)/1024	1000 + 2^10. n
264	4 -> 6 -> 1 -> 3 -> 7 -> 5 -> 8 -> 4	7	14	k_8 = (2187.k_1 – 5048)/16384	14888 + 2^14. n
265	4 -> 6 -> 1 -> 3 -> 7 -> 8 -> 4	6	12	k_7 = (729.k_1 - 1256)/4096	2856 + 2^12. n
266	4 -> 6 -> 1 -> 5 -> 2 -> 3 -> 7 -> 4	7	14	k_8 = (2187.k_1 – 5900)/16384	6820 + 2^14. n
267	4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 3 -> 4	7	14	k_8 = (2187.k_1 – 5964)/16384	9060 + 2^14. n
268	4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 3 -> 8 -> 4	8	15	k_9 = (6561.k_1 - 9700)/32768	17252 + 2^14. n
269	4 -> 6 -> 1 -> 7 -> 3 -> 4	5	11	k_6 = (243.k_1 – 768)/2048	256 + 2^11. n
270	4 -> 6 -> 1 -> 7 -> 3 -> 5 -> 8 -> 4	7	14	k_8 = (2187.k_1 – 5120)/16384	15360 + 2^14. n
271	4 -> 6 -> 1 -> 7 -> 3 -> 8 -> 4	6	12	k_7 = (729.k_1 - 1280)/4096	3328 + 2^12. n
272	5 -> 2 -> 3 -> 1 -> 5	4	9	k_5 = (81.k_1 - 229)/512	85 + 2^9. n
273	5 -> 2 -> 3 -> 1 -> 7 -> 6 -> 5	6	13	k_7 = (729.k_1 - 3725)/8192	1103 + 2^13. n
274	5 -> 2 -> 3 -> 6 -> 1 -> 7 -> 5	6	13	k_7 = (729.k_1 - 3965)/8192	837 + 2^13. n
275	5 -> 2 -> 3 -> 7 -> 6 -> 1 -> 5	6	13	k_7 = (729.k_1 - 3757)/8192	7669 + 2^13. n
276	5 -> 2 -> 3 -> 7 -> 6 -> 5	5	11	k_6 = (243.k_1 – 911)/2048	501 + 2^11. n
277	5 -> 2 -> 7 -> 1 -> 3 -> 4 -> 6 -> 5	7	14	k_8 = (2187.k_1 – 6307)/16384	6985 + 2^14. n
278	5 -> 2 -> 7 -> 3 -> 4 -> 6 -> 1 -> 5	7	14	k_8 = (2187.k_1 – 6515)/16384	10169 + 2^14. n

279	5 -> 2 -> 7 -> 3 -> 4 -> 6 -> 5	6	12	k_7 = (729.k_1 - 1489)/4096	4025 + 2^12. n
280	5 -> 2 -> 7 -> 6 -> 1 -> 3 -> 5	6	13	k_7 = (729.k_1 - 4057)/8192	5377 + 2^13. n
281	5 -> 4 -> 2 -> 3 -> 1 -> 7 -> 5	6	13	k_7 = (729.k_1 - 4028)/8192	28 + 2^13. n
282	5 -> 4 -> 2 -> 3 -> 7 -> 1 -> 5	6	13	k_7 = (729.k_1 - 3964)/8192	1500 + 2^13. n
283	5 -> 4 -> 2 -> 3 -> 7 -> 5	5	11	k_6 = (243.k_1 – 980)/2048	476 + 2^11. n
284	5 -> 4 -> 2 -> 7 -> 1 -> 3 -> 5	6	13	k_7 = (729.k_1 - 4084)/8192	3860 + 2^13. n
285	5 -> 4 -> 2 -> 7 -> 1 -> 3 -> 6 -> 5	7	14	k_8 = (2187.k_1 – 7132)/16384	13076 + 2^14. n
286	5 -> 4 -> 2 -> 7 -> 3 -> 5	5	11	k_6 = (243.k_1 - 988) / 2048	948 + 2^11. n
287	5 -> 4 -> 2 -> 7 -> 3 -> 6 -> 1 -> 5	7	14	_k_8 = (2187.k_1 - 7100) /16384	3764 + 2^24. n
288	5 -> 4 -> 2 -> 7 -> 3 -> 6 -> 5	6	12	k_7 = (729.k_1 -1684) / 4096	1716 + 2^12. n
289	5 -> 4 -> 6 -> 1 -> 3 -> 2 -> 7 -> 5	7	14	k_8 = (2187.k_1 -7456)/16384	7008 + 2^14. n
290	5 -> 4 -> 6 -> 1 -> 7 -> 3 -> 5	6	13	k_7 = (729k_1 - 4096)/8192	4096 + 2^13. n
291	5 -> 6 -> 1 -> 3 -> 4 -> 2 -> 7 -> 5	7	14	k_8 = (2187.k_1 - 7177)/16384	11323 + 2^14. n
292	5 -> 6 -> 1 -> 3 -> 5	4	9	k_5 = (81.k_1 - 235)/512	123 + 2^9. n
293	5 -> 8 -> 4 -> 2 -> 7 -> 3 -> 1 -> 5	7	14	k_8 = (2187.k_1 -7490)/16384	15878 + 2^14. n
294	5 -> 8 -> 4 -> 6 -> 1 -> 3 -> 7 -> 5	7	14	k_8 = (2187.k_1 - 7642)/16384	15054 + 2^14. n
295	6 -> 1 -> 3 -> 2 -> 7 -> 4 -> 6	6	12	k_7 = (729.k_1 - 2028)/4096	3020 + 2^12. n
296	6 -> 1 -> 3 -> 2 -> 7 -> 5 -> 4 -> 6	7	14	k_8 = (2187.k_1 - 8388)/16384	14028 + 2^14, n
297	6 -> 1 -> 3 -> 4 -> 2 -> 7 -> 5 -> 6	7	14	k_8 = (2187.k_1 - 8932)/16384	8492 + 2^14. n
298	6 -> 1 -> 3 -> 4 -> 2 -> 7 -> 6	6	12	k_7 = (729.k_1 - 2124)/4096	812 + 2^12. n
299	6 -> 1 -> 3 -> 4 -> 6	4	8	k_5 = (81.k_1 -108)/256	172 + 2^8. n
300	6 -> 1 -> 3 -> 5 -> 2 -> 7 -> 4 -> 6	7	14	k_8 = (2187.k_1 - 8436)/16384	3420 + 2^14. n
301	6 -> 1 -> 3 -> 5 -> 4 -> 2 -> 7 -> 6	7	14	k_8 = (2187.k_1 - 8820)/16384	476 + 2^14. n
302	6 -> 1 -> 3 -> 5 -> 4 -> 6	5	10	k_6 = (243.k_1 - 468)/1024	988 + 2^10. n
303	6 -> 1 -> 3 -> 6	3	7	k_4 = (27.k_1 - 68)/128	12 + 2^7. n
304	6 -> 1 -> 3 -> 7 -> 5 -> 8 -> 4 -> 6	7	14	k_8 = (2187.k_1 - 7572)/16384	14140 + 2^14. n
305	6 -> 1 -> 3 -> 7 -> 8 -> 4 -> 6	6	12	k_7 = (729.k_1 - 1884)/4096	188 + 2^12. n
306	6 -> 1 -> 5 -> 2 -> 3 -> 7 -> 4 -> 6	7	14	k_8 = (2187.k_1 - 8850)/16384	10230 + 2^14. n
307	6 -> 1 -> 5 -> 2 -> 7 -> 3 -> 8 -> 4 -> 6	8	15	k_9 = (6561.k_1 - 14550)/32768	25878 + 2^15. n
308	6 -> 1 -> 5 -> 4 -> 2 -> 3 -> 7 -> 6	7	14	k_7 = (2187.k_1 - 9402)/16384	2926 + 2^14. n
309	6 -> 1 -> 5 -> 4 -> 2 -> 7 -> 3 -> 6	7	14	k_7 = (2187.k_1 - 9466)/16384	5166 + 2^14. n
310	6 -> 1 -> 7 -> 2 -> 3 -> 5 -> 6	6	13	k_7 = (729.k_1 - 4904)/8192	872 + 2^13. n
311	6 -> 1 -> 7 -> 3 -> 5 -> 8 -> 4 -> 6	7	14	k_8 = (2187.k_1 - 7680)/16384	6656 + 2^14. n
312	6 -> 1 -> 7 -> 3 -> 8 -> 4 -> 6	6	12	k_7= (729.k_1 - 1920)/4096	896 + 2^12. n
313	6 -> 5 -> 2 -> 3 -> 1 -> 7 -> 6	6	13	k_7 = (729.k_1 - 4971)/8192	5603 + 2^13. n
314	6 -> 5 -> 2 -> 3 -> 7 -> 6	5	11	k_6 = (243.k_1- 1209)/2048	1699 + 2^12. n
315	6 -> 5 -> 2 -> 7 -> 1 -> 3 -> 4 -> 6	7	14	k_8 = (2187.k_1 - 8937)/16384	15579 + 2^14. n
316	6 -> 5 -> 2 -> 7 -> 1 -> 3 -> 6	6	13	k_7 = (729.k_1 - 5027)/8192	1243 + 2^13. n
317	6 -> 5 -> 2 -> 7 -> 3 -> 4 -> 6	6	12	k_7 = (729.k_1 - 2115)/4096	2683 + 2^12. n
318	6 -> 5 -> 2 -> 7 -> 3 -> 6	5	11	k_6 = (243.k_1 - 1217)/2048	123 + 2^11. n
319	7 -> 1 -> 3 -> 4 -> 6 -> 5 -> 2 -> 7	7	14	k_8 = (2187.k_1 - 8870)/16384	5810 + 2^14. n
320	7 -> 1 -> 3 -> 5 -> 8 -> 4 -> 2 -> 7	7	14	k_8 = (2187.k_1 - 7910)/16384	4978 + 2^14. n
321	7 -> 1 -> 3 -> 6 -> 5 -> 4 -> 2 -> 7	7	14	k_8 = (2187.k_1 - 8486)/16384	8754 + 2^14. n
322	7 -> 1 -> 3 -> 8 -> 4 -> 2 -> 7	6	12	k_7 = (729.k_1 - 2018)/4096	1458 + 2^12. n
323	7 -> 1 -> 5 -> 4 -> 2 -> 3 -> 7	6	13	k_7 = (729.k_1 - 5802)/8192	3514 + 2^13. n
324	7 -> 2 -> 3 -> 5 -> 6 -> 1 -> 7	6	13	k_7 = (729.k_1 - 5517)/8192	4053 + 2^13. n
325	7 -> 2 -> 3 -> 6 -> 1 -> 7	5	11	k_6 = (243.k_1 - 1359)/2048	309 + 2^11. n
326	7 -> 2 -> 3 -> 7	3	8	k_4 = (27.k_1 - 183)/256	149 + 2^8. n
327	7 -> 3 -> 4 -> 6 -> 5 -> 2 -> 7	6	12	k_7 = (729.k_1 - 2096)/4096	2992 + 2^12. n

