hexagon:2611.00902v1

EXACT P-ADIC VALUATIONS OF ALGEBRAIC BINOMIAL COEFFICIENTS AND TRANSFER-OPERATOR ENUMERATION OF SEPARATED PERMUTATIONS: RESOLUTIONS OF OPEN PROBLEMS IN OEIS A387569, A392059, AND A179967

SEON-WOO HONG

Abstract. We resolve three open problems in the On-Line Encyclopedia of Integer Sequences (OEIS) and establish two broader generalizations. First, for $n \geq 1$, let $A387569(n)$ denote the least $y \in \mathbb{Z}_{>0}$ for which there exists $x \in \mathbb{Z}$ with $\binom{x+iy}{n} \in \mathbb{Z}[i]$. Using local completions, we prove that primes $p \equiv 1 \pmod 4$ never divide $A387569(n)$, prove via the Chinese Remainder Theorem that $\binom{x+iy}{n} \in \mathbb{Z}[i]$ is solvable in $x \in \mathbb{Z}$ if and only if $A387569(n) \mid y$, and derive an explicit prime-power formula extending the known table beyond $n = 23$. More generally, for any imaginary quadratic ring of integers $\mathcal{O}_K$ (including non-UFDs such as $\mathbb{Z}[\sqrt{-5}]$), we prove by local valuation decoupling and CRT that unramified split rational primes never divide the minimal existential shift, while showing via $\mathbb{Q}(\sqrt{-3})$ that $2y$-periodicity fails in general and proving an explicit universal period bound via Chu–Vandermonde. Second, for $A392059(n)$, we characterize universal integrality over all $x \in \mathbb{Z}$, establish $\binom{x+A392059(n)i}{c} \equiv \binom{x+2^{\lfloor \log_2 n \rfloor}}{c} \pmod{1+i}$ for $0 \leq c \leq n$, disprove the OEIS congruence conjecture for all $n \geq 1$ via $x^*(n) = n - 2^{\lfloor \log_2 n \rfloor}$, and prove that the minimal period modulo $(1+i)$ is $2 \cdot A392059(n)$. Third, we construct an explicit finite quotient transfer operator resolving Manyama's (2026) conjectures on OEIS A179967 (permutations of $[2n+9]$ with $|p(j)-p(j-1)| > n$), proving pure-integer Stage 1 prefix-graph orbit contraction, bijective Stage 2–3 transition weights, direct enumeration of all pre-bulk terms $0 \leq n \leq 7$, Krylov annihilator minimality, and exact closed forms across $2 \leq k \leq 9$ (OEIS A322255).

1. Introduction

The arithmetic of integer-valued polynomials over algebraic number rings—initiated by Pólya and Ostrowski and generalized via $\mathfrak{p}$-orderings by Bhargava [1, 2]—motivates integrality questions for binomial polynomials along affine slices. For $n \geq 1$ and $\mathbb{Z} + iy \subset \mathbb{Z}[i]$ ($y \in \mathbb{Z}_{>0}$), two dual problems appear in the OEIS:

(1) *Existential integrality* (OEIS A387569 [10]): Find the least $y \geq 1$ with $\binom{x+iy}{n} \in \mathbb{Z}[i]$ for *some* $x \in \mathbb{Z}$. Direct search previously reached only $n = 23$, leaving periodicity in x, the absence of prime factors $p \equiv 1 \pmod 4$, and a general formula open.

Date: November 2026.

2020 *Mathematics Subject Classification.* Primary 11B65, 05A05; Secondary 11R04, 05A15, 13F20.

Key words and phrases. Gaussian integers, Dedekind domains, integer-valued polynomials, binomial coefficients, p-adic valuation, anti-bandwidth permutations, transfer operator, rational generating function, OEIS A387569, OEIS A392059, OEIS A179967, OEIS A322255.

Version 2.6 Final (joint AI–human collaborative edition incorporating Stage 1 prefix-graph automorphism proofs, explicit Stage 2–3 combinatorial weight identities, Vandermonde period bounds, exact 2-adic factorial valuations, and vectorized meet-in-the-middle enumeration through $n = 7$). Primary derivations, transfer-operator constructions, verification code, and manuscript synthesis by Google Gemini (Gemini 4 Carbon); independent adversarial auditing, the $\mathbb{Q}(\sqrt{-3})$ periodicity counterexample, and proof-completeness critiques by OpenAI ChatGPT; confidential prompting architecture and workflow orchestration by Seon-Woo Hong; see Section 5.

(2) *Universal integrality and congruence* (OEIS A392059 [11]): For

$$A392059(n) = \prod_{\substack{p\le n\\ p=2 \text{ or } p\equiv 3 \pmod 4}} p^{\lfloor \log_p n\rfloor}, \tag{1}$$

it was conjectured in [11] that $A392059(n)$ is the minimal $k \ge 1$ satisfying $\binom{x+ki}{n} \equiv \binom{x}{n} \pmod{1+i}$ for *all* $x \in \mathbb{Z}$.

Sections 2 and 3 resolve both problems completely and extend split-prime invisibility to arbitrary imaginary quadratic Dedekind domains $\mathcal{O}_K$. Section 4 resolves Manyama's (2026) conjectures on OEIS A179967 [8] enumerating permutations p of $[2n+9]$ with $|p(j)-p(j-1)| > n$, together with $[2n+k]$ ($2 \le k \le 9$, diagonals of OEIS A322255 [9]).