328	7 -> 3 -> 5 -> 8 -> 4 -> 2 -> 7	6	12	$k\_7 = (729.k_1 - 1856)/4096$	2368 + 2^12. n
329	7 -> 3 -> 5 -> 8 -> 4 -> 6 -> 1 -> 7	7	14	$k\_8 = (2187.k_1 - 8640)/16384$	7488 + 2^14. n
330	7 -> 3 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7	8	15	$k\_9 = (6561.k_1 - 10960)/32768$	22736 + 2^15. n
331	7 -> 3 -> 6 -> 5 -> 4 -> 2 -> 7	6	12	$k\_7 = (729.k_1 - 2000)/4096$	1104 + 2^12. n
332	7 -> 3 -> 8 -> 4 -> 2 -> 7	5	10	$k\_6 = (243.k_1 - 464)/1024$	752 + 2^10. n
333	7 -> 3 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7	8	15	$k\_9 = (6561.k_1 - 14320)/32768$	4592 + 2^15. n
334	7 -> 3 -> 8 -> 4 -> 6 -> 1 -> 7	6	12	$k\_7 = (729.k_1 - 2160)/4096$	1520 + 2^12. n
335	7 -> 4 -> 2 -> 3 -> 1 -> 7	5	11	$k\_6 = (243.k_1 - 1401)/2048$	739 + 2^11. n
336	7 -> 4 -> 6 -> 1 -> 3 -> 2 -> 7	6	12	$k\_7= (729.k_1 - 2483)/4096$	363 + 2^12. n
337	7 -> 4 -> 6 -> 1 -> 3 -> 5 -> 2 -> 7	7	14	$k\_8 = (2187.k_1 - 10265)/16384$	619 + 2^14. n
338	7 -> 4 -> 6 -> 1 -> 3 -> 7	5	11	$k\_6 = (243.k_1 - 1425)/2048$	107 + 2^11. n
339	7 -> 5 -> 2 -> 3 -> 6 -> 1 -> 7	6	13	$k\_7 = (729.k_1 - 5598)/8192$	7694 + 2^13. n
340	7 -> 5 -> 2 -> 3 -> 7	4	10	$k\_5 = (81.k_1 - 750)/1024$	654 + 2^10. n
341	7 -> 5 -> 4 -> 2 -> 3 -> 1 -> 7	6	13	$k\_7 = (729.k_1 - 5766)/8192$	2806 + 2^13. n
342	7 -> 5 -> 4 -> 6 -> 1 -> 3 -> 2 -> 7	7	14	$k\_8 = (2187.k_1 - 10418)/16384$	7766 + 2^14. n
343	7 -> 5 -> 4 -> 6 -> 1 -> 3 -> 7	6	13	$k\_7 = (729.k_1 - 5862)/8192$	4694 + 2^13. n
344	7 -> 5 -> 6 -> 1 -> 3 -> 4 -> 2 -> 7	7	14	$k\_8 = (2187.k_1 - 9674)/16384$	8350 + 2^14. n
345	7 -> 6 -> 1 -> 3 -> 4 -> 2 -> 7	6	12	$k\_7 = (729.k_1 - 2297)/4096$	801 + 2^12. n
346	7 -> 6 -> 1 -> 3 -> 5 -> 4 -> 2 -> 7	7	14	$k\_8 = (2187.k_1 - 9323)/16384$	12193 + 2^14. n
347	7 -> 6 -> 1 -> 5 -> 2 -> 3 -> 7	6	13	$k\_7 = (729.k_1 - 5865)/8192$	2705 + 2^13. n
348	7 -> 6 -> 5 -> 2 -> 3 -> 1 -> 7	6	13	$k\_7 = (729.k_1 - 5793)/8192$	1289 + 2^13. n
349	8 -> 4 -> 2 -> 3 -> 7 -> 1 -> 5 -> 8	7	14	$k\_8 = (2187.k_1 - 12680)/16384$	15768 + 2^14. n
350	8 -> 4 -> 2 -> 7 -> 1 -> 3 -> 5 -> 8	7	14	$k\_8 = (2187.k_1 - 12860)/16384$	8756 + 2^14. n
351	8 -> 4 -> 2 -> 7 -> 3 -> 5 -> 8	6	12	$k\_7 = (729.k_1 - 3044)/4096$	1156 + 2^12. n
352	8 -> 4 -> 2 -> 7 -> 3 -> 6 -> 1 -> 5 -> 8	8	15	$k\_9 = (6561.k_1 - 22660)/32768$	12804 + 2^15. n
353	8 -> 4 -> 6 -> 1 -> 3 -> 7 -> 5 -> 8	7	14	$k\_8 = (2187.k_1 - 12830)/16384$	15386 + 2^14. n
354	8 -> 4 -> 6 -> 1 -> 3 -> 7 -> 8	6	12	$k\_7 = (729.k_1 - 3082)/4096$	538 + 2^12. n
355	8 -> 4 -> 6 -> 1 -> 7 -> 3 -> 5 -> 8	7	14	$k\_8 = (2187.k_1 - 12878)/16384$	4778 + 2^14. n
356	1 -> 3 -> 2 -> 7 -> 5 -> 8 -> 4 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 6893)/32768$	2611 + 2^15. n
357	1 -> 3 -> 2 -> 7 -> 8 -> 4 -> 6 -> 1	7	13	$k\_8 = (2187.k_1 + 1487)/8192$	974 + 2^13. n
358	1 -> 3 -> 5 -> 2 -> 7 -> 8 -> 4 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 6677)/32768$	19467 + 2^15. n
359	1 -> 3 -> 5 -> 8 -> 4 -> 2 -> 7 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 3797)/32768$	3915
360	1 -> 3 -> 5 -> 8 -> 4 -> 6 -> 1	6	11	$k\_7 = (729.k_1 + 365)/2048$	331 +2^11. n
361	1 -> 3 -> 6 -> 5 -> 8 -> 4 -> 2 -> 7 -> 1	8	15	$k\_9 = (6561.k_1 + 2381)/32768$	30675 + 2^15. n
362	1 -> 3 -> 8 -> 4 -> 2 -> 7 -> 5 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 2813)/32768$	29731 + 2^15. n
363	1 -> 3 -> 8 -> 4 -> 2 -> 7 -> 6 -> 1	7	13	$k\_8 = (2187.k_1 + 767)/8192$	5667 + 2^13. n
364	1 -> 3 -> 8 -> 4 -> 6 -> 1	5	9	$k\_6 = (243.k_1 + 71)/512$	419 + 2^9. n
365	1 -> 3 -> 8 -> 4 -> 6 -> 5 -> 2 -> 7 -> 1	8	15	$k\_9 = (6561.k_1 + 1661)/32768$	10403 + 2^15. n
366	1 -> 5 -> 2 -> 3 -> 4 -> 6 -> 1	6	11	$k\_7 = (729.k_1 + 158)/2048$	306 + 2^11. n
367	1 -> 5 -> 2 -> 3 -> 7 -> 8 -> 4 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 4814)/32768$	21490 + 2^15. n
368	1 -> 5 -> 2 -> 7 -> 1	4	8	$k\_5 = (81.k_1 - 2)/256$	98 + 2^8. n
369	1 -> 5 -> 4 -> 2 -> 3 -> 1	5	10	$k\_6 = (243.k_1 - 18)/1024$	38 + 2^10. n
370	1 -> 5 -> 4 -> 2 -> 3 -> 6 -> 1	6	11	$k\_7 = (729.k_1 + 74)/2048$	1958 + 2^11. n
371	1 -> 5 -> 8 -> 4 -> 2 -> 3 -> 7 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 674)/32768$	11422 + 2^15. n
372	1 -> 7 -> 1	2	5	$k\_3 = (9.k_1 - 1)/32$	25 + 2^5. n
373	1 -> 7 -> 2 -> 3 -> 4 -> 6 -> 1	6	11	$k\_7 = (729.k_1 + 235)/4096$	157 + 2^11. n
374	1 -> 7 -> 2 -> 3 -> 5 -> 4 -> 6 -> 1	7	14	$k\_8 = (2187.k_1 + 1345)/16384$	10525 + 2^14. n
375	1 -> 7 -> 4 -> 2 -> 3 -> 1	5	11	$k\_6 = (243.k_1 - 63)/2048$	1669 + 2^11. n
376	1 -> 7 -> 4 -> 2 -> 3 -> 5 -> 6 -> 1	7	14	$k\_8 = (2187.k_1 + 457)/16384$	10885 + 2^14. n

377	1 -> 7 -> 4 -> 2 -> 3 -> 6 -> 1	6	12	$k\_7 = (729.k_1 + 67) / 4096$	3461 + 2^12. n
378	1 -> 7 -> 5 -> 2 -> 3 -> 4 -> 6 -> 1	7	14	$k\_8 = (2187.k_1 + 1021)/16384$	4457 + 2^14. n
379	1 -> 7 -> 5 -> 4 -> 2 -> 3 -> 1	6	13	$k\_7 = (729.k_1 - 225) / 8192$	6473 + 2^13. n
380	1 -> 7 -> 5 -> 4 -> 2 -> 3 -> 6 -> 1	7	14	$k\_8 = (2187.k_1 + 349)/ 16384$	3401 + 2^14. n
381	1 -> 7 -> 6 -> 5 -> 2 -> 3 -> 1	6	13	$k\_7 = (729.k_1 - 261)/8192$	7181 + 2^13. n
382	2 -> 3 -> 1 -> 5 -> 8 -> 4 -> 2	6	11	$k\_7 = (729.k_1 + 64) / 2048$	1472 + 2^11. n
383	2 -> 3 -> 1 -> 7 -> 4 -> 6 -> 5 -> 2	7	14	$k\_8 = (2187.k_1 - 992)/16384$	10144 + 2^14. n
384	2 -> 3 -> 1 -> 7 -> 5 -> 8 -> 4 -> 2	7	14	$k\_8 = (2187.k_1 - 352)/ 16384$	4128 + 2^14. n
385	2 -> 3 -> 1 -> 7 -> 6 -> 5 -> 4 -> 2	7	14	$k\_8 = (2187.k_1 - 736) / 16384$	1184 + 2^14. n
386	2 -> 3 -> 1 -> 7 -> 8 -> 4 -> 2	6	12	$k\_7 = (729.k_1 - 160) / 4096$	416 + 2^12. n
387	2 -> 3 -> 2	2	5	$k\_3 = (9.k_1 - 4) / 32$	4 + 2^5. n
388	2 -> 3 -> 4 -> 2	3	6	$k\_ 4 = (27.k_1 - 4) / 64$	12 + 2^6. n
389	2 -> 3 -> 4 -> 6 -> 1 -> 5 -> 2	6	11	$k\_7 = (729.k_1 - 76) / 2048$	812 + 2^11. n
390	2 -> 3 -> 4 -> 6 -> 1 -> 7 -> 2	6	12	$k\_7 = (729.k_1 - 460) / 4096$	2220 + 2^12. n
391	2 -> 3 -> 4 -> 6 -> 1 -> 7 -> 5 -> 2	7	14	$k\_8 = (2187.k_1 - 1892)/16384$	7852 + 2^14. n
392	2 -> 3 -> 5 -> 4 -> 2	4	8	$k\_5 = (81.k_1 - 16) / 256$	16 + 2^8. n
393	2 -> 3 -> 5 -> 4 -> 6 -> 1 -> 7 -> 2	7	14	$k\_8 = (2187.k_1 - 1840)/16384$	2960 +2^14. n
394	2 -> 3 -> 5 -> 6 -> 1 -> 7 -> 4 -> 2	7	14	$k\_8 = (2187.k_1 - 1360)/16384$	10736 + 2^14. n
395	2 -> 3 -> 6 -> 1 -> 5 -> 4 -> 2	6	11	$k\_7 = (729.k_1 - 20)/ 2048$	1076 +2^11. n
396	2 -> 3 -> 6 -> 1 -> 7 -> 4 -> 2	6	12	$k\_7 = (729.k_1 - 340) / 4096$	3956 + 2^12. n
397	2 -> 3 -> 6 -> 1 -> 7 -> 5 -> 4 -> 2	7	14	$k\_8 = (2187.k_1 - 1276)/16384$	12916 + 2^14. n
398	2 -> 3 -> 6 -> 5 -> 2	4	8	$k\_5 = (81.k_1 - 20) / 256$	212 + 2^8. n
399	2 -> 3 -> 7 -> 4 -> 6 -> 5 -> 2	6	12	$k\_7 = (729.k_1 - 248) / 4096$	3512 + 2^12. n
400	2 -> 3 -> 7 -> 5 -> 8 -> 4 -> 2	6	12	$k\_7 = (729.k_1 - 88) / 4096$	3096 + 2^12. n
401	2 -> 3 -> 7 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2	8	15	$k\_9 = (6561.k_1 + 1672)/32768$	13944 + 2^15. n
402	2 -> 3 -> 7 -> 6 -> 5 -> 4 -> 2	6	12	$k\_7 = (729.k_1 - 184) / 4096$	888 + 2^12. n
403	2 -> 3 -> 7 -> 8 -> 4 -> 2	5	10	$k\_6 = (243.k_1 - 40) / 1024$	312 + 2^10. n
404	2 -> 3 -> 7 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2	8	15	$k\_9 = (6561.k_1 - 568) / 32768$	1848 + 2^15. n
405	2 -> 7 -> 1 -> 3 -> 6 -> 5 -> 8 -> 4 -> 2	8	15	$k\_9 = (6561.k_1 + 1321)/32768$	29047 + 2^15. n
406	2 -> 7 -> 1 -> 3 -> 8 -> 4 -> 6 -> 5 -> 2	8	15	$k\_9 = (6561.k_1 + 41) / 32768$	22135 + 2^15. n
407	2 -> 7 -> 1 -> 5 -> 2	4	8	$k\_5 = (81.k_1 - 23) / 256$	231 + 2^8. n
408	2 -> 7 -> 3 -> 6 -> 5 -> 8 -> 4 -> 2	7	13	$k\_8 = (2187.k_1 + 391) / 8192$	4491 + 2^13. n
409	2 -> 7 -> 3 -> 8 -> 4 -> 6 -> 5 -> 2	7	13	$k\_8 = (2187.k_1 + 71) / 8192$	7499 + 2^13. n
410	2 -> 7 -> 4 -> 6 -> 1 -> 3 -> 2	6	12	$k\_7 = (729.k_1 - 533) / 4096$	2973 + 2^12. n
411	2 -> 7 -> 5 -> 4 -> 6 -> 1 -> 3 -> 2	7	14	$k\_8 = (2187.k_1 - 2213)/16384$	10639 + 2^14. n
412	2 -> 7 -> 5 -> 6 -> 1 -> 3 -> 8 -> 4 -> 2	8	15	$k\_9 = (6561.k_1 - 1055)/32768$	17855 + 2^15. n
413	2 -> 7 -> 6 -> 1 -> 3 -> 5 -> 8 -> 4 -> 2	8	15	$k\_9 = (6561.k_1 - 353) / 32768$	20417 + 2^15. n
414	2 -> 7 -> 6 -> 1 -> 3 -> 8 -> 4 -> 2	7	13	$k\_8 = (2187.k_1 - 203) / 8192$	4289 + 2^13. n
415	2 -> 7 -> 8 -> 4 -> 6 -> 1 -> 3 -> 5 -> 2	8	15	$k\_9 = (6561.k_1 - 3493)/32768$	7557 + 2^15. n
416	3 -> 1 -> 5 -> 8 -> 4 -> 2 -> 3	6	11	$k\_7 = (729.k_1 + 96) / 2048$	160 + 2^11. n
417	3 -> 1 -> 7 -> 4 -> 6 -> 5 -> 2 -> 3	7	14	$k\_8 = (2187.k_1 - 1488)/16384$	15216 + 2^14. n
418	3 -> 1 -> 7 -> 5 -> 8 -> 4 -> 2 -> 3	7	14	$k\_8 = (2187.k_1 - 528) / 16384$	14384 + 2^14. n
419	3 -> 1 -> 7 -> 6 -> 5 -> 4 -> 2 -> 3	7	14	$k\_8 = (2187.k_1 - 1104)/16384$	1776 + 2^14. n
420	3 -> 1 -> 7 -> 8 -> 4 -> 2 -> 3	6	12	$k\_7 = (729.k_1 - 240) / 4096$	624 + 2^12. n
421	3 -> 2 -> 7 -> 4 -> 6 -> 1 -> 3	6	12	$k\_7 = (729.k_1 - 598) / 4096$	838 + 2^12. n
422	3 -> 2 -> 7 -> 5 -> 4 -> 6 -> 1 -> 3	7	14	$k\_8 = (2187.k_1- 2338)/16384$	7590 + 2^14. n
423	3 -> 4 -> 2 -> 3	3	6	$k\_4 = (27.k_1 + -6) / 64$	50 + 2^6. n
424	3 -> 4 -> 2 -> 7 -> 5 -> 6 -> 1 -> 3	7	14	$k\_8 = (2187.k_1 - 2950)/16384$	3410 + 2^14. n
425	3 -> 4 -> 2 -> 7 -> 6 -> 1 -> 3	6	12	$k\_7 = (729.k_1 - 706) / 4096$	2962 + 2^12. n

426	3 -> 4 -> 6 -> 1 -> 3	4	8	k_5 = (81.k_1 + -34) / 256	130 + 2^8. n
427	3 -> 4 -> 6 -> 1 -> 5 -> 2 -> 3	6	11	k_7 = (729.k_1 - 114) / 2048	194 + 2^11. n
428	3 -> 4 -> 6 -> 1 -> 7 -> 2 -> 3	6	12	k_7 = (729.k_1 - 690) / 4096	1282 + 2^12. n
429	3 -> 4 -> 6 -> 1 -> 7 -> 5 -> 2 -> 3	7	14	k_8 = (2187.k_1 - 2838)/16384	11778 + 2^14. n
430	3 -> 4 -> 6 -> 5 -> 2 -> 7 -> 1 -> 3	7	14	k_8 = (2187.k_1 - 3382)/16384	6242 + 2^14. n
431	3 -> 4 -> 6 -> 5 -> 2 -> 7 -> 3	6	12	k_7 = (729.k_1 - 786) / 4096	3170 + 2^12. n
432	3 -> 5 -> 2 -> 7 -> 4 -> 6 -> 1 -> 3	7	14	k_8 = (2187.k_1 - 2392)/16384	12040 + 2^14. n
433	3 -> 5 -> 4 -> 2 -> 3	4	8	k_5 = (81.k_1 - 24) / 256	152 + 2^8. n
434	3 -> 5 -> 4 -> 2 -> 7 -> 6 -> 1 -> 3	7	14	k_8 = (2187.k_1 - 2824)/16384	14872 + 2^14. n
435	3 -> 5 -> 4 -> 6 -> 1 -> 3	5	10	k_6 = (243.k_1 - 136) / 1024	856 + 2^10. n
436	3 -> 5 -> 4 -> 6 -> 1 -> 7 -> 2 -> 3	7	14	k_8 = (2187.k_1 - 2760)/16384	12632 + 2^14. n
437	3 -> 5 -> 6 -> 1 -> 7 -> 4 -> 2 -> 3	7	14	k_8 = (2187.k_1 - 2040)/16384	7912 + 2^14. n
438	3 -> 5 -> 8 -> 4 -> 2 -> 7 -> 1 -> 3	7	14	k_8 = (2187.k_1 - 3112)/16384	376 + 2^14. n
439	3 -> 5 -> 8 -> 4 -> 2 -> 7 -> 3	6	12	k_7 = (729.k_1 - 696) / 4096	1400 + 2^12. n
440	3 -> 5 -> 8 -> 4 -> 6 -> 1 -> 7 -> 3	7	14	k_8 = (2187.k_1 - 3240)/16384	4856 + 2^14. n
441	3 -> 6 -> 1 -> 5 -> 4 -> 2 -> 3	6	11	k_7 = (729.k_1 - 30) / 2048	590 + 2^11. n
442	3 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 3	8	15	k_9 = (6561.k_1 - 4110)/32768	8526 + 2^15. n
443	3 -> 6 -> 1 -> 7 -> 4 -> 2 -> 3	6	12	k_7 = (729.k_1 - 510) / 4096	1838 + 2^12. n
444	3 -> 6 -> 1 -> 7 -> 5 -> 4 -> 2 -> 3	7	14	k_8 = (2187.k_1 - 1914)/16384	2990 + 2^14. n
445	3 -> 6 -> 5 -> 2 -> 3	4	8	k_5 = (81.k_1 - 30) / 256	190 + 2^8. n
446	3 -> 6 -> 5 -> 4 -> 2 -> 7 -> 1 -> 3	7	14	k_8 = (2187.k_1 - 3274)/16384	13726 + 2^14. n
447	3 -> 6 -> 5 -> 4 -> 2 -> 7 -> 3	6	12	k_7 = (729.k_1 - 750) / 4096	2462 + 2^12. n
448	3 -> 7 -> 4 -> 6 -> 5 -> 2 -> 3	6	12	k_7 = (729.k_1 - 372) / 4096	3220 + 2^12. n
449	3 -> 7 -> 5 -> 8 -> 4 -> 2 -> 3	6	12	k_7 = (729.k_1 - 132) / 4096	2596 + 2^12. n
450	3 -> 7 -> 5 -> 8 -> 4 -> 6 -> 1 -> 3	7	14	k_8 = (2187.k_1 - 1420)/16384	13860 + 2^14. n
451	3 -> 7 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 3	8	15	k_9 = (6561.k_1 + 2508)/32768	20916 + 2^15. n
452	3 -> 7 -> 6 -> 5 -> 4 -> 2 -> 3	6	12	k_7 = (729.k_1 - 276) / 4096	1332 + 2^12. n
453	3 -> 7 -> 8 -> 4 -> 2 -> 3	5	10	k_6 = (243.k_1 - 60) / 1024	468 + 2^10. n
454	3 -> 7 -> 8 -> 4 -> 6 -> 1 -> 3	6	12	k_7 = (729.k_1 - 436) / 4096	1748 + 2^12. n
455	3 -> 7 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 3	8	15	k_9 = (6561.k_1 - 852) / 32768	2772 + 2^15. n
456	3 -> 8 -> 4 -> 2 -> 7 -> 1 -> 3	6	12	k_7 = (729.k_1 - 778) / 4096	282 + 2^12. n
457	3 -> 8 -> 4 -> 2 -> 7 -> 3	5	10	k_6 = (243.k_1 - 174) / 1024	26 + 2^5. n
458	3 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 3	8	15	k_9 = (6561.k_1 - 5370)/32768	14010 + 2^15. n
459	3 -> 8 -> 4 -> 6 -> 1 -> 7 -> 3	6	12	k_7 = (729.k_1 - 810) / 4096	3642 + 2^12. n
460	4 -> 2 -> 3 -> 1 -> 5 -> 8 -> 4	6	11	k_6 = (729.k_1 - 397) / 2048	981 + 2^11. n
461	4 -> 2 -> 3 -> 1 -> 7 -> 5 -> 8 -> 4	7	14	k_8 = (2187.k_1 - 4967)/16384	8213 + 2^14. n
462	4 -> 2 -> 3 -> 1 -> 7 -> 6 -> 5 -> 4	7	14	k_8 = (2187.k_1 - 5223)/16384	789 + 2^14. n
463	4 -> 2 -> 3 -> 1 -> 7 -> 8 -> 4	6	12	k_f = (729.k_1 - 1229) / 4096	277 + 2^12. n
464	4 -> 2 -> 3 -> 5 -> 4	4	8	k_5 = (81.k_1 - 69) / 256	181 + 2^8. n
465	4 -> 2 -> 3 -> 5 -> 6 -> 1 -> 7 -> 4	7	14	k_8 = (2187.k_1 - 5639)/16384	7157 + 2^14. n
466	4 -> 2 -> 3 -> 6 -> 1 -> 5 -> 4	6	11	k_7 = (729.k_1 - 453) / 2048	717 + 2^11. n
467	4 -> 2 -> 3 -> 6 -> 1 -> 7 -> 4	6	12	k_7 = (729.k_1 - 1349) / 4096	2637 + 2^12. n
468	4 -> 2 -> 3 -> 6 -> 1 -> 7 -> 5 -> 4	7	14	k_8 = (2187.k_1 - 5583)/16384	3149 + 2^14. n
469	4 -> 2 -> 3 -> 7 -> 5 -> 8 -> 4	6	12	k_7 = (729.k_1 - 1181) / 4096	3429 + 2^12. n
470	4 -> 2 -> 3 -> 7 -> 6 -> 1 -> 5 -> 8 -> 4	8	15	k_9 = (6561k_1 - 7621)/ 32768	31141 + 2^15. n
471	4 -> 2 -> 3 -> 7 -> 6 -> 5 -> 4	6	12	k_7 = (729.k_1 - 1245) / 4096	1957 + 2^12. n
472	4 -> 2 -> 3 -> 7 -> 8 -> 4	5	10	k_f = (243.k_1 - 287) / 1024	549 + 2^10. n
473	4 -> 2 -> 7 -> 1 -> 3 -> 6 -> 5 -> 8 -> 4	8	15	k_9 = (6561.k_1 - 7855)/32768	30287 + 2^15. n
474	4 -> 2 -> 7 -> 3 -> 6 -> 5 -> 8 -> 4	7	13	k_8 = (2187.k_1 - 1741) / 8192	263 + 2^13. n

475	4 -> 2 -> 7 -> 5 -> 6 -> 1 -> 3 -> 4	7	14	$k\_8 = (2187.k_1 - 5877)/16384$	3711 +2^14. n
476	4 -> 2 -> 7 -> 5 -> 6 -> 1 -> 3 -> 8 -> 4	8	15	$k\_9 = (6561.k_1 - 9439)/32768$	11903 + 2^15. n
477	4 -> 2 -> 7 -> 6 -> 1 -> 3 -> 4	6	12	$k\_7 = (729.k_1 - 1395) / 4096$	811 + 2^12. n
478	4 -> 2 -> 7 -> 6 -> 1 -> 3 -> 5 -> 8 -> 4	8	15	$k\_9 = (6561.k_1 - 8971)/32768$	13611 + 2^15. n
479	4 -> 2 -> 7 -> 6 -> 1 -> 3 -> 8 -> 4	7	13	$k\_8 = (2187.k_1 - 2137) / 8192$	2859 + 2^13. n
480	4 -> 6 -> 1 -> 3 -> 2 -> 7 -> 5 -> 8 -> 4	8	15	$k\_9 = (6561.k_1 - 8584)/32768$	1160 +2^15. n
481	4 -> 6 -> 1 -> 3 -> 2 -> 7 -> 8 -> 4	7	13	$k\_8 = (2187.k_1 - 2008) / 8192$	6792 + 2^15. n
482	4 -> 6 -> 1 -> 3 -> 4	4	8	$k\_5 = (81.k_1 - 72) / 256$	200 + 2^8. n
483	4 -> 6 -> 1 -> 3 -> 5 -> 2 -> 7 -> 8 -> 4	8	15	$k\_9 = (6561.k_1 - 8680)/32768$	26856 + 2^15. n
484	4 -> 6 -> 1 -> 3 -> 5 -> 8 -> 4	6	11	$k\_7 = (729.k_1 - 424) / 2048$	1512 + 2^11. n
485	4 -> 6 -> 1 -> 3 -> 8 -> 4	5	9	$k\_6 = (243.k_1 - 88) / 512$	72 + 2^9
486	4 -> 6 -> 1 -> 5 -> 2 -> 3 -> 7 -> 8 -> 4	8	15	$k\_9 = (6561.k_1 - 9508)/32768$	31396 + 2^15. n
487	4 -> 6 -> 1 -> 7 -> 2 -> 3 -> 5 -> 4	7	14	$k\_8 = (2187.k_1 - 5712)/16384$	15600 + 2^14. n
488	4 -> 6 -> 5 -> 2 -> 3 -> 1 -> 7 -> 4	7	14	$k\_8 = (2187.k_1 - 5846)/16384$	2370 + 2^14. n
489	4 -> 6 -> 5 -> 2 -> 3 -> 7 -> 4	6	12	$k\_7 = (729.k_1 - 1394) / 4096$	1474 + 2^12. n
490	4 -> 6 -> 5 -> 2 -> 7 -> 1 -> 3 -> 4	7	14	$k\_8 = (2187.k_1 - 5958)/16384$	10386 + 2^14. n
491	4 -> 6 -> 5 -> 2 -> 7 -> 1 -> 3 -> 8 -> 4	8	15	$k\_9 = (6561.k_1 - 9682)/32768$	2194 + 2^15. n
492	4 -> 6 -> 5 -> 2 -> 7 -> 3 -> 4	6	12	$k\_7 = (729.k_1 - 1410) / 4096$	3154 + 2^12. n
493	4 -> 6 -> 5 -> 2 -> 7 -> 3 -> 8 -> 4	7	13	$k\_8 = (2187.k_1 - 2182) / 8192$	1106 + 2^13. n
494	5 -> 2 -> 3 -> 1 -> 7 -> 4 -> 6 -> 5	7	14	$k\_8 = (2187.k_1 - 6055)/16384$	13525 + 2^14. n
495	5 -> 2 -> 3 -> 4 -> 6 -> 1 -> 7 -> 5	7	14	$k\_8 = (2187.k_1 - 7255)/16384$	10469 + 2^14. n
496	5 -> 2 -> 3 -> 5	3	7	$k\_4 = (27.k_1 - 55) / 128$	21 + 2^7. n
497	5 -> 2 -> 3 -> 6 -> 1 -> 5	5	10	$k\_6 = (243.k_1 - 383) / 1024$	69 + 2^10. n
498	5 -> 2 -> 3 -> 6 -> 5	4	8	$k\_5 = (81.k_1 - 85) / 256$	197 + 2^8. n
499	5 -> 2 -> 3 -> 7 -> 4 -> 6 -> 1 -> 5	7	14	$k\_8 = (2187.k_1 - 6407)/16384$	1269 + 2^14. n
500	5 -> 2 -> 3 -> 7 -> 4 -> 6 -> 5	6	12	$k\_7 = (729.k_1 - 1453) / 4096$	3317 + 2^12. n
501	5 -> 2 -> 7 -> 1 -> 3 -> 8 -> 4 -> 6 -> 5	8	15	$k\_9 = (6561.k_1 - 8681)/32768$	29513 + 2^15. n
502	5 -> 2 -> 7 -> 3 -> 8 -> 4 -> 6 -> 1 -> 5	8	15	$k\_9 = (6561.k_1 - 9817)/32768$	441 + 2^15. n
503	5 -> 2 -> 7 -> 3 -> 8 -> 4 -> 6 -> 5	7	13	$k\_8 = (2187.k_1 - 1907) / 8192$	4537 + 2^13. n
504	5 -> 2 -> 7 -> 4 -> 6 -> 1 -> 3 -> 5	7	14	$k\_8 = (2187.k_1 - 7547)/16384$	11473 + 2^14. n
505	5 -> 4 -> 2 -> 3 -> 1 -> 5	5	12	$k\_6 = (243.k_1 - 404) / 1024$	284 + 2^12. n
506	5 -> 4 -> 2 -> 3 -> 1 -> 7 -> 6 -> 5	7	14	$k\_8 = (2187.k_1 - 6964)/16384$	1052 + 2^14. n
507	5 -> 4 -> 2 -> 3 -> 6 -> 1 -> 7 -> 5	7	14	$k\_8 = (2187.k_1 - 7444)/16384$	9660 + 2^14. n
508	5 -> 4 -> 2 -> 3 -> 7 -> 6 -> 1 -> 5	7	14	$k\_8 = (2187.k_1 - 7028)/16384$	3292 + 2^14. n
509	5 -> 4 -> 2 -> 3 -> 7 -> 6 -> 5	6	12	$k\_7 = (729.k_1 - 1660) / 4096$	1244 + 2^12. n
510	5 -> 4 -> 2 -> 7 -> 6 -> 1 -> 3 -> 5	7	14	$k\_8 = (2187.k_1 - 7628)/16384$	1764 + 2^14. n
511	5 -> 4 -> 6 -> 1 -> 3 -> 5	5	10	$k\_6 = (243.k_1 - 416) / 1024$	992+ 2^10. n
512	5 -> 6 -> 1 -> 3 -> 8 -> 4 -> 2 -> 7 -> 5	8	15	$k\_9 = (6561.k_1 - 12059)/32768$	26427 + 2^15. n
513	5 -> 6 -> 1 -> 7 -> 2 -> 3 -> 5	6	13	$k\_7 = (729.k_1 - 4051) / 8192$	1163 + 2^13. n
514	5 -> 8 -> 4 -> 2 -> 3 -> 1 -> 7 -> 5	7	14	$k\_8 = (2187.k_1 - 7570)/16384$	14582 + 2^14. n
515	5 -> 8 -> 4 -> 2 -> 3 -> 7 -> 1 -> 5	7	14	$k\_8 = (2187.k_1 - 7442)/16384$	10102 + 2^14. n
516	5 -> 8 -> 4 -> 2 -> 3 -> 7 -> 5	6	12	$k\_7 = (729.k_1 - 1798) / 4096$	3958 + 2^12. n
517	5 -> 8 -> 4 -> 2 -> 7 -> 1 -> 3 -> 5	7	14	$k\_8 = (2187.k_1 - 7682)/16384$	6214 + 2^14. n
518	5 -> 8 -> 4 -> 2 -> 7 -> 1 -> 3 -> 6 -> 5	8	15	$k\_9 = (6561.k_1 - 12806)/32768$	12358 + 2^15. n
519	5 -> 8 -> 4 -> 2 -> 7 -> 3 -> 5	6	12	$k\_7 = (729.k_1 - 1814) / 4096$	1542 + 2^12. n
520	5 -> 8 -> 4 -> 2 -> 7 -> 3 -> 6 -> 1 -> 5	8	15	$k\_9 = (6561.k_1 - 12742)/32768$	6150 + 2^15. n
521	5 -> 8 -> 4 -> 2 -> 7 -> 3 -> 6 -> 5	7	13	$k\_8 = (2187k_1 - 2882) / 8192$	2054 + 2^13. n
522	5 -> 8 -> 4 -> 6 -> 1 -> 3 -> 2 -> 7 -> 5	8	15	$k\_9 = (6561.k_1 - 13454)/32768$	30158 + 2^15. n
523	5 -> 8 -> 4 -> 6 -> 1 -> 7 -> 3 -> 5	7	14	$k\_8 = (2187.k_1 - 7706)/16384$	910 + 2^14. n