1.1. **Notation and Binomial Conventions.** Lowercase $p, q \in \mathbb{Z}_{>0}$ denote rational primes, and $v_p(u) \in \mathbb{Z}$ denotes the p-adic valuation on $\mathbb{Q}^\times$ ($v_p(0) = +\infty$). For an algebraic number field K with ring of integers $\mathcal{O}_K$, gothic letters $\mathfrak{P}, \mathfrak{p}, \mathfrak{Q}$ denote nonzero prime ideals of $\mathcal{O}_K$, with normalized discrete valuation $v_{\mathfrak{P}}(z) \in \mathbb{Z}$ on $\mathcal{O}_{K,\mathfrak{P}}$. In $\mathbb{Z}[i]$, inert $p \equiv 3 \pmod 4$ has $\mathfrak{P} = p\mathbb{Z}[i]$ with $v_{p\mathbb{Z}[i]}(u) = v_p(u)$ on $\mathbb{Q}^\times$, while $2\mathbb{Z}[i] = (\varpi)^2$ with $\varpi = 1+i$ and $v_{(\varpi)}(2) = 2$. For any $\mathbb{Q}$-algebra R and $z \in R$, set $\binom{z}{m} := z(z-1)\cdots(z-m+1)/m!$ ($m \ge 1$), $\binom{z}{0} := 1$, and $\binom{z}{m} := 0$ ($m < 0$). Then $\binom{x}{m} \in \mathbb{Z}$ for *all* $x \in \mathbb{Z}$ ($\binom{x}{m} = 0$ for $0 \le x < m$; $\binom{-r}{m} = (-1)^m\binom{r+m-1}{m}$ for $r \ge 1$), and $\binom{z_1+z_2}{n} = \sum_{m=0}^{n}\binom{z_1}{n-m}\binom{z_2}{m}$ holds in $\mathbb{Q}[z_1, z_2]$.

1.2. **Related Work.** For classical binomial rows $\binom{n}{m}$ ($0 \le m \le n$), Rowland [12] gave a p-regular matrix product for $\sum_{m=0}^{n} x^{v_p(\binom{n}{m})}$, generalized by Chand [3] to Fibonomial and C-nomial coefficients. Whereas those works count valuations across $0 \le m \le n$, A387569 and A392059 fix n and vary $x \in \mathbb{Z}$ along $x + iy \in \mathbb{Z}[i]$, governed by prime splitting in $\mathbb{Q}(i)$. In combinatorics, maximizing $\min_{\{u,v\}\in E(G)} |p(u) - p(v)|$ is the *anti-bandwidth* (separation number) problem in frequency assignment [4], scheduling [5], and graph layouts [7]; our transfer operator yields exact rational generating functions and exponential growth rates $\lim_{n\to\infty} n^{-1}\log a_k(n) = \log(\lfloor k/2\rfloor\lceil k/2\rceil)$ on paths P_{2n+k}.

2. Local-to-Global Valuations for OEIS A387569 and Dedekind Domains

Lemma 2.1 (Split primes)**.** *If $p \equiv 1 \pmod 4$ splits in $\mathbb{Z}[i]$ as $p\mathbb{Z}[i] = \mathfrak{p}\bar{\mathfrak{p}}$, then $v_{\mathfrak{p}}(\binom{x+iy}{n}) \ge 0$ and $v_{\bar{\mathfrak{p}}}(\binom{x+iy}{n}) \ge 0$ for all $n \ge 0$ and $x, y \in \mathbb{Z}$.*

Proof. By Hensel's lemma, $T^2 + 1 = 0$ has roots $\pm\iota \in \mathbb{Z}_p$, inducing $\widehat{\mathbb{Z}[i]}_{\mathfrak{p}} \cong \mathbb{Z}_p$ ($i \mapsto +\iota$) and $\widehat{\mathbb{Z}[i]}_{\bar{\mathfrak{p}}} \cong \mathbb{Z}_p$ ($i \mapsto -\iota$). Thus $x + iy$ maps to $\alpha_\pm = x \pm \iota y \in \mathbb{Z}_p$. By Mahler's theorem [6], $T \mapsto \binom{T}{n}$ is continuous on $\mathbb{Z}_p$ and maps the dense set $\mathbb{Z}_{\ge 0}$ into $\mathbb{Z} \subset \mathbb{Z}_p$, so $\binom{\alpha_\pm}{n} \in \mathbb{Z}_p$. □

Lemma 2.2 (Inert primes)**.** *Let $p \equiv 3 \pmod 4$ be prime and $y \in \mathbb{Z}\setminus\{0\}$ with $t = v_p(y) \ge 0$. For $n \ge 1$ and $x \in \mathbb{Z}$,*

$$V_p(n,t;x) := v_{p\mathbb{Z}[i]}\left(\prod_{j=0}^{n-1}(x-j+iy)\right) = \sum_{s=1}^{t}\left|\{0 \le j \le n-1 : p^s \mid (x-j)\}\right| \tag{2}$$

depends only on t and $x \bmod p^t$, with $\max_{x\in\mathbb{Z}} V_p(n,t;x) = \sum_{s=1}^{t}\lceil n/p^s\rceil$ (attained whenever $x \equiv 0 \pmod{p^t}$) and $\min_{x\in\mathbb{Z}} V_p(n,t;x) = \sum_{s=1}^{t}\lfloor n/p^s\rfloor$ (attained at $x \equiv -1 \pmod{p^t}$).

Proof. In $\mathbb{Z}_p[i] = \mathbb{Z}_p \oplus \mathbb{Z}_p i$, $v_{p\mathbb{Z}[i]}(u+iv) = \min(v_p(u), v_p(v))$, so $v_{p\mathbb{Z}[i]}(x-j+iy) = \min(v_p(x-j), t) = \sum_{s=1}^{t}\mathbf{1}_{p^s\mid(x-j)}$, yielding (2). Any n consecutive integers contain between $\lfloor n/p^s\rfloor$ and $\lceil n/p^s\rceil$ multiples of p^s; $x \equiv 0 \pmod{p^t}$ gives $\lfloor (n-1)/p^s\rfloor + 1 = \lceil n/p^s\rceil$ multiples in $\{0,\dots,n-1\}$ for all $1 \le s \le t$, while $x \equiv -1 \pmod{p^t}$ gives $\lfloor n/p^s\rfloor$ multiples in $\{1,\dots,n\}$. □

Lemma 2.3 (Ramified norm resonance)**.** *Let $\varpi = 1 + i$. For $y = 2^t y_0 \in \mathbb{Z} \setminus \{0\}$ ($t = v_2(y) \geq 0$, y_0 odd) and $u \in \mathbb{Z}$,*

$$v_{(\varpi)}(u + iy) = v_2(u^2 + y^2) = 2\sum_{s=1}^{t} \mathbf{1}_{2^s | u} + \mathbf{1}_{u \equiv 2^t \pmod{2^{t+1}}}. \tag{3}$$

Hence $V_{(\varpi)}(n, t; x) := \sum_{j=0}^{n-1} v_{(\varpi)}(x - j + iy)$ depends only on t and $x \bmod 2^{t+1}$ (independently of y_0), with

$$\max_{x \in \mathbb{Z}} V_{(\varpi)}(n, t; x) = 2\sum_{s=1}^{t} \left\lceil \frac{n}{2^s} \right\rceil + \left\lceil \frac{n}{2^{t+1}} \right\rceil, \tag{4}$$

achieved whenever $x \equiv 2^t \pmod{2^{t+1}}$.