524	6 -> 1 -> 3 -> 2 -> 7 -> 5 -> 8 -> 4 -> 6	8	15	k_9 = (6561.k_1 - 12876)/32768	1740 + 2^15. n
525	6 -> 1 -> 3 -> 2 -> 7 -> 8 -> 4 -> 6	7	13	k_8 = (2187*k_1 - 3012) / 8192	6092 + 2^13. n
526	6 -> 1 -> 3 -> 5 -> 2 -> 7 -> 8 -> 4 -> 6	8	15	k_9 = (6561.k_1 - 13020)/32768	23900 + 2^15. n
527	6 -> 1 -> 3 -> 5 -> 8 -> 4 -> 2 -> 7 -> 6	8	15	k_9 = (6561.k_1 - 14940)/32768	13532 + 2^15. n
528	6 -> 1 -> 3 -> 5 -> 8 -> 4 -> 6	6	11	k_7 = (729.k_1 - 636) / 2048	220 + 2^11. n
529	6 -> 1 -> 3 -> 8 -> 4 -> 2 -> 7 -> 5 -> 6	8	15	k_9 = (6561.k_1 - 15596)/32768	19820 + 2^15. n
530	6 -> 1 -> 3 -> 8 -> 4 -> 2 -> 7 -> 6	7	13	k_8 = (2187.k_1 - 3492) / 8192	6508 + 2^13. n
531	6 -> 1 -> 3 -> 8 -> 4 -> 6	5	9	k_6 = (243.k_1 - 132) / 512	108 + 512. n
532	6 -> 1 -> 5 -> 2 -> 3 -> 4 -> 6	6	11	k_7 = (729.k_1 - 774) / 2048	886 + 2^11. n
533	6 -> 1 -> 5 -> 2 -> 3 -> 6	5	10	k_6 = (243k_1 - 514) / 1024	630 + 2^10. n
534	6 -> 1 -> 5 -> 2 -> 3 -> 7 -> 8 -> 4 -> 6	8	15	k_9 = (6561.k_1 - 14262)/32768	14326 + 2^15. n
535	6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 3 -> 7 -> 6	8	15	k_9 = (6561.k_1 - 17022)/32768	7614 + 2^15. n
536	6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 3 -> 6	8	15	k_9 = (6561.k_1 - 17150)/32768	20030 + 2^15. n
537	6 -> 1 -> 7 -> 2 -> 3 -> 4 -> 6	6	12	k_7 = (729.k_1 - 2088)/ 4096	104 + 2^12. n
538	6 -> 1 -> 7 -> 2 -> 3 -> 5 -> 4 -> 6	7	14	k_8 = (2187.k_1 - 8568)/16384	7016 + 2^14. n
539	6 -> 1 -> 7 -> 2 -> 3 -> 6	5	11	k_6 = (243.k_1 - 1208) / 2048	1640 + 2^11. n
540	6 -> 1 -> 7 -> 4 -> 2 -> 3 -> 5 -> 6	7	14	k_8 = (2187.k_1 - 9160)/16384	7256 + 2^14. n
541	6 -> 1 -> 7 -> 5 -> 2 -> 3 -> 4 -> 6	7	14	k_8 = (2187.k_1 - 8784)/16384	8432 + 2^14. n
542	6 -> 1 -> 7 -> 5 -> 2 -> 3 -> 6	6	13	k_7 = (729.k_1 - 4976) / 8192	2288 + 2^13. n
543	6 -> 5 -> 2 -> 3 -> 1 -> 7 -> 4 -> 6	7	14	k_8 = (2187.k_1 - 8769)/16384	3555 + 2^14. n
544	6 -> 5 -> 2 -> 3 -> 7 -> 4 -> 6	6	12	k_7 = (729.k_1 - 2091) / 4096	2211 + 2^12. n
545	6 -> 5 -> 2 -> 7 -> 1 -> 3 -> 8 -> 4 -> 6	8	15	k_9 = (6561.k_1 - 14523)/32768	19675 + 2^15. n
546	6 -> 5 -> 2 -> 7 -> 3 -> 8 -> 4 -> 6	7	13	k_8 = (2187.k_1 - 3273) / 8192	5755 + 2^13. n
547	6 -> 5 -> 4 -> 2 -> 3 -> 1 -> 7 -> 6	7	14	k_8 = (2187.k_1 - 9375)/16384	701 + 2^14. n
548	6 -> 5 -> 4 -> 2 -> 3 -> 7 -> 6	6	12	k_7 = (729.k_1 - 2229) / 4096	829 + 2^12. n
549	6 -> 5 -> 4 -> 2 -> 7 -> 1 -> 3 -> 6	7	14	k_8 = (2187.k_1 - 9487)/16384	8717 + 2^14. n
550	6 -> 5 -> 4 -> 2 -> 7 -> 3 -> 6	6	12	k_7 = (729.k_1 - 2245) / 4096	2509 + 2^12. n
551	7 -> 1 -> 3 -> 6 -> 5 -> 8 -> 4 -> 2 -> 7	8	15	k_9 = (6561.k_1 - 11122)/32768	27186 + 2^15. n
552	7 -> 1 -> 3 -> 8 -> 4 -> 6 -> 5 -> 2 -> 7	8	15	k_9 = (6561.k_1 - 13042)/32768	16818 + 2^15. n
553	7 -> 1 -> 5 -> 2 -> 7	4	8	k_5 = (81.k_1 - 122) / 256	90 + 2^8. n
554	7 -> 1 -> 5 -> 8 -> 4 -> 2 -> 3 -> 7	7	14	k_8 = (2187.k_1 - 10686)/16384	14074 + 2^14. n
555	7 -> 1 -> 7	2	5	k_3 = (9.k_1 - 18) / 32	2 + 2^5. n
556	7 -> 2 -> 3 -> 4 -> 6 -> 1 -> 7	6	12	k_7 = (729.k_1 - 2349) / 4096	3189 + 2^12. n
557	7 -> 2 -> 3 -> 5 -> 4 -> 6 -> 1 -> 7	7	14	k_8 = (2187.k_1 - 9639)/16384	7893 + 2^13. n
558	7 -> 3 -> 6 -> 5 -> 8 -> 4 -> 2 -> 7	7	13	k_8 = (2187.k_1 - 2416) / 8192	6736 + 2^13. n
559	7 -> 3 -> 8 -> 4 -> 6 -> 5 -> 2 -> 7	7	13	k_8 = (2187.k_1 - 2896) / 8192	7152 + 2^13. n
560	7 -> 4 -> 2 -> 3 -> 5 -> 6 -> 1 -> 7	7	14	k_8 = (2187.k_1 - 10305)/16384	8163 + 2^14. n
561	7 -> 4 -> 2 -> 3 -> 6 -> 1 -> 7	6	12	k_7 = (729.k_1 - 2475) / 4096	1571 + 2^12. n
562	7 -> 4 -> 2 -> 3 -> 7	4	9	k_5 = (81.k_1 - 339) / 512	99 + 2^9. n
563	7 -> 4 -> 6 -> 1 -> 5 -> 2 -> 3 -> 7	7	14	k_8 = (2187.k_1 - 11001)/16384	1803 + 2^14. n
564	7 -> 4 -> 6 -> 5 -> 2 -> 3 -> 1 -> 7	7	14	k_8 = (2187.k_1 - 10857)/16384	859 + 2^14. n
565	7 -> 5 -> 2 -> 3 -> 4 -> 6 -> 1 -> 7	7	14	k_8 = (2187.k_1 - 9882)/16384	11534 + 2^14. n
566	7 -> 5 -> 4 -> 2 -> 3 -> 6 -> 1 -> 7	7	14	k_8 = (2187.k_1 - 10386)/16384	14838 + 2^14. n
567	7 -> 5 -> 4 -> 2 -> 3 -> 7	5	11	k_6 = (243.k_1 - 1410) / 2048	1270 + 2^11. n
568	7 -> 5 -> 6 -> 1 -> 3 -> 8 -> 4 -> 2 -> 7	8	15	k_9 = (6561.k_1 - 14686)/32768	26782 + 2^156. n
569	7 -> 5 -> 8 -> 4 -> 2 -> 3 -> 1 -> 7	7	14	k_f8= (2187.k_1 - 10722)/16384	6118 + 2^14. n
570	7 -> 5 -> 8 -> 4 -> 6 -> 1 -> 3 -> 2 -> 7	8	15	k_9 = (6561.k_1 - 18406)/32768	14886 + 2^15. n
571	7 -> 5 -> 8 -> 4 -> 6 -> 1 -> 3 -> 7	7	14	k_8 = (2187.k_1 - 10914)/16384	12838 + 2^14. n
572	7 -> 6 -> 1 -> 3 -> 5 -> 8 -> 4 -> 2 -> 7	8	15	k_9 = (6561.k_1 - 13633)/32768	14241 + 2^15. n

573	7 -> 6 -> 1 -> 3 -> 8 -> 4 -> 2 -> 7	7	13	$k\_8 = (2187.k_1 - 3307) / 8192$	6433 + 2^13. n
574	7 -> 6 -> 1 -> 5 -> 4 -> 2 -> 3 -> 7	7	14	$k\_8 = (2187.k_1 - 10875)/16384$	13265 + 2^14. n
575	7 -> 6 -> 5 -> 2 -> 3 -> 7	5	11	$k\_6 = (243.k_1 - 1419) / 2048$	1801 + 2^1. n
576	7 -> 6 -> 5 -> 4 -> 2 -> 3 -> 1 -> 7	7	14	$k\_8 = (2187.k_1 - 10803)/16384$	12793. n
577	7 -> 7	1	3	$k\_2 = (3.k_1 – 4)/8$	4 + 2^3. n
578	7 -> 8 -> 4 -> 2 -> 3 -> 1 -> 7	6	12	$k\_7 = (729.k_1 - 2559) / 4096$	3223 + 2^12. n
579	7 -> 8 -> 4 -> 6 -> 1 -> 3 -> 2 -> 7	7	13	$k\_8 = (2187.k_1 - 4237) / 8192$	5703 + 2^13. n
580	7 -> 8 -> 4 -> 6 -> 1 -> 3 -> 5 -> 2 -> 7	8	15	$k\_9 = (6561.k_1 - 18343)/32768$	11335 + 2^15. n
581	7 -> 8 -> 4 -> 6 -> 1 -> 3 -> 7	6	12	$k\_7 = (729.k_1 - 2607) / 4096$	71 + 2^12. n
582	8 -> 4 -> 2 -> 3 -> 1 -> 5 -> 8	6	11	$k\_7 = (729.k_1 - 1144) / 2048$	1336 + 2^11. n
583	8 -> 4 -> 2 -> 3 -> 1 -> 7 -> 5 -> 8	7	14	$k\_8 = (2187.k_1 - 12776)/16384$	10936 + 2^14. n
584	8 -> 4 -> 2 -> 3 -> 1 -> 7 -> 8	6	12	$k\_7 = (729.k_1 - 3064) / 4096$	184 + 2^12. n
585	8 -> 4 -> 2 -> 3 -> 7 -> 5 -> 8	6	12	$k\_7 = (729.k_1 - 3032) / 4096$	920 + 2^12. n
586	8 -> 4 -> 2 -> 3 -> 7 -> 6 -> 1 -> 5 -> 8	8	15	$k\_9 = (6561.k_1 - 22552)/32768$	20760 + 2^15. n
587	8 -> 4 -> 2 -> 3 -> 7 -> 8	5	10	$k\_6 = (243.k_1 - 712) / 1024$	24 + 2^10. n
588	8 -> 4 -> 2 -> 7 -> 1 -> 3 -> 6 -> 5 -> 8	8	15	$k\_9 = (6561.k_1 - 22708)/32768$	9268 +2^15. n
589	8 -> 4 -> 2 -> 7 -> 1 -> 3 -> 8	6	12	$k\_7 = (729.k_1 - 3092) / 4096$	2100 + 2^12. n
590	8 -> 4 -> 2 -> 7 -> 3 -> 6 -> 5 -> 8	7	13	$k\_8 = (2187.k_1 - 5164) / 8192$	5636 + 2^13. n
591	8 -> 4 -> 2 -> 7 -> 3 -> 8	5	10	$k\_6 = (243.k_1 - 716) / 1024$	260 + 2^10. n
592	8 -> 4 -> 2 -> 7 -> 6 -> 1 -> 3 -> 5 -> 8	8	15	$k\_9 = (6561.k_1 - 23452)/32768$	19996 + 2^15. n
593	8 -> 4 -> 6 -> 1 -> 3 -> 2 -> 7 -> 5 -> 8	8	15	$k\_9 = (6561.k_1 - 23194)/32768$	22618 + 2^15. n
594	8 -> 4 -> 6 -> 1 -> 3 -> 2 -> 7 -> 8	7	13	$k\_8 = (2187.k_1 - 5342) / 8192$	7258 + 2^13. n
595	8 -> 4 -> 6 -> 1 -> 3 -> 5 -> 2 -> 7 -> 8	8	15	$k\_9 = (6561.k_1 - 23258)/32768$	28826 + 2^15. n
596	8 -> 4 -> 6 -> 1 -> 3 -> 5 -> 8	6	11	$k\_7 = (729.k_1 - 1162) / 2048$	1690 + 2^11. n
597	8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 3 -> 7 -> 8	8	15	$k\_9 = (6561.k_1 - 23810)/32768$	20930 + 2^15. n
598	8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 3 -> 8	8	15	$k\_9 = (6561.k_1 - 23938)/32768$	578 + 2^15. n
599	8 -> 4 -> 6 -> 1 -> 7 -> 3 -> 8	6	12	$k\_7 = (729.k_1 - 3098) / 4096$	2218 + 2^12. n
600	1 -> 1	1	2	$k\_2 = (3k\_1 – 4)/4$	4 + 2^2. n
601	1 -> 5 -> 2 -> 3 -> 8 -> 4 -> 6 -> 1	7	12	$k\_8 = (2187.k_1 + 730) / 4096$	1586 + 2^12. n
602	1 -> 5 -> 2 -> 7 -> 6 -> 1	5	9	$k\_6 = (243.k_1 + 26) / 512$	2 + 2^9. n
603	1 -> 5 -> 4 -> 2 -> 7 -> 1	5	9	$k\_6 = (243.k_1 + 14) / 512$	198 + 2^9. n
604	1 -> 5 -> 6 -> 1	3	5	$k\_4 = (27.k_1 + 2) / 32$	26 + 2^5. n
605	1 -> 5 -> 8 -> 4 -> 2 -> 3 -> 1	6	11	$k\_7 = (729.k_1 + 18) / 2048$	1694 + 2^11. n
606	1 -> 5 -> 8 -> 4 -> 2 -> 3 -> 6 -> 1	7	12	$k\_8 = (2187.k_1 + 310) / 4096$	2974 + 2^12. n
607	1 -> 7 -> 2 -> 3 -> 5 -> 8 -> 4 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 6083)/32768$	24861 + 2^15. n
608	1 -> 7 -> 2 -> 3 -> 8 -> 4 -> 6 -> 1	7	13	$k\_8 = (2187.k_1 + 1217)/ 8192$	6813 + 2^13. n
609	1 -> 7 -> 4 -> 6 -> 5 -> 2 -> 3 -> 1	7	14	$k\_8 = (2187.k_1 - 279) / 16384$	12069 + 2^14. n
610	1 -> 7 -> 5 -> 2 -> 3 -> 8 -> 4 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 5111)/32768$	18793 + 2^15. n
611	1 -> 7 -> 5 -> 6 -> 1	4	8	$k\_5 = (81.k_1 + 7) / 256$	41 + 2^8. n
612	1 -> 7 -> 5 -> 8 -> 4 -> 2 -> 3 -> 1	7	14	$k\_8 = (2187.k_1 - 99) / 16384$	2697 + 2^14. n
613	1 -> 7 -> 5 -> 8 -> 4 -> 2 -> 3 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 1751)/32768$	649 + 2^15. n
614	1 -> 7 -> 6 -> 1	3	6	$k\_4 = (27.k_1 + 1) / 64$	45 + 2^6. n
615	1 -> 7 -> 6 -> 5 -> 4 -> 2 -> 3 -> 1	7	14	$k\_8 = (2187.k_1 - 207)/ 16384$	11597 + 2^14. n
616	1 -> 7 -> 8 -> 4 -> 2 -> 3 -> 1	6	12	$k\_7 = (729.k_1 - 45) / 4096$	2933 + 2^12. n
617	1 -> 7 -> 8 -> 4 -> 2 -> 3 -> 5 -> 6 -> 1	8	15	$k\_9 = (6561.k_1 + 1643)/32768$	25461 + 2^15. n
618	1 -> 7 -> 8 -> 4 -> 2 -> 3 -> 6 -> 1	7	13	$k\_8 = (2187.k_1 + 377) / 8192$	1397 + 2^13. n
619	2 -> 3 -> 1 -> 7 -> 6 -> 5 -> 8 -> 4 -> 2	8	15	$k\_9 = (6561.k_1 + 1888)/32768$	29856 + 2^15. n
620	2 -> 3 -> 1 -> 7 -> 8 -> 4 -> 6 -> 5 -> 2	8	15	$k\_9 = (6561.k_1 + 608)/ 32768$	22944 + 2^15. n
621	2 -> 3 -> 4 -> 6 -> 5 -> 2	5	9	$k\_6 = (243.k_1 - 4) / 512$	236 + 2^9. n

622	2 -> 3 -> 5 -> 6 -> 1 -> 7 -> 8 -> 4 -> 2	8	15	k_9 = (6561.k_1 + 16) / 32768	23024 + 2^15. n
623	2 -> 3 -> 5 -> 8 -> 4 -> 2	5	9	k_6 = (243.k_1 + 16) / 512	80 + 2^9. n
624	2 -> 3 -> 5 -> 8 -> 4 -> 6 -> 1 -> 7 -> 2	8	15	k_9 = (6561.k_1 - 2384)/32768	10064 + 2^15. n
625	2 -> 3 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2	7	12	k_8 = (2187.k_1 + 452) / 4096	1588 + 2^12. n
626	2 -> 3 -> 6 -> 1 -> 7 -> 5 -> 8 -> 4 -> 2	8	15	k_9 = (6561.k_1 + 268)/ 32768	8820 + 2^15. n
627	2 -> 3 -> 6 -> 1 -> 7 -> 8 -> 4 -> 2	7	13	k_8 = (2187.k_1 + 4) / 8192	884 + 2^13. n
628	2 -> 3 -> 6 -> 5 -> 4 -> 2	5	9	k_6 = (243.k_1 + 4) / 512	276 + 2^9. n
629	2 -> 3 -> 7 -> 6 -> 5 -> 8 -> 4 -> 2	7	13	k_8 = (2187.k_1 + 472) / 8192	6008 + 2^13. n
630	2 -> 3 -> 7 -> 8 -> 4 -> 6 -> 5 -> 2	7	13	k_8 = (218.k_1 + 152) / 8192	824 + 2^13. n
631	2 -> 3 -> 8 -> 4 -> 2	4	7	k_5 = (81.k_1 + 4) / 128	60 + 128. n
632	2 -> 3 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2	7	12	k_8 = (2187.k_1 + 172) / 4096	1148 + 2^12. n
633	2 -> 3 -> 8 -> 4 -> 6 -> 1 -> 7 -> 2	7	13	k_8 = (2187.k_1 - 596) / 8192	7548 + 2^13. n
634	2 -> 3 -> 8 -> 4 -> 6 -> 1 -> 7 -> 5 -> 2	8	15	k_9 = (6561.k_1 - 2812)/32768	380 + 2^15. n
635	2 -> 7 -> 1 -> 5 -> 4 -> 2	5	9	k_6 = (243.k_1 - 5) / 512	295 + 2^9. n
636	2 -> 7 -> 2	2	4	k_3 = (9.k_1 - 1) / 16	9 + 2^4. n
637	2 -> 7 -> 5 -> 2	3	6	k_4 = (27.k_1 - 5) / 64	31 + 2^6. n
638	2 -> 7 -> 5 -> 8 -> 4 -> 6 -> 1 -> 3 -> 2	8	15	k_9 = (6561.k_1 - 3535)/32768	20847 + 2^15. n
639	2 -> 7 -> 6 -> 1 -> 5 -> 2	5	9	k_6 = (243.k_1 - 19) / 512	97 + 2^9. n
640	2 -> 7 -> 8 -> 4 -> 6 -> 1 -> 3 -> 2	7	13	k_8 = (2187.k_1 - 823) / 8192	6533 + 2^13. n
641	3 -> 1 -> 7 -> 6 -> 5 -> 8 -> 4 -> 2 -> 3	8	15	k_9 = (6561.k_1 + 2832)/32768	12016 + 2^15. n
642	3 -> 1 -> 7 -> 8 -> 4 -> 6 -> 5 -> 2 -> 3	8	15	k_9 = (6561.k_1 + 912)/ 32768	1648 + 2^15. n
643	3 -> 2 -> 7 -> 5 -> 8 -> 4 -> 6 -> 1 -> 3	8	15	k_9 = (6561.k_1 - 1382)/32768	1958 + 2^15. n
644	3 -> 2 -> 7 -> 8 -> 4 -> 6 -> 1 -> 3	7	13	k_8 = (2187.k_1 - 386) / 8192	4806 + 2^13. n
645	3 -> 4 -> 6 -> 5 -> 2 -> 3	5	9	k_6 = (243.k_1 - 6) / 512	354 + 2^9. n
646	3 -> 5 -> 2 -> 7 -> 8 -> 4 -> 6 -> 1 -> 3	8	15	k_9 = (6561.k_1 - 1544)/32768	6408 + 2^15. n
647	3 -> 5 -> 6 -> 1 -> 7 -> 8 -> 4 -> 2 -> 3	8	15	k_9 = (6561.k_1 + 24) / 32768	1768 + 2^15. n
648	3 -> 5 -> 8 -> 4 -> 2 -> 3	5	9	k_6 = (243.k_1 + 24) / 512	120 + 2^9
649	3 -> 5 -> 8 -> 4 -> 2 -> 7 -> 6 -> 1 -> 3	8	15	k_9 = (6561.k_1 - 3704)/32768	11128 + 2^15. n
650	3 -> 5 -> 8 -> 4 -> 6 -> 1 -> 3	6	11	k_7 = (729.k_1 - 56) / 2048	1784 + 2^11. n
651	3 -> 5 -> 8 -> 4 -> 6 -> 1 -> 7 -> 2 -> 3	8	15	k_9 = (6561.k_1 - 3576)/32768	31480 + 2^15. n
652	3 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 3	7	12	k_8 = (2187.k_1 + 678) / 4096	2382 + 2^12. n
653	3 -> 6 -> 1 -> 7 -> 5 -> 8 -> 4 -> 2 -> 3	8	15	k_9 = (6561.k_1 + 402)/ 32768	13230 + 2^15. n
654	3 -> 6 -> 1 -> 7 -> 8 -> 4 -> 2 -> 3	7	13	k_8 = (2187.k_1 + 6) / 8192	1326 + 2^13. n
655	3 -> 6 -> 5 -> 4 -> 2 -> 3	5	9	k_6 = (243.k_1 + 6) / 512	158 + 2^9. n
656	3 -> 6 -> 5 -> 8 -> 4 -> 2 -> 7 -> 1 -> 3	8	15	k_9 = (6561.k_1 - 4766)/32768	14814 + 2^15. n
657	3 -> 6 -> 5 -> 8 -> 4 -> 2 -> 7 -> 3	7	13	k_8 = (2187.k_1 - 906) / 8192	4574 + 2^13. n
658	3 -> 7 -> 6 -> 5 -> 8 -> 4 -> 2 -> 3	7	13	k_8 = (2187.k_1 + 708) / 8192	820 + 2^13. n
659	3 -> 7 -> 8 -> 4 -> 6 -> 5 -> 2 -> 3	7	13	k_8 = (2187.k_1 + 228) / 8192	1236 + 2^13. n
660	3 -> 8 -> 4 -> 2 -> 3	4	7	k_5 = (81.k_1 + 6) / 128	90 + 2^7. n
661	3 -> 8 -> 4 -> 2 -> 7 -> 5 -> 6 -> 1 -> 3	8	15	k_9 = (6561.k_1 - 4442)/32768	5914 + 2^15. n
662	3 -> 8 -> 4 -> 2 -> 7 -> 6 -> 1 -> 3	7	13	k_8 = (2187.k_1 - 926) / 8192	154 + 2^13. n
663	3 -> 8 -> 4 -> 6 -> 1 -> 3	5	9	k_6 = (243.k_1 - 14) / 512	314 + 2^9. n
664	3 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 3	7	12	k_8 = (2187.k_1 + 258) / 4096	3770 + 2^12. n
665	3 -> 8 -> 4 -> 6 -> 1 -> 7 -> 2 -> 3	7	13	k_8 = (2187.k_1 - 894) / 8192	7226 + 2^13. n
666	3 -> 8 -> 4 -> 6 -> 1 -> 7 -> 5 -> 2 -> 3	8	15	k_9 = (6561.k_1 - 94218)/32768	570 + 2^15. n
667	3 -> 8 -> 4 -> 6 -> 5 -> 2 -> 7 -> 1 -> 3	8	15	k_9 = (6561.k_1 - 5306)/32768	7802 + 2^15. n
668	3 -> 8 -> 4 -> 6 -> 5 -> 2 -> 7 -> 3	7	13	k_8 = (2187.k_1 - 1086) / 8192	5754 + 2^13. n
669	4 -> 2 -> 3 -> 1 -> 7 -> 6 -> 5 -> 8 -> 4	8	15	k_9 = (6561.k_1 - 7477)/32768	8981 + 2^15. n
670	4 -> 2 -> 3 -> 4	3	6	k_4 = (27.k_1 - 15) / 64	29 + 2^6. n