Proof. Since $\varpi\bar{\varpi} = 2$ and $\bar{\varpi} = -i\varpi$, $v_{(\varpi)}(u + iy) = v_2(u^2 + y^2)$. If $v_2(u) = r < t$, $v_2(u^2 + y^2) = 2r$; if $v_2(u) > t$, $v_2(u^2 + y^2) = 2t$; if $v_2(u) = t$, writing $u = 2^t u_0$ (u_0 odd) gives $u_0^2 + y_0^2 \equiv 2 \pmod 8$, so $v_2(u^2 + y^2) = 2t + 1$, proving (3). For $x \equiv 2^t \pmod{2^{t+1}}$, $2^s \mid (x - j) \iff 2^s \mid j$ $(1 \leq s \leq t)$ and $x - j \equiv 2^t \pmod{2^{t+1}} \iff 2^{t+1} \mid j$, attaining each upper bound simultaneously on $0 \leq j \leq n - 1$. □

Theorem 2.4 (Complete resolution of OEIS A387569)**.** *For $n \geq 1$, set*

$$B(n) := 2^{t_2(n)} \prod_{\substack{p \leq n \\ p \equiv 3 \pmod 4}} p^{t_p(n)}, \tag{5}$$

where $t_2(n) = \min\{t \geq 0 : 2\sum_{s=1}^{t} \lceil n/2^s \rceil + \lceil n/2^{t+1} \rceil \geq 2v_2(n!)\}$ and $t_p(n) = \min\{t \geq 0 : \sum_{s=1}^{t} \lceil n/p^s \rceil \geq v_p(n!)\}$. Then:

(i) *For fixed $n \geq 1$ and $y \in \mathbb{Z} \setminus \{0\}$, $\binom{x+iy}{n} \in \mathbb{Z}[i]$ is periodic in $x \in \mathbb{Z}$ with period dividing $2|y|$.*

(ii) *For any $y \in \mathbb{Z}$, there exists $x \in \mathbb{Z}$ with $\binom{x+iy}{n} \in \mathbb{Z}[i]$ if and only if $B(n) \mid y$. In particular, $A387569(n) = B(n)$, and $x = y$ satisfies $\binom{y(1+i)}{n} \in \mathbb{Z}[i]$ whenever $B(n) \mid y$.*

Proof. Since $\mathbb{Z}[i] = \bigcap_{\mathfrak{P}} \mathbb{Z}[i]_{\mathfrak{P}}$ and split primes satisfy $v_{\mathfrak{p}}(\binom{x+iy}{n}) \geq 0$ by Lemma 2.1 while primes $p > n$ do not divide $n!$, membership $\binom{x+iy}{n} \in \mathbb{Z}[i]$ ($y \neq 0$) is equivalent to

$$V_{(\varpi)}(n, v_2(y); x) \geq 2v_2(n!) \quad \text{and} \quad V_p(n, v_p(y); x) \geq v_p(n!) \quad (\forall p \leq n,\ p \equiv 3 \pmod 4). \tag{6}$$

By Lemmas 2.2–2.3, these depend on x only modulo $2^{v_2(y)+1}$ and $p^{v_p(y)}$, whose least common multiple divides $2|y|$, proving (i).

For (ii), $y = 0$ is trivial. For $y \neq 0$, $M_2(t) := 2\sum_{s=1}^{t} \lceil n/2^s \rceil + \lceil n/2^{t+1} \rceil$ increases by ≥ 2 per step, and $M_p(t) := \sum_{s=1}^{t} \lceil n/p^s \rceil$ is strictly increasing. *Necessity:* If $B(n) \nmid y$, either $v_2(y) < t_2(n)$ ($\max_x V_{(\varpi)} = M_2(v_2(y)) < 2v_2(n!)$) or $v_q(y) < t_q(n)$ for some inert $q \leq n$ ($\max_x V_q = M_q(v_q(y)) < v_q(n!)$), leaving (6) unsolvable. *Sufficiency:* If $B(n) \mid y$, then $M_2(v_2(y)) \geq 2v_2(n!)$ (attained on $x \equiv 2^{v_2(y)} \pmod{2^{v_2(y)+1}}$ independently of odd factors of y) and $M_p(v_p(y)) \geq v_p(n!)$ (attained on $x \equiv 0 \pmod{p^{v_p(y)}}$). By pairwise coprimality and the Chinese Remainder Theorem, a simultaneous solution exists; explicitly, $x = y = 2^{v_2(y)} y_{\text{odd}}$ satisfies both congruences simultaneously, giving $A387569(n) = B(n)$. □

Theorem 2.5 (Extension to imaginary quadratic Dedekind domains)**.** *Let $d < 0$ be squarefree, $K = \mathbb{Q}(\sqrt{d})$, $\mathcal{O}_K = \mathbb{Z}[\theta]$ ($\theta = \sqrt{d}$ if $d \equiv 2, 3 \pmod 4$; $\theta = \frac{1+\sqrt{d}}{2}$ if $d \equiv 1 \pmod 4$), and $y_K(n) := \min\{y \in \mathbb{Z}_{>0} : \exists x \in \mathbb{Z}$ with $\binom{x+y\theta}{n} \in \mathcal{O}_K\}$. Even when $h(K) > 1$ (non-UFD), no rational prime p splitting completely as $p\mathcal{O}_K = \mathfrak{p}\bar{\mathfrak{p}}$ ($\mathfrak{p} \neq \bar{\mathfrak{p}}$) ever divides $y_K(n)$.*