671	4 -> 2 -> 3 -> 5 -> 6 -> 1 -> 7 -> 8 -> 4	8	15	k_9 = (6561.k_1 - 8725)/32768	15349 + 2^15. n
672	4 -> 2 -> 3 -> 5 -> 8 -> 4	5	9	k_6 = (243.k_1 - 79) / 512	53 + 2^9. n
673	4 -> 2 -> 3 -> 6 -> 1 -> 5 -> 8 -> 4	7	12	k_8 = (2187.k_1 - 335) / 4096	3789 + 2^12. n
674	4 -> 2 -> 3 -> 6 -> 1 -> 7 -> 5 -> 8 -> 4	8	15	k_9 = (6561.k_1 - 8557)/32768	27725 + 2^15. n
675	4 -> 2 -> 3 -> 6 -> 1 -> 7 -> 8 -> 4	7	13	k_8 = (2187*k_1 - 1999) / 8192	589 + 2^13. n
676	4 -> 2 -> 3 -> 6 -> 5 -> 4	5	9	k_6 = (243.k_1 - 87) / 512	13 + 2^9. n
677	4 -> 2 -> 3 -> 7 -> 6 -> 5 -> 8 -> 4	7	13	k_9 = (2187.k_1 - 1687) / 8192	4005 + 2^13. n
678	4 -> 2 -> 3 -> 8 -> 4	4	7	k_5 = (81.k_1 - 13) / 128	125 + 2^7. n
679	4 -> 2 -> 7 -> 1 -> 5 -> 4	5	9	k_6 = (243.k_1 - 93) / 512	367 + 2^9. n
680	4 -> 6 -> 1 -> 5 -> 2 -> 3 -> 4	6	11	k_7 = (729k_1 - 516) / 2048	1956 + 2^11. n
681	4 -> 6 -> 1 -> 5 -> 2 -> 3 -> 8 -> 4	7	12	k_8 = (2187.k_1 - 524) / 4096	2980 + 2^12. n
682	4 -> 6 -> 1 -> 7 -> 2 -> 3 -> 4	6	12	k_7 = (729.k_1 - 1392) / 4096	2800 + 2^12. n
683	4 -> 6 -> 1 -> 7 -> 2 -> 3 -> 5 -> 8 -> 4	8	15	k_9 = (6561.k_1 - 8944)/32768	7408 + 2^15. n
684	4 -> 6 -> 1 -> 7 -> 2 -> 3 -> 8 -> 4	7	13	k_8 = (2187.k_1 - 2128) / 8192	4848 + 2^13. n
685	4 -> 6 -> 1 -> 7 -> 5 -> 2 -> 3 -> 4	7	14	k_8 = (2187.k_1 - 5856)/16384	160 + 2^14. n
686	4 -> 6 -> 1 -> 7 -> 5 -> 2 -> 3 -> 8 -> 4	8	15	k_9 = (6561.k_1 - 9376)/32768	8352 + 2^15. n
687	4 -> 6 -> 5 -> 2 -> 3 -> 1 -> 7 -> 8 -> 4	8	15	k_9 = (6561.k_1 - 9346)/32768	26946 + 2^15. n
688	4 -> 6 -> 5 -> 2 -> 3 -> 7 -> 8 -> 4	7	13	k_8 = (2187.k_1 - 2134) / 8192	3522 + 2^13. n
689	5 -> 2 -> 3 -> 1 -> 7 -> 8 -> 4 -> 6 -> 5	8	15	k_9 = (6561.k_1 - 7925)/32768	19669 + 2^15. n
690	5 -> 2 -> 3 -> 4 -> 6 -> 1 -> 5	6	11	k_7 = (729.k_1 - 541) / 2048	1765 + 2^11. n
691	5 -> 2 -> 3 -> 4 -> 6 -> 5	5	9	k_6 = (243.k_1 - 95) / 512	485 + 2^9. n
692	5 -> 2 -> 3 -> 7 -> 8 -> 4 -> 6 -> 1 -> 5	8	15	k_9 = (6561.k_1 - 9493)32768	24309 + 2^15. n
693	5 -> 2 -> 3 -> 7 -> 8 -> 4 -> 6 -> 5	7	13	k_8 = (2187k_1 - 1799) / 8192	3829 + 2^13. n
694	5 -> 2 -> 3 -> 8 -> 4 -> 6 -> 1 -> 7 -> 5	8	15	k_9 = (6561.k_1 - 12485)/32768	11429 + 2^15. n
695	5 -> 2 -> 7 -> 1 -> 5	4	8	k_5 = (81.k_1 - 89) / 256	137 + 2^8. n
696	5 -> 2 -> 7 -> 5	3	6	k_4 = (27.k_1 - 19) / 64	41 + 2^6. n
697	5 -> 2 -> 7 -> 8 -> 4 -> 6 -> 1 -> 3 -> 5	8	15	k_9 = (6561.k_1 - 13393)/32768	31921 + 2^15. n
698	5 -> 4 -> 2 -> 3 -> 5	4	8	k_5 = (81.k_1 - 92) / 256	156 + 2^8. n
699	5 -> 4 -> 2 -> 3 -> 6 -> 1 -> 5	6	11	k_8 = (729.k_1 - 604) / 2048	956 + 2^11. n
700	5 -> 4 -> 2 -> 3 -> 6 -> 5	5	9	k_6 = (243.k_1 - 116) / 512	188 + 2^9. n
701	5 -> 4 -> 6 -> 1 -> 7 -> 2 -> 3 -> 5	7	14	k_8 = (2187.k_1 - 7616)/16384	4416 + 2^14. n
702	5 -> 6 -> 1 -> 7 -> 4 -> 2 -> 3 -> 5	7	14	k_8 = (2187.k_1 - 7481)/16384	9675 + 2^14. n
703	5 -> 6 -> 1 -> 7 -> 5	4	8	k_5 = (81.k_1 - 91) / 256	235 + 2^8. n
704	5 -> 8 -> 4 -> 2 -> 3 -> 1 -> 5	6	11	k_7 = (729.k_1 - 646) / 2048	1782 + 2^11. n
705	5 -> 8 -> 4 -> 2 -> 3 -> 1 -> 7 -> 6 -> 5	8	15	k_9 = (6561k_1 - 12470)/32768	4342 + 2^15. n
706	5 -> 8 -> 4 -> 2 -> 3 -> 6 -> 1 -> 7 -> 5	8	15	k_9 = (6561.k_1 - 13430)/32768	31926 + 2^15. n
707	5 -> 8 -> 4 -> 2 -> 3 -> 7 -> 6 -> 1 -> 5	8	15	k_9 = (6561.k_1 - 12598)/32768	16758 + 2^15. n
708	5 -> 8 -> 4 -> 2 -> 3 -> 7 -> 6 -> 5	7	13	k_8 = (2187k_1- 2834) / 8192	4470 + 2^13. n
709	5 -> 8 -> 4 -> 2 -> 7 -> 6 -> 1 -> 3 -> 5	8	15	k_9 = (6561.k_1 - 13798)/32768	26662 + 2^15. n
710	5 -> 8 -> 4 -> 6 -> 1 -> 3 -> 5	6	11	k_7 = (729.k_1 - 670) / 2048	206 + 2^11. n
711	6 -> 1 -> 5 -> 2 -> 3 -> 8 -> 4 -> 6	7	12	k_8 = (2187.k_1 - 786) / 4096	2422 + 2^12. n
712	6 -> 1 -> 5 -> 2 -> 7 -> 6	5	9	k_6 = (243.k_1 - 162) / 512	342 + 2^9. n
713	6 -> 1 -> 5 -> 4 -> 2 -> 3 -> 6	6	11	k_7 = (729.k_1 - 830) / 2048	622 + 2^11. n
714	6 -> 1 -> 5 -> 6	3	5	k_4 = (27.k_1 - 2) / 32	6 + 2^5. n
715	6 -> 1 -> 7 -> 2 -> 3 -> 5 -> 8 -> 4 -> 6	8	15	k_9 = (6561.k_1 - 13416)/32768	27496 + 2^15. n
716	6 -> 1 -> 7 -> 2 -> 3 -> 8 -> 4 -> 6	7	13	k_8 = (2187.k_1 - 3192) / 8192	7272 + 2^13. n
717	6 -> 1 -> 7 -> 4 -> 2 -> 3 -> 6	6	11	k_7 = (729.k_1 - 2200) / 4096	3672 + 2^11. n
718	6 -> 1 -> 7 -> 5 -> 2 -> 3 -> 8 -> 4 -> 6	8	15	k_9 = (6561.k_1 - 14064)/32768	12528 + 2^15. n
719	6 -> 1 -> 7 -> 5 -> 4 -> 2 -> 3 -> 6	7	14	k_8 = (2187.k_1 - 9232)/16384	7728 + 2^14. n

720	6 -> 1 -> 7 -> 5 -> 6	4	8	k_5 = (81.k_1 - 112)/256	112 + 2^8. n
721	6 -> 1 -> 7 -> 6	3	6	k_4 = (27.k_1 - 24) / 64	8 + 2^6. n
722	6 -> 1 -> 7 -> 8 -> 4 -> 2 -> 3 -> 5 -> 6	8	15	k_9 = (6561.k_1 - 16376)/32768	27896 + 2^15. n
723	6 -> 5 -> 2 -> 3 -> 1 -> 7 -> 8 -> 4 -> 6	8	15	k_9 = (6561.k_1 - 14019)/32768	24035 + 2^15. n
724	6 -> 5 -> 2 -> 3 -> 4 -> 6	5	9	k_6 = (243.k_1 - 153) / 512	323 + 2^9. n
725	6 -> 5 -> 2 -> 3 -> 6	4	8	k_5 = (81.k_1 - 115) / 256	131 + 2^8. n
726	6 -> 5 -> 2 -> 3 -> 7 -> 8 -> 4 -> 6	7	13	k_8 = (2187k_1 - 3201) / 8192	5283 + 2^13. n
727	6 -> 5 -> 8 -> 4 -> 2 -> 3 -> 1 -> 7 -> 6	8	15	k_9 = (6561.k_1 - 17049)/32768	13817 + 2^15. n
728	6 -> 5 -> 8 -> 4 -> 2 -> 3 -> 7 -> 6	7	13	k_8 = (2187.k_1 - 3891) / 8192	249 + 2^13. n
729	6 -> 5 -> 8 -> 4 -> 2 -> 7 -> 1 -> 3 -> 6	8	15	k_9 = (6561.k_1 - 17273)/32768	19161 + 2^15. n
730	6 -> 5 -> 8 -> 4 -> 2 -> 7 -> 3 -> 6	7	13	k_8 = (2187k_1 - 3923) / 8192	1369 + 2^13. n
731	7 -> 1 -> 5 -> 4 -> 2 -> 7	5	9	k_6 = (243.k_1 - 142) / 512	186 + 2^9. n
732	7 -> 2 -> 3 -> 5 -> 8 -> 4 -> 6 -> 1 -> 7	8	15	k_9 = (6561.k_1 - 15093)/32768	26837 + 2^15. n
733	7 -> 2 -> 3 -> 8 -> 4 -> 6 -> 1 -> 7	7	13	k_8 = (2187.k_1 - 3591) / 8192	1013 + 2^13. n
734	7 -> 2 -> 7	2	4	k_3 = (9.k_1 - 5) / 16	13 + 2^4. n
735	7 -> 4 -> 6 -> 5 -> 2 -> 3 -> 7	6	12	k_7 = (729.k_1 - 2595) / 4096	3931 + 2^12. n
736	7 -> 5 -> 2 -> 3 -> 8 -> 4 -> 6 -> 1 -> 7	8	15	k_9 = (6561.k_1 - 15822)/32768	30478 + 2^15. n
737	7 -> 5 -> 2 -> 7	3	6	k_4 = (27.k_1 - 26) / 64	46 + 2^6. n
738	7 -> 5 -> 6 -> 1 -> 7	4	8	k_5 = (81.k_1 - 126) / 256	30 + 2^8. n
739	7 -> 5 -> 8 -> 4 -> 2 -> 3 -> 6 -> 1 -> 7	8	15	k_9 = (6561.k_1 - 18342)/32768	8678 + 2^15. n
740	7 -> 5 -> 8 -> 4 -> 2 -> 3 -> 7	6	12	k_7 = (729.k_1 - 2550) / 4096	998 + 2^12. n
741	7 -> 6 -> 1 -> 5 -> 2 -> 7	5	9	k_6 = (243.k_1 - 163) / 512	401 + 2^9. n
742	7 -> 6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 3 -> 7	8	15	k_9 = (6561.k_1 - 19185)/32768	20305 + 2^15. n
743	7 -> 6 -> 1 -> 7	3	6	k_4 = (27.k_1 - 27) / 64	1 + 2^6. n
744	7 -> 6 -> 5 -> 4 -> 2 -> 3 -> 7	6	12	k_7 = (729.k_1 - 2577) / 4096	3577 + 2^12. n
745	7 -> 6 -> 5 -> 8 -> 4 -> 2 -> 3 -> 1 -> 7	8	15	k_9 = (6561.k_1 - 19257)/32768	15001 + 2^15. n
746	7 -> 8 -> 4 -> 2 -> 3 -> 5 -> 6 -> 1 -> 7	8	15	k_9 = (6561.k_1 - 18423)/32768	27287 + 2^15. n
747	7 -> 8 -> 4 -> 2 -> 3 -> 6 -> 1 -> 7	7	12	k_8 = (2187.k_1 - 4221) / 8192	1047 + 2^12. n
748	7 -> 8 -> 4 -> 2 -> 3 -> 7	5	10	k_6 = (243.k_1 - 597) / 1024	407 + 2^10. n
749	7 -> 8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 3 -> 7	8	15	k_9 = (6561.k_1 - 19815)/32768	23047 + 2^15. n
750	7 -> 8 -> 4 -> 6 -> 5 -> 2 -> 3 -> 1 -> 7	8	15	k_9 = (6561.k_1 - 19527)/32768	11495 + 2^15. n
751	8 -> 4 -> 2 -> 3 -> 1 -> 7 -> 6 -> 5 -> 8	8	15	k_9 = (6561.k_1 - 22456)/32768	27832 + 2^15. n
752	8 -> 4 -> 2 -> 3 -> 5 -> 6 -> 1 -> 7 -> 8	8	15	k_9 = (6561.k_1 - 23288)/32768	10232 + 2^15. n
753	8 -> 4 -> 2 -> 3 -> 5 -> 8	5	9	k_6 = (243.k_1 - 232) / 512	376 + 2^9. n
754	8 -> 4 -> 2 -> 3 -> 6 -> 1 -> 5 -> 8	7	12	k_8 = (2187.k_1 - 1496)/ 4096	1160 + 2^12. n
755	8 -> 4 -> 2 -> 3 -> 6 -> 1 -> 7 -> 5 -> 8	8	15	k_9 = (6561.k_1 - 23176)/32768	7560 + 2^15. n
756	8 -> 4 -> 2 -> 3 -> 6 -> 1 -> 7 -> 8	7	13	k_8 = (2187.k_1 - 5336) / 8192	392 + 2^13. n
757	8 -> 4 -> 2 -> 3 -> 7 -> 6 -> 5 -> 8	7	13	k_8 = (2187.k_1 - 5128) / 8192	5400 + 2^13. n
758	8 -> 4 -> 2 -> 7 -> 5 -> 6 -> 1 -> 3 -> 8	8	15	k_9 = (6561.k_1 - 23764)/32768	29780 + 2^15. n
759	8 -> 4 -> 2 -> 7 -> 6 -> 1 -> 3 -> 8	7	13	k_8 = (2187.k_1 - 5428) / 8192	4636 + 2^13. n
760	8 -> 4 -> 6 -> 1 -> 3 -> 8	5	9	k_6 = (243.k_1 - 238) / 512	218 + 2^9. n
761	8 -> 4 -> 6 -> 1 -> 7 -> 2 -> 3 -> 5 -> 8	8	15	k_9 = (6561k_1 - 23434)/32768	4938 + 2^15. n
762	8 -> 4 -> 6 -> 5 -> 2 -> 3 -> 1 -> 7 -> 8	8	15	k_9 = (6561.k_1 - 23702)/32768	28886 + 2^15. n
763	8 -> 4 -> 6 -> 5 -> 2 -> 3 -> 7 -> 8	7	13	k_8 = (2187.k_1 - 5426) / 8192	5078 + 2^13. n
764	8 -> 4 -> 6 -> 5 -> 2 -> 7 -> 1 -> 3 -> 8	8	15	k_9 = (6561.k_1 - 23926)/32768	1462 + 2^15. n
765	8 -> 4 -> 6 -> 5 -> 2 -> 7 -> 3 -> 8	7	13	k_8 = (2187.k_1 - 5458) / 8192	6198 + 2^13. n
766	1 -> 5 -> 8 -> 4 -> 2 -> 7 -> 1	6	10	k_7 = (729.k_1 + 82) / 1024	94 + 2^10. n
767	1 -> 7 -> 4 -> 6 -> 1	4	7	k_5 = (81.k_1 + 11) / 128	101 + 2^7. n
768	1 -> 7 -> 5 -> 4 -> 6 -> 1	5	9	k_6 = (243.k_1 + 53) / 512	457 + 2^9. n

769	1 -> 7 -> 6 -> 5 -> 8 -> 4 -> 2 -> 3 -> 1	8	15	k_9 = (6561.k_1 + 531)/ 32768	30925 + 2^15. n
770	1 -> 7 -> 8 -> 4 -> 6 -> 5 -> 2 -> 3 -> 1	8	15	k_9 = (6561.k_1 + 171)/ 32768	4405 + 2^15. n
771	2 -> 3 -> 6 -> 5 -> 8 -> 4 -> 2	6	10	k_7 = (729.k_1 + 140) / 1024	660 + 2^10. n
772	2 -> 3 -> 8 -> 4 -> 6 -> 5 -> 2	6	10	k_7 = (729.k_1 + 100) / 1024	764 + 2^10. n
773	2 -> 7 -> 1 -> 5 -> 8 -> 4 -> 2	6	10	k_7 = (729.k_1 + 113) / 1024	167 + 2^10. n
774	2 -> 7 -> 4 -> 2	3	5	k_4 = (27.k_1 + 1) / 32	13 + 2^5. n
775	2 -> 7 -> 4 -> 6 -> 1 -> 5 -> 2	6	10	k_7 = (729.k_1 + 43) / 1024	861 + 2^10. n
776	2 -> 7 -> 5 -> 4 -> 2	4	7	k_5 = (81.k_1 + 1) / 128	79 + 2^7. n
777	2 -> 7 -> 6 -> 1 -> 5 -> 4 -> 2	6	10	k_7 = (729.k_1 + 71) / 1024	993 + 2^10. n
778	2 -> 7 -> 6 -> 5 -> 2	4	7	k_5 = (81.k_1 + -1) / 128	49 + 2^7. n
779	3 -> 6 -> 5 -> 8 -> 4 -> 2 -> 3	6	10	k_7 = (729.k_1 + 210) / 1024	990 + 2^10. n
780	3 -> 8 -> 4 -> 6 -> 5 -> 2 -> 3	6	10	k_7 = (729.k_1 + 150) / 1024	122 + 2^10. n
781	4 -> 2 -> 3 -> 6 -> 5 -> 8 -> 4	6	10	k_7 = (729.k_1 - 5) / 1024	781 + 2^10. n
782	4 -> 2 -> 7 -> 1 -> 5 -> 8 -> 4	6	10	k_7 = (729.k_1 - 23) / 1024	111 + 2^10. N
783	4 -> 2 -> 7 -> 4	3	5	k_4 = (27.k_1 - 1) / 32	19 + 2^5. n
784	4 -> 2 -> 7 -> 5 -> 4	4	7	k_5 = (81.k_1 - 15) / 128	95 + 2^7. n
785	4 -> 2 -> 7 -> 6 -> 1 -> 5 -> 4	6	10	k_7 = (729.k_1 - 51) / 1024	1003 + 2^10. n
786	4 -> 6 -> 1 -> 5 -> 2 -> 7 -> 4	6	10	k_7 = (729.k_1 - 68) / 1024	996 + 2^10. n
787	4 -> 6 -> 1 -> 7 -> 4	4	7	k_5 = (81.k_1 - 16) / 128	16 + 2^7. n
788	4 -> 6 -> 1 -> 7 -> 5 -> 4	5	9	k_6 = (243.k_1 - 96) / 512	32 + 2^9. n
789	4 -> 6 -> 5 -> 2 -> 3 -> 4	5	9	k_6 = (243.k_1 - 102) / 512	386 + 2^9. n
790	4 -> 6 -> 5 -> 2 -> 3 -> 8 -> 4	6	11	k_7 = (729.k_1 - 50) / 1024	642 + 2^11. n
791	5 -> 2 -> 3 -> 8 -> 4 -> 6 -> 1 -> 5	7	12	k_8 = (2187.k_1 - 407) / 4096	165 + 2^12. n
792	5 -> 2 -> 3 -> 8 -> 4 -> 6 -> 5	6	10	k_7 = (729.k_1 + 35) / 1024	677 + 2^10. n
793	5 -> 2 -> 7 -> 6 -> 1 -> 5	5	9	k_6 = (243.k_1 - 115) / 512	129 + 2^9. n
794	5 -> 2 -> 7 -> 6 -> 5	4	7	k_5 = (81.k_1 - 17) / 128	65 + 2^7. n
795	5 -> 4 -> 2 -> 7 -> 1 -> 5	5	9	k_6 = (243.k_1 - 124) / 512	148 + 2^9. n
796	5 -> 4 -> 2 -> 7 -> 5	4	7	k_5 = (81.k_1 - 20) / 128	84 + 2^7. n
797	5 -> 4 -> 6 -> 1 -> 7 -> 5	5	9	k_6 = (243.k_1 - 128) / 512	384 + 2^9. n
798	5 -> 6 -> 1 -> 5	3	5	k_4 = (27.k_1 - 1) / 32	19 + 2^5. n
799	5 -> 6 -> 1 -> 7 -> 8 -> 4 -> 2 -> 3 -> 5	8	15	k_9 = (6561.k_1 - 13099)/32768	4427 + 2^15. n
800	5 -> 8 -> 4 -> 2 -> 3 -> 5	5	9	k_f = (243.k_1 + -130) / 512	502 + 2^9. n
801	5 -> 8 -> 4 -> 2 -> 3 -> 6 -> 1 -> 5	7	12	k_8 = (2187.k_1 - 722) / 4096	182 + 2^12. n
802	5 -> 8 -> 4 -> 2 -> 3 -> 6 -> 5	6	10	k_7 = (729.k_1 - 70) / 1024	694 + 2^10. n
803	5 -> 8 -> 4 -> 6 -> 1 -> 7 -> 2 -> 3 -> 5	8	15	k_9 = (6561.k_1 - 13774)/32768	28430 + 2^15. n
804	6 -> 1 -> 5 -> 2 -> 7 -> 4 -> 6	6	10	k_7 = (729.k_1 - 102) / 1024	982 + 2^10. n
805	6 -> 1 -> 5 -> 4 -> 2 -> 7 -> 6	6	10	k_7 = (729.k_1 - 126) / 1024	430 + 2^10. n
806	6 -> 1 -> 5 -> 8 -> 4 -> 2 -> 3 -> 6	7	12	k_8 = (2187.k_1 - 1066) / 4096	1982 + 2^12. n
807	6 -> 1 -> 7 -> 4 -> 6	4	7	k_5 = (81.k_1 - 24) / 128	24 + 2^7. n
808	6 -> 1 -> 7 -> 5 -> 4 -> 6	5	9	k_6 = (243.k_1 - 144) / 512	304 + 2^9. n
809	6 -> 1 -> 7 -> 5 -> 8 -> 4 -> 2 -> 3 -> 6	8	15	k_9 = (6561.k_1 - 16304)/32768	432 + 2^15. n
810	6 -> 1 -> 7 -> 8 -> 4 -> 2 -> 3 -> 6	7	13	k_8 = (2187.k_1 - 3752) / 8192	6392 + 2^13. n
811	6 -> 5 -> 2 -> 3 -> 8 -> 4 -> 6	6	10	k_7 = (729.k_1 - 75) / 1024	451 + 2^10. n
812	6 -> 5 -> 2 -> 7 -> 6	4	7	k_5 = (81.k_1 - 27) / 128	43 + 2^7. n
813	6 -> 5 -> 4 -> 2 -> 3 -> 6	5	9	k_6 = (243.k_1 - 167) / 512	125 + 2^9. n
814	7 -> 1 -> 5 -> 8 -> 4 -> 2 -> 7	6	10	k_7 = (729.k_1 + 22) / 1024	250 + 2^10. n
815	7 -> 4 -> 2 -> 7	3	5	k_f = (27.k_1 - 1) / 32	19 + 2^5. n
816	7 -> 4 -> 6 -> 1 -> 5 -> 2 -> 7	6	10	k_7 = (729.k_1 - 83) / 1024	267 + 2^10. n
817	7 -> 4 -> 6 -> 1 -> 7	4	7	k_5 = (81.k_1 - 27) / 128	43 + 2^7. n