TABLE 1. Exact values of $A387569(n)$ for $24 \le n \le 50$ from (5).

n	$A387569(n)$	n	$A387569(n)$	n	$A387569(n)$
24	2,422,728	33	75,104,568	42	75,104,568
25	1,211,364	34	75,104,568	43	3,229,496,424
26	2,422,728	35	75,104,568	44	6,458,992,848
27	3,634,092	36	450,627,408	45	9,688,489,272
28	2,422,728	37	75,104,568	46	3,229,496,424
29	1,211,364	38	75,104,568	47	151,786,331,928
30	2,422,728	39	75,104,568	48	303,572,663,856
31	37,552,284	40	150,209,136	49	1,062,504,323,496
32	300,418,272	41	75,104,568	50	151,786,331,928

Proof. At any split prime p ($p\mathcal{O}_K = \mathfrak{p}\bar{\mathfrak{p}}$), $\mathcal{O}_{K,\mathfrak{p}}$ is a DVR with $e(\mathfrak{p}/p) = f(\mathfrak{p}/p) = 1$, so $\widehat{\mathcal{O}}_{K,\mathfrak{p}} \cong \mathbb{Z}_p$ via Hensel lifting; Mahler's theorem gives $\binom{x+y\theta}{n} \in \widehat{\mathcal{O}}_{K,\mathfrak{p}} \cap K = \mathcal{O}_{K,\mathfrak{p}}$ for *all* $x, y \in \mathbb{Z}$.

Suppose a split prime p divides $y := y_K(n)$, set $y' := y/p \in \mathbb{Z}_{>0}$, and fix $x_0 \in \mathbb{Z}$ with $\binom{x_0+y\theta}{n} \in \mathcal{O}_K$. For each non-split rational prime $q \le n$ ($q \ne p$, $v_q(y') = v_q(y)$), let $\mathfrak{Q}$ be the unique prime ideal above q and set $N_q := q^{v_q(n!)+1}$. We find $x'_q \in \mathbb{Z}$ preserving $v_{\mathfrak{Q}}(\binom{x'_q+y'\theta}{n}) \ge 0$:

(1) q *inert:* $\{1, \theta\}$ is an integral $\mathbb{Z}_q$-basis of $\mathcal{O}_{K,\mathfrak{Q}} = \mathbb{Z}_q[\theta]$, so $v_{\mathfrak{Q}}(u + v\theta) = \min(v_q(u), v_q(v))$; since $v_q(y') = v_q(y)$, $x'_q := x_0$ works.

(2) q *odd and ramified* ($q \mid d$)*:* If $d \equiv 2, 3 \pmod 4$ ($\theta = \sqrt{d}$), $v_{\mathfrak{Q}}(u+v\theta) = \min(2v_q(u), 2v_q(v)+1)$, so $x'_q := x_0$ works. If $d \equiv 1 \pmod 4$ ($\theta = \frac{1+\sqrt{d}}{2}$), $2 \in \mathbb{Z}_q^\times$ and $2(x - j + v\theta) = (2x + v - 2j) + v\sqrt{d}$, giving $v_{\mathfrak{Q}}(x - j + v\theta) = \min(2v_q(2x + v - 2j), 2v_q(v) + 1)$. Choosing $x'_q \in \mathbb{Z}$ with $2x'_q + y' \equiv 2x_0 + y \pmod{N_q}$ ($\gcd(2, q) = 1$) preserves $\min(v_{\mathfrak{Q}}(\cdot), 2v_q(n!) + 2)$ on every factor, hence preserves $v_{\mathfrak{Q}}(\binom{x'_q+y'\theta}{n}) \ge 0$.

(3) $q = 2$ *ramified* ($d \equiv 2, 3 \pmod 4$, $\theta = \sqrt{d}$)*:* Here $v_{\mathfrak{Q}_2}(u + v\sqrt{d}) = v_2(u^2 - dv^2)$. If $d \equiv 2 \pmod 4$, parity gives $\min(2v_2(u), 2v_2(v)+1)$. If $d \equiv 3 \pmod 4$ and $v = 2^t v_0$ (v_0 odd), then $v_2(u) \ne t$ gives $v_2(u^2 - dv^2) = 2\min(v_2(u), t)$, whereas $v_2(u) = t$ ($u = 2^t u_0$, u_0 odd) gives $u_0^2 - dv_0^2 \equiv 1 - d \equiv 2 \pmod 4$ and $v_2(u^2 - dv^2) = 2t + 1$ independently of v_0; thus $x'_2 := x_0$ works.

Since local integrality at $q \le n$ depends on x only modulo $N_q = q^{v_q(n!)+1}$ and $\{N_q\}_{q \le n \text{ non-split}}$ are pairwise coprime, CRT yields $x' \in \mathbb{Z}$ with $x' \equiv x'_q \pmod{N_q}$ for all non-split $q \le n$, giving $\binom{x'+y'\theta}{n} \in \bigcap_{\mathfrak{P}} \mathcal{O}_{K,\mathfrak{P}} = \mathcal{O}_K$ and contradicting $y' < y_K(n)$. □

Remark 2.6 (Failure of $2y$-periodicity in general $\mathcal{O}_K$ and explicit universal period bound)**.** Unlike Theorem 2.4(i) for $\mathbb{Z}[i]$, the period in x need not divide $2|y|$ in general: in $K = \mathbb{Q}(\sqrt{-3})$ ($\theta = \frac{1+\sqrt{-3}}{2}$) at $n = 4, y = 4$ ($2y = 8$), $\binom{-10+4\theta}{4} = \frac{931}{3} - \frac{1463}{3}\theta \notin \mathcal{O}_K$ whereas $\binom{-2+4\theta}{4} = -5 + 11\theta \in \mathcal{O}_K$, despite $-10 \equiv -2 \pmod 8$. Instead, for *every* $\mathcal{O}_K$ and $y \in \mathbb{Z}$, integrality is periodic across all $x \in \mathbb{Z}$ with period dividing $P_K(n) := \prod_{q \le n,\, q \text{ non-split}} q^{v_q(n!)}$. Indeed, for $z = x + y\theta$ and $P = P_K(n)$, Chu–Vandermonde gives $\binom{z+P}{n} - \binom{z}{n} = \sum_{m=1}^{n} \binom{z}{n-m}\binom{P}{m}$. Each term ($1 \le m \le n$) is locally integral at split primes (as in Theorem 2.5) and at all primes $> n$ (which do not divide $(n-m)!$), since $\binom{P}{m} \in \mathbb{Z}$. At each non-split prime $q \le n$ ($\mathfrak{Q} \mid q$, $e = e(\mathfrak{Q}/q) \in \{1, 2\}$), $v_q(P) = v_q(n!) \ge \lfloor \log_q n \rfloor \ge v_q(j)$ for $1 \le j < m \le n$ gives $v_q(P - j) \ge v_q(j)$ and $v_q(\binom{P}{m}) \ge v_q(n!) - v_q(m) \ge v_q((n-m)!)$ (as $\binom{n}{m} \in \mathbb{Z}$), while $z - j \in \mathcal{O}_{K,\mathfrak{Q}}$ implies $v_{\mathfrak{Q}}(\binom{z}{n-m}) \ge -e\, v_q((n-m)!)$; thus $v_{\mathfrak{Q}}(\binom{z}{n-m}\binom{P}{m}) \ge 0$ for all $1 \le m \le n$, so $\binom{z+P}{n} - \binom{z}{n} \in \mathcal{O}_K$. Full-period CRT verification (Appendix A) confirms $(y_{\mathbb{Q}(\sqrt{-5})}(n))_{n=1}^{12} = (1, 2, 1, 4, 10, 2, 2, 8, 4, 20, 220, 220)$ in $\mathbb{Z}[\sqrt{-5}]$ ($h = 2$).

3. Universal Integrality and Disproof of the Conjecture in OEIS A392059

Lemma 3.1 (Chu–Vandermonde reduction)**.** *For any additive subgroup $I \subset \mathbb{Q}(i)$, $n \geq 1$, and $k \in \mathbb{Z}$, $\binom{x+ki}{n} - \binom{x}{n} \in I$ holds for all $x \in \mathbb{Z}$ if and only if $\binom{ki}{m} \in I$ for every $1 \leq m \leq n$.*

Proof. Writing $g(x) := \binom{x+ki}{n} - \binom{x}{n} = \sum_{m=1}^{n} \binom{x}{n-m}\binom{ki}{m}$ under Section 1.1, $\binom{x}{n-m} \in \mathbb{Z}$ for all $x \in \mathbb{Z}$ gives sufficiency; conversely, iterated forward differences satisfy $(\Delta^d g)(0) = \binom{ki}{n-d} \in I$ for $0 \leq d \leq n-1$. □

Theorem 3.2 (Characterization and correction of OEIS A392059)**.** *Let $A392059(n)$ be given by* (1) *for $n \geq 1$, and $\tau(n) := \lfloor \log_2 n \rfloor$. Under Section 1.1:*

(i) *For $k \in \mathbb{Z}_{>0}$, $\binom{x+ki}{n} \in \mathbb{Z}[i]$ for all $x \in \mathbb{Z}$ if and only if $A392059(n) \mid k$.*

(ii) *For $k = A392059(n)$, every $0 \leq c \leq n$, and every $x \in \mathbb{Z}$ (including $x < 0$ and $x + 2^{\tau(n)} < c$),*

$$\binom{x + A392059(n)i}{c} \equiv \binom{x}{c} + \binom{x}{c - 2^{\tau(n)}} \equiv \binom{x + 2^{\tau(n)}}{c} \pmod{1+i}. \tag{7}$$

(iii) *For every $n \geq 1$, $k = A392059(n)$ and $x^*(n) = n - 2^{\tau(n)}$ give $\binom{x^*(n)+ki}{n} - \binom{x^*(n)}{n} \equiv 1 \not\equiv 0 \pmod{1+i}$, disproving the conjecture in OEIS A392059 for all $n \geq 1$.*

(iv) *For every $n \geq 1$, the minimal $k \geq 1$ with $\binom{x+ki}{n} \in \mathbb{Z}[i]$ and $\binom{x+ki}{n} \equiv \binom{x}{n} \pmod{1+i}$ for all $x \in \mathbb{Z}$ is $2 \cdot A392059(n)$.*

Proof. By Lemma 3.1 ($I = \mathbb{Z}[i]$) and Lemma 2.1, universal integrality holds iff $\binom{ki}{m}$ is locally integral at $p = 2$ and odd inert $p \leq n$ for all $1 \leq m \leq n$. For inert $p \equiv 3 \pmod 4$ and $t = v_p(k)$, $v_{p\mathbb{Z}[i]}(\binom{ki}{m}) = t + \sum_{j=1}^{m-1} \min(v_p(j), t) - v_p(m!)$, which is $t - v_p(m) \geq 0$ for $1 \leq m \leq n$ when $t \geq \lfloor \log_p n \rfloor$, and -1 at $m = p^{t+1} \leq n$ when $t < \lfloor \log_p n \rfloor$. At $p = 2$ ($t = v_2(k)$), $v_{(\varpi)}(\binom{ki}{m}) = \sum_{j=0}^{m-1} v_2(j^2+k^2) - 2v_2(m!)$:

- If $t \geq \tau(n)+1$ ($2^t > n$), $v_2(j) < t$ for $1 \leq j < m$, giving $2(t - v_2(m)) \geq 2$ for all $1 \leq m \leq n$.
- If $t = \tau(n)$, $1 \leq m \leq 2^t$ gives $2(t - v_2(m))$ (≥ 2 for $m < 2^t$, and 0 at $m = 2^t$); for $m = 2^t + s$ ($1 \leq s < 2^t$), $j = 2^t$ contributes $2t+1$ by Lemma 2.3, giving $v_{(\varpi)}(\binom{ki}{2^t+s}) = 2(t - v_2(s)) + 1 \geq 3$.
- If $t < \tau(n)$, setting $m = 2^{t+1} \leq n$ gives numerator valuation $2t + (2t+1) + 4v_2((2^t-1)!) = 2^{t+2} - 3$ and denominator valuation $2v_2((2^{t+1})!) = 2^{t+2} - 2$ (via $v_2((2^r)!) = 2^r - 1$), hence $(2^{t+2} - 3) - (2^{t+2} - 2) = -1 < 0$.

This proves (i). When $k = A392059(n)$ ($t = \tau(n)$), $\binom{ki}{m} \equiv \delta_{m,2^{\tau(n)}} \pmod{\varpi}$ for $1 \leq m \leq n$. Reducing $\binom{x+ki}{c} = \sum_{m=0}^{c} \binom{x}{c-m}\binom{ki}{m}$ modulo (ϖ) gives $\binom{x}{c} + \binom{x}{c-2^{\tau(n)}}$. In $\mathbb{Z}[[T]]$, $(1+T)^x = \sum_{d=0}^{\infty} \binom{x}{d} T^d$ holds for *all* $x \in \mathbb{Z}$; reducing modulo 2 in $\mathbb{F}_2[[T]]$ gives $(1+T)^x(1+T)^{2^{\tau(n)}} = (1+T)^x(1+T^{2^{\tau(n)}})$, equating coefficients of T^c to prove (7). Setting $c = n$ and $x^*(n) = n - 2^{\tau(n)}$ gives $\binom{n}{n} - \binom{x^*(n)}{n} = 1$, proving (iii); requiring $v_{(\varpi)}(\binom{ki}{2^{\tau(n)}}) \geq 1$ forces $v_2(k) \geq \tau(n) + 1$, proving (iv). □

4. Self-Contained Transfer-Operator Theory for Separated Permutations

Let $G_{n,k}$ be the graph on $[2n+k]$ with edges $\{x, y\}$ for $|x - y| > n$; $a_k(n)$ is twice the number of undirected Hamiltonian paths in $G_{n,k}$.

Proposition 4.1 (Half-graph decomposition)**.** *For $n \geq k-1$ and $m := n - k + 1 \geq 0$, partition $[2n+k]$ into five disjoint sets:*

$$U_L = \{1, \ldots, k-1\}, \quad U_{\text{bulk}} = \{\tilde{u}_1, \ldots, \tilde{u}_m\}, \quad C = \{c_1, \ldots, c_k\},$$
$$V_{\text{bulk}} = \{\tilde{v}_1, \ldots, \tilde{v}_m\}, \quad V_R = \{w_1, \ldots, w_{k-1}\},$$

with $\tilde{u}_t = k - 1 + t$, $c_r = n + r$, $\tilde{v}_t = n + k + t$, $w_j = 2n + 1 + j$. Set $U := U_L \cup U_{\text{bulk}}$ and $V := V_{\text{bulk}} \cup V_R$. Then U, V, and C are independent sets in $G_{n,k}$; no edge joins C to $U_{\text{bulk}} \cup V_{\text{bulk}}$,

whereas between C and $U_L \cup V_R$, $c_r \sim u \in U_L \iff u \le r-1$ and $c_r \sim w_j \in V_R \iff j \ge r$; furthermore, $U_L \times V \subset E(G_{n,k})$, $U \times V_R \subset E(G_{n,k})$, and $\tilde{u}_t \sim \tilde{v}_{t'} \iff t \le t'$.

Proof. Internal differences in U, V are $\le n-1$ and in C are $\le k-1 \le n$. For $c_r = n+r$, $(n+r)-x > n \iff x \le r-1$ ($x \in U_L$), and $(n+k+s)-(n+r) > n \iff s \ge m+r$ ($v_s = w_j \in V_R$ with $j \ge r$). For $u_i = i \in U$ and $v_s = n+k+s \in V$, $(n+k+s)-i > n \iff i-s \le k-1$, which holds if $i \le k-1$ or $s \ge m+1$, and gives $t \le t'$ for $i = k-1+t, s = t'$. □

Every undirected Hamiltonian path P of $G_{n,k}$ ($n \ge k-1$) has two degree-1 global endpoints partitioned across $(C \cup U_L) \sqcup (U_{\text{bulk}} \cup V_{\text{bulk}}) \sqcup V_R$, built uniquely in three stages: *Stage 1* selects all edges of P incident to C (into $U_L \cup V_R$) and designates whichever global endpoints (0, 1, or 2) lie in $C \cup U_L$; *Stage 2* processes $(\tilde{u}_t, \tilde{v}_t)$ for $t = 1, \dots, m$, adding $\tilde{u}_t$ to the active U-pool, saturating $\tilde{v}_t$, and designating any global endpoints in $U_{\text{bulk}} \cup V_{\text{bulk}}$; *Stage 3* designates any remaining global endpoints on V_R, saturates V_R against the surviving active U-pool, and multiplies by 2.

Lemma 4.2 (Intertwining quotient)**.** *For an active U-pool with q untouched vertices, m_{pair} twin-paired U-to-U segments, and f boundary-tethered ends ($C_U := q + m_{\text{pair}}$), the internal bulk block $M_{C_U,f}$ on $q \in \{0, \dots, C_U\}$ satisfies $\mathbf{y}^\top M_{C_U,f} = (C_U+1)(C_U+f)\mathbf{y}^\top$ for $\mathbf{y}^\top = (2^{C_U-q})_{q=0}^{C_U}$. Moreover, all bulk and Stage 3 operators intertwine $\Pi(q, m_{\text{pair}}) = 2^{m_{\text{pair}}}$, collapsing (q, m_{pair}) onto C_U.*

Proof. After $\tilde{u}_t$ sends $q \mapsto q+1$, Π-weighted transitions from (q, m_{pair}) sum to $2\binom{q+1}{2} + (q+1)(2m_{\text{pair}} + f) + m_{\text{pair}}(m_{\text{pair}} + f - 1) = (C_U+1)(C_U+f)$, independent of m_{pair}. Similarly, single-slot and pair-slot extractions without $\tilde{u}_t$ yield $q + m_{\text{pair}} = C_U$ and $q(q-1) + q(2m_{\text{pair}} + f) + m_{\text{pair}}(m_{\text{pair}} + f - 1) = C_U(C_U + f - 1)$, with terminal weight $2^0 = 1$ at $C_U = 0$. □

Definition 4.3 (Quotiented state tuple)**.** A non-closed cut configuration is given by $s = (C_U, b_{UV}, e_U, q_{V_R}, m_{V_RV_R}, e_{V_R}, e_{\text{tot}}) \in \mathbb{Z}^7_{\ge 0}$ (plus terminal state CLOSED when V_R is pre-saturated in Stage 1), where $C_U = q + m_{\text{pair}}$ is the Π-quotiented U-capacity; b_{UV} counts U-to-V_R bridges; $e_U \in \{0,1,2\}$ counts U-to-endpoint segments; q_{V_R} counts untouched V_R-vertices; $m_{V_RV_R}$ counts V_R-to-V_R arches; $e_{V_R} \in \{0,1,2\}$ counts V_R-to-endpoint segments; and $e_{\text{tot}} \in \{0,1,2\}$ counts placed global endpoints.

Lemma 4.4 (Degree-balance invariant and finiteness)**.** *Setting $H_U(s) := 2C_U + b_{UV} + e_U$ and $H_V^{\max}(s) := 2q_{V_R} + 2m_{V_RV_R} + b_{UV} + e_{V_R}$, any completable state satisfies*

$$\left|H_U(s) - H_V^{\max}(s)\right| \le 2 - e_{\text{tot}} \qquad \text{and} \qquad H_U(s) - H_V^{\max}(s) \equiv 2 - e_{\text{tot}} \pmod 2. \tag{8}$$

Hence the reachable state set $\mathcal{V}_{\text{reach}}(k)$ is finite for every $k \ge 2$.

Proof. If $r_U, r_{\tilde{v}}, r_{V_R} \ge 0$ ($r_U + r_{\tilde{v}} + r_{V_R} = r_{\text{rem}} := 2 - e_{\text{tot}}$) denote remaining endpoints placed on future $\tilde{u}$, future $\tilde{v}$, and V_R, Stage 3 degree equality requires $H_U(s) - r_U + r_{\tilde{v}} = H_V^{\max}(s) - r_{V_R}$, so $H_U(s) - H_V^{\max}(s) = 2r_U - r_{\text{rem}}$, yielding (8). Since $q_{V_R} + 2m_{V_RV_R} + b_{UV} + e_{V_R} \le k-1$, (8) bounds all coordinates of s. □

Lemma 4.5 (Exactness of Stage 1 staircase contraction and orbit double-counting)**.** *Process $c_1, \dots, c_k$ across $r = 1, \dots, k$. At step r, define the active left pool $U_a^{(r)} := \{u_1, \dots, u_{r-1}\}$ (u_{r-1} entering before c_r for $r \ge 2$, untouched or as a global endpoint), the active right pool $V_a^{(r)} := \{w_r, \dots, w_{k-1}\}$ of size $N_{V_a} := k-r$, and the frozen pool $V_f^{(r-1)} := \{w_1, \dots, w_{r-1}\}$. Classify each labeled partial forest F after c_r by the 13-tuple*

$$\tilde{s} = (q_U, q_{V_a}, q_{V_f},\ m_{UU}, b_{UV_a}, b_{UV_f}, e_U,\ m_{V_aV_a}, b_{V_aV_f}, e_{V_a},\ m_{V_fV_f}, e_{V_f},\ e_{\text{tot}}) \in \mathbb{Z}^{13}_{\ge 0},$$

where (q_U, q_{V_a}, q_{V_f}) count degree-0 vertices in $(U_a^{(r)}, V_a^{(r)}, V_f^{(r-1)})$, the next nine coordinates count open paths by the unordered pool-pair of their two ends in $\{U_a^{(r)}, V_a^{(r)}, V_f^{(r-1)}, E\}$ ((E,E) forbidden), and $e_{\text{tot}} \in \{0,1,2\}$. Let $W_r(\tilde{s}) := |\mathcal{F}_r(\tilde{s})|$ be the number of labeled forests in class $\tilde{s}$ after c_r. Then:

(a) *For $1 \le r \le k-1$, moving w_r from $V_a^{(r)}$ into $V_f^{(r)}$ partitions $\mathcal{F}_r(\tilde{s})$ by the role R of w_r among six disjoint roles (saturated degree-2, counted by $s_{V_a} := N_{V_a} - q_{V_a} - (b_{UV_a} + 2m_{V_aV_a} + b_{V_aV_f} + e_{V_a})$, or role counts $c_R(\tilde{s}) \in \{q_{V_a}, b_{UV_a}, 2m_{V_aV_a}, b_{V_aV_f}, e_{V_a}\}$). For every $\tilde{s}$ and R, N_{V_a} divides $W_r(\tilde{s}) \cdot c_R(\tilde{s})$ in $\mathbb{Z}$, and the exact labeled count with w_r in role R is $W_r(\tilde{s}) \cdot c_R(\tilde{s})/N_{V_a} \in \mathbb{Z}_{\ge 0}$.*

(b) *At $r = k$, $N_{V_a} = 0$, so $(q_{V_a}, b_{UV_a}, m_{V_aV_a}, b_{V_aV_f}, e_{V_a}) = (0,0,0,0,0)$, and $\Pi(q_U, m_{UU}) = 2^{m_{UU}}$ projects $(W_k(\tilde{s}))$ losslessly onto the Stage 1 initial vector $\mathbf{u}_0$ on Definition 4.3.*

Proof. Let H_r denote the Stage 1 processed prefix graph at step r on vertex set $U_a^{(r)} \sqcup \{c_1, \dots, c_r\} \sqcup V_f^{(r-1)} \sqcup V_a^{(r)}$, whose edges are the Stage 1 incidences $\{\{c_s, u_i\} : 1 \le i < s \le r\} \cup \{\{c_s, w_j\} : 1 \le s \le r,\ s \le j \le k-1\}$. By Proposition 4.1, $V_a^{(r)} = \{w_r, \dots, w_{k-1}\}$ is an independent set, every $u \in U_a^{(r)}$ and every processed c_s $(1 \le s \le r)$ is adjacent in $G_{n,k}$ to every $w \in V_a^{(r)}$, and every $w \in V_a^{(r)}$ has the identical neighborhood $\{c_1, \dots, c_r\}$ in H_r. Extending each permutation in $\mathfrak{S}(V_a^{(r)}) \cong S_{N_{V_a}}$ by the identity outside $V_a^{(r)}$ therefore yields graph automorphisms of the processed prefix graph H_r (though not of $G_{n,k}$ for $r < k$, since future c_{r+1} is non-adjacent to w_r), preserving $\mathcal{F}_r(\tilde{s})$. Counting pairs $(F, w) \in \mathcal{F}_r(\tilde{s}) \times V_a^{(r)}$ where w has role R in F gives $W_r(\tilde{s}) \cdot c_R(\tilde{s})$ summing over F first, and $N_{V_a} \cdot |\{F \in \mathcal{F}_r(\tilde{s}) : w_r \text{ has role } R\}|$ by transitivity of $\mathfrak{S}(V_a^{(r)})$ summing over w first, proving (a). At $r = k$, $V_a^{(k)} = \emptyset$ forces all five V_a-coordinates to vanish and $V_f^{(k-1)} = V_R$, mapping via Lemma 4.2 directly onto Definition 4.3. □

Proposition 4.6 (Bijective exactness of Stage 2–3 transition multiplicities)**.** *Every undirected Hamiltonian path P of $G_{n,k}$ $(n \ge k-1)$ corresponds to a unique trajectory of labeled partial forests across Stages 1–3 without double-counting, and Lemma 4.5 yields $a_k(n) = \langle \mathbf{w}_{\text{close}}, T^{n-k+1}\mathbf{u}_0 \rangle$ with exact rules on $s = (C, b, e, q_V, m_V, e_V, e_{\text{tot}})$ ($v_{\text{rem}} := q_V + 2m_V + b + e_V$):*

(1) $\tilde{u}$-insertion $T_{\tilde{u}}$: *$\tilde{u}_t$ enters as internal ($C \mapsto C+1$, wt 1) or, if $e_{\text{tot}} < 2$, as endpoint ($e \mapsto e+1, e_{\text{tot}} \mapsto e_{\text{tot}}+1$, wt 1).*

(2) $\tilde{v}$-saturation $T_{\tilde{v}}$: *Complete bipartiteness makes valid completions invariant under $S_q \times (S_2 \wr S_{m_{\text{pair}}}) \times S_b \times S_{q_V} \times (S_2 \wr S_{m_V})$. Cycle-freeness forbids joining twin ends of an m_{pair}-arch, picking an untouched U-vertex twice, or joining two e-ends unless $v_{\text{rem}} = 0$. Applying $\Pi(q, m_{\text{pair}}) = 2^{m_{\text{pair}}}$ from Lemma 4.2:*

- *Endpoint $\tilde{v}_t$ ($e_{\text{tot}} < 2$) picks 1 U-slot: from (q, m_{pair}) in $q + 2m_{\text{pair}} \cdot \frac{1}{2} = C$ ways $\to (C-1, b, e+1, \dots, e_{\text{tot}}+1)$; from b bridges in b ways $\to (C, b-1, e, \dots, e_V+1, e_{\text{tot}}+1)$; or $(0,0,1,0,0,0,1) \to$ CLOSED (wt 1).*
- *Internal $\tilde{v}_t$ picks 2 U-slots: both from (q, m_{pair}) in $2\binom{q}{2} + 2m_{\text{pair}}q + m_{\text{pair}}(m_{\text{pair}}-1) = C(C-1)$ ways or one from (q, m_{pair}) and one from $b+e$ in $(q+m_{\text{pair}})(b+e) = C(b+e)$ ways, totaling $C(C-1+b+e)$ ways $\to (C-1, b, e, \dots)$; two bridges in $\binom{b}{2}$ ways $\to (C, b-2, e, q_V, m_V+1, \dots)$; one bridge and one e-end in be ways $\to (C, b-1, e-1, q_V, m_V, e_V+1, \dots)$; or $(0,0,2,0,0,0,2) \to$ CLOSED (wt 1).*

(3) Stage 3 closure $\mathbf{w}_{\text{close}}$: *Saturating open vertices of V_R in canonical order avoids double-counting: $q_V \to q_V - 1$ applies $T_{\tilde{v}}$; $e_V \to e_V - 1$ consumes 1 U-slot (C or b ways, or closes at $v_{\text{rem}} = 1$); the first end of $m_V \to m_V - 1$ turns its partner into e_V (endpoint: 1 way; joined to e: e ways), b (C ways), or m_V (b ways); and $b \to b-1$ becomes e (1 or e ways), joins another bridge ($b-1$ ways, avoiding its own U-end), or joins (q, m_{pair}) in $2q + 2m_{\text{pair}} = 2C$ ways, multiplying closed spanning paths by 2.*

Proof. Each spanning path $P \subset G_{n,k}$ restricts uniquely to every prefix; P is a Hamiltonian path iff degrees are 2 (except 2 endpoints of degree 1), no step closes a cycle, and the final edge joins the two endpoint components. Lemma 4.5, items (1)–(3), and Lemma 4.2 partition all valid local selections bijectively. □

Example 4.7 (Worked 12-state system for $k = 3$, A129534)**.** For $k = 3$ ($n \geq 2$), $\mathcal{V}_{\text{reach}}(3)$ has $D_3 = 12$ states: $s_0 = (0,0,0,0,0,1,1)$ [u_0=1, w=2], $s_1 = (0,0,1,0,0,1,2)$ [4, 2], $s_2 = (0,0,1,0,1,0,1)$ [1, 4], $s_3 = (0,0,1,1,0,0,1)$ [1, 2], $s_4 = (0,0,2,0,1,0,2)$ [1, 4], $s_5 = (0,0,2,1,0,0,2)$ [1, 2], $s_6 = (0,1,0,0,0,0,0)$ [1, 0], $