818	7 -> 5 -> 4 -> 2 -> 7	4	7	k_5 = (81.k_1 - 22) / 128	54 + 2^7. n
819	7 -> 5 -> 4 -> 6 -> 1 -> 7	5	9	k_6 = (243.k_1 - 162) / 512	342 + 2^9. n
820	7 -> 6 -> 1 -> 5 -> 4 -> 2 -> 7	6	10	k_7 = (729.k_1 - 41) / 1024	465 + 2^10. n
821	7 -> 6 -> 5 -> 2 -> 7	4	7	k_5 = (81.k_1 - 25) / 128	73 + 2^7. n
822	7 -> 6 -> 5 -> 8 -> 4 -> 2 -> 3 -> 7	7	13	k_8 = (2187k_1 - 4371) / 8192	665 + 2^13. n
823	7 -> 8 -> 4 -> 6 -> 5 -> 2 -> 3 -> 7	7	13	k_8 = (2187.k_1 - 4461) / 8192	5351 + 2^13. n
824	8 -> 4 -> 2 -> 3 -> 6 -> 5 -> 8	6	10	k_7 = (729.k_1 - 200) / 1024	520 + 2^10. n
825	8 -> 4 -> 2 -> 3 -> 8	4	7	k_5 = (81.k_1 - 40) / 128	40 + 2^7. n
826	8 -> 4 -> 2 -> 7 -> 1 -> 5 -> 8	6	10	k_7 = (729.k_1 - 212) / 1024	756 + 2^10. n
827	8 -> 4 -> 6 -> 1 -> 5 -> 2 -> 3 -> 8	7	12	k_8 = (2187.k_1 - 1622) / 4096	1986 + 2^12. n
828	8 -> 4 -> 6 -> 1 -> 7 -> 2 -> 3 -> 8	7	13	k_8 = (2187.k_1 - 5422) / 8192	5962 + 2^13. n
829	8 -> 4 -> 6 -> 1 -> 7 -> 5 -> 2 -> 3 -> 8	8	15	k_9 = (6561.k_1 - 23722)/32768	16490 + 2^15. n
830	1 -> 7 -> 5 -> 8 -> 4 -> 6 -> 1	6	10	k_7 = (729.k_1 + 223) / 1024	393 + 2^10. n
831	1 -> 7 -> 8 -> 4 -> 6 -> 1	5	8	k_6 = (243.k_1 + 49) / 256	181 + 2^8. n
832	2 -> 7 -> 4 -> 6 -> 5 -> 2	5	8	k_6 = (243.k_1 + 25) / 256	61 + 2^8. n
833	2 -> 7 -> 5 -> 8 -> 4 -> 2	5	8	k_6 = (243.k_1 + 35) / 256	239 + 2^8. n
834	2 -> 7 -> 6 -> 5 -> 4 -> 2	5	8	k_6 = (243.k_1 + 29) / 256	81 + 2^8. n
835	4 -> 2 -> 7 -> 5 -> 8 -> 4	5	8	k_6 = (243.k_1 + 19) / 256	159 + 2^8. n
836	4 -> 2 -> 7 -> 6 -> 5 -> 4	5	8	k_6 = (243.k_1 + 15) / 256	139 + 2^8. n
837	4 -> 6 -> 1 -> 7 -> 5 -> 8 -> 4	6	10	k_7 = (729.k_1 - 32) / 1024	288 + 2^10. n
838	4 -> 6 -> 1 -> 7 -> 8 -> 4	5	8	k_6 = (243.k_1 + 16) / 256	80 + 2^8. n
839	4 -> 6 -> 5 -> 2 -> 7 -> 4	5	8	k_6 = (243.k_1 + 10) / 256	178 + 2^8. n
840	5 -> 2 -> 7 -> 4 -> 6 -> 1 -> 5	6	10	k_7 = (729.k_1 - 41) / 1024	465 + 2^10. n
841	5 -> 2 -> 7 -> 4 -> 6 -> 5	5	8	k_6 = (243.k_1 + 29) / 256	81 + 2^8. n
842	5 -> 4 -> 2 -> 7 -> 6 -> 1 -> 5	6	10	k_7 = (729.k_1 - 68) / 1024	996 + 2^10. n
843	5 -> 4 -> 2 -> 7 -> 6 -> 5	5	8	k_6 = (243.k_1 + 20) / 256	100 + 2^8. n
844	5 -> 8 -> 4 -> 2 -> 7 -> 1 -> 5	6	10	k_7 = (729.k_1 - 86) / 1024	326 + 2^10. n
845	5 -> 8 -> 4 -> 2 -> 7 -> 5	5	8	k_6 = (243.k_1 + 14) / 256	198 + 2^8. n
846	5 -> 8 -> 4 -> 6 -> 1 -> 7 -> 5	6	10	k_7 = (729.k_1 - 94) / 1024	142 + 2^10. n
847	6 -> 1 -> 7 -> 5 -> 8 -> 4 -> 6	6	10	k_7 = (729.k_1 - 48) / 1024	944 + 2^10. n
848	6 -> 1 -> 7 -> 8 -> 4 -> 6	5	8	k_6 = (243.k_1 + 24) / 256	120 + 2^8. n
849	6 -> 5 -> 2 -> 7 -> 4 -> 6	5	8	k_6 = (243.k_1 + 15) / 256	139 + 2^8. n
850	6 -> 5 -> 4 -> 2 -> 7 -> 6	5	8	k_6 = (243.k_1 + 9) / 256	237 + 2^8. n
851	6 -> 5 -> 8 -> 4 -> 2 -> 3 -> 6	6	10	k_7 = (729.k_1 - 145) / 1024	121 + 2^10. n
852	7 -> 4 -> 6 -> 5 -> 2 -> 7	5	8	k_6 = (243.k_1 + 31) / 256	219 + 2^8. n
853	7 -> 5 -> 8 -> 4 -> 2 -> 7	5	8	k_6 = (243.k_1 + 46) / 256	102 + 2^8. n
854	7 -> 5 -> 8 -> 4 -> 6 -> 1 -> 7	6	10	k_7 = (729.k_1 - 54) / 1024	38 + 2^10. n
855	7 -> 6 -> 5 -> 4 -> 2 -> 7	5	8	k_6 = (243.k_1 + 37) / 256	121 + 2^8. n
856	7 -> 8 -> 4 -> 6 -> 1 -> 7	5	8	k_6 = (243.k_1 + 27) / 256	199 + 2^8. n
857	8 -> 4 -> 2 -> 7 -> 5 -> 8	5	8	k_6 = (243.k_1 + 4) / 256	20 + 2^8. n
858	8 -> 4 -> 6 -> 1 -> 7 -> 5 -> 8	6	10	k_7 = (729k_1 - 218) / 1024	874 + 2^10. n
859	8 -> 4 -> 6 -> 1 -> 7 -> 8	5	8	k_6 = (243.k_1 + 2) / 256	138 + 2^8. n
860	8 -> 4 -> 6 -> 5 -> 2 -> 3 -> 8	6	10	k_7 = (729.k_1 - 230) / 1024	86 + 2^10. n
861	1-> 5 ->2 ->7 ->4 ->6 -> 1	6	10	k_7 = (729.k_1 + 142) / 1024	962 + 2^10. n
862	1 -> 5 -> 4 -> 2 -> 7 -> 6 -> 1	6	10	k_7 = (729.k_1 + 106) / 1024	646 + 2^10. n

Appendix E: Mod 16 Brute Force Checker:

Code begins:

```
#include <bits/stdc++.h>
#include <zlib.h>
using namespace std;

// Global constants
static const int N = 16;
static const int n_eq = 16;

// ---------- Utility: Write a permutation line ----------
void write_gz_line(gzFile gz, const vector<int> &perm, long long R_num, long long R_den, int special_exp = -1) {
  ostringstream oss;
  if (special_exp >= 0) {
    oss << "special=" << special_exp << " : ";
  }
  for (int i = 0; i < (int)perm.size(); i++) {
    oss << perm[i];
    if (i != (int)perm.size() - 1) oss << " ";
  }
  oss << " | R = ";
  if (R_den != 0 && R_num % R_den == 0) {
    oss << (R_num / R_den);
  } else {
    oss << R_num << "/" << R_den;
  }
  oss << "\n";
  gzputs(gz, oss.str().c_str());
}

// ---------- Compute N_val ----------
long long compute_N(const vector<int> &exps) {
  long long N_val = 0;
  int running_exp = 0;
  for (int i = 0; i < (int)exps.size(); i++) {
    if (i > 0) running_exp += exps[i - 1];
    N_val += (long long)pow(3, n_eq - 1 - i) * (1LL << running_exp);
  }
  return N_val;
}

// ---------- Part A: Original 10.8M permutations ----------
void run_original_permutations() {
  vector<int> exps = {1,1,1,1,1,1,1,1, 2,2,2,2, 3,3, 4, 5};
  gzFile gz = gzopen("original_10810800.csv.gz", "wb");
  if (!gz) {
    cerr << "Error opening output file for original dataset.\n";
    return;
  }

  sort(exps.begin(), exps.end());
  long long count = 0;

  do {
    long long S = accumulate(exps.begin(), exps.end(), 0LL);
```

```cpp
        long long D = (1LL << S) - (long long)pow(3, n_eq);
        long long N_val = compute_N(exps);

        write_gz_line(gz, exps, N_val, D);
        count++;
        if (count % 1000000 == 0) cerr << "[Original] Generated " << count << " permutations\n";
    } while (next_permutation(exps.begin(), exps.end()));

    gzclose(gz);
    cerr << "[Original] Finished: " << count << " permutations.\n";
}

// ---------- Part B: Variable special exponent ----------
void run_variable_special_exponent() {
    vector<int> base_exps = {1,1,1,1,1,1,1,1, 2,2,2,2, 3,3, 4};
    gzFile gz = gzopen("variable_special_exp.csv.gz", "wb");
    if (!gz) {
        cerr << "Error opening output file for variable special exp dataset.\n";
        return;
    }

    for (int special_exp = 5; special_exp <= 64; ++special_exp) {
        vector<int> exps = base_exps;
        exps.push_back(special_exp);

        sort(exps.begin(), exps.end());
        bool found_R_gt1 = false;
        long long count = 0;

        do {
            long long S = accumulate(exps.begin(), exps.end(), 0LL);
            long long D = (1LL << S) - (long long)pow(3, n_eq);
            long long N_val = compute_N(exps);

            if (N_val > 0 && D > 0 && N_val % D == 0) {
                long long R = N_val / D;
                if (R > 1) found_R_gt1 = true;
                write_gz_line(gz, exps, N_val, D, special_exp);
            }
            count++;
        } while (next_permutation(exps.begin(), exps.end()));

        cerr << "[Variable] Special exp " << special_exp
             << " done. Found R>1? " << (found_R_gt1 ? "YES" : "NO") << "\n";
        if (!found_R_gt1) {
            cerr << "[Variable] Stopping: All R <= 1 from here on.\n";
            break;
        }
    }

    gzclose(gz);
}

// ---------- Main ----------
int main() {
```

```
    cerr << "=== Starting Composite Program ===\n";
    run_original_permutations();
    run_variable_special_exponent();
    cerr << "=== Finished Both Tasks ===\n";
    return 0;
}
```

(**Cod ends**)

Appendix F: Mod 32 Brute Force Checker:
Code Begins:

```
#include <bits/stdc++.h>
#include <zlib.h>
using namespace std;

// ===== PARAMETERS =====
static const int v = 6;             // modulus exponent (2^v = 64 here)
static const int N = 1 << (v-1);    // number of exponents = 2^(v-1)
static const int n_eq = N;          // loop length = N

// ===== Utility: write permutation line =====
void write_gz_line(gzFile gz, const vector<int> &perm, long long R_num, long long R_den, int special_exp = -1) {
    ostringstream oss;
    if (special_exp >= 0) {
        oss << "special=" << special_exp << " : ";
    }
    for (int i = 0; i < (int)perm.size(); i++) {
        oss << perm[i];
        if (i != (int)perm.size() - 1) oss << " ";
    }
    oss << " | R = ";
    if (R_den != 0 && R_num % R_den == 0) {
        oss << (R_num / R_den);
    } else {
        oss << R_num << "/" << R_den;
    }
    oss << "\n";
    gzputs(gz, oss.str().c_str());
}

// ===== Compute numerator N =====
long long compute_N(const vector<int> &exps) {
    long long N_val = 0;
    int running_exp = 0;
    for (int i = 0; i < (int)exps.size(); i++) {
        if (i > 0) running_exp += exps[i - 1];
        N_val += (long long)pow(3, n_eq - 1 - i) * (1LL << running_exp);
    }
    return N_val;
}

// ===== Part A: Full dataset (original permutation set) =====
void run_original_permutations() {
    // Construct base exponent multiset for mod 64
```

```
    vector<int> exps;
    for (int i = 0; i < 16; i++) exps.push_back(1);
    for (int i = 0; i < 8; i++)  exps.push_back(2);
    for (int i = 0; i < 4; i++)  exps.push_back(3);
    for (int i = 0; i < 2; i++)  exps.push_back(4);
    exps.push_back(5);
    exps.push_back(6);

    gzFile gz = gzopen("mod64_original.csv.gz", "wb");
    if (!gz) {
        cerr << "Error opening output file for mod64 original dataset.\n";
        return;
    }

    sort(exps.begin(), exps.end());
    long long count = 0;

    do {
        long long S = accumulate(exps.begin(), exps.end(), 0LL);
        long long D = (1LL << S) - (long long)pow(3, n_eq);
        long long N_val = compute_N(exps);

        write_gz_line(gz, exps, N_val, D);
        count++;
        if (count % 1000000 == 0) cerr << "[Mod64 Original] Generated " << count << " perms\n";
    } while (next_permutation(exps.begin(), exps.end()));

    gzclose(gz);
    cerr << "[Mod64 Original] Finished: " << count << " permutations.\n";
}

// ===== Part B: Variable special exponent sweep =====
void run_variable_special_exponent() {
    // Base set without last exponent
    vector<int> base_exps;
    for (int i = 0; i < 16; i++) base_exps.push_back(1);
    for (int i = 0; i < 8; i++)  base_exps.push_back(2);
    for (int i = 0; i < 4; i++)  base_exps.push_back(3);
    for (int i = 0; i < 2; i++)  base_exps.push_back(4);
    base_exps.push_back(5); // leave out the last exponent, will vary

    gzFile gz = gzopen("mod64_variable.csv.gz", "wb");
    if (!gz) {
        cerr << "Error opening output file for mod64 variable exp dataset.\n";
        return;
    }

    for (int special_exp = 6; special_exp <= 64; ++special_exp) {
        vector<int> exps = base_exps;
        exps.push_back(special_exp);

        sort(exps.begin(), exps.end());
        bool found_R_gt1 = false;
        long long count = 0;
```

```
    do {
      long long S = accumulate(exps.begin(), exps.end(), 0LL);
      long long D = (1LL << S) - (long long)pow(3, n_eq);
      long long N_val = compute_N(exps);

      if (N_val > 0 && D > 0 && N_val % D == 0) {
        long long R = N_val / D;
        if (R > 1) found_R_gt1 = true;
        write_gz_line(gz, exps, N_val, D, special_exp);
      }
      count++;
    } while (next_permutation(exps.begin(), exps.end()));

    cerr << "[Mod64 Variable] Special exp " << special_exp
         << " done. Found R>1? " << (found_R_gt1 ? "YES" : "NO") << "\n";
    if (!found_R_gt1) {
      cerr << "[Mod64 Variable] Stopping: All R <= 1 from here on.\n";
      break;
    }
  }

  gzclose(gz);
}

// ===== Main =====
int main() {
  cerr << "=== Starting Mod64 Composite Program ===\n";
  run_original_permutations();
  run_variable_special_exponent();
  cerr << "=== Finished Mod64 Tasks ===\n";
  return 0;
}
```

(**Code ends**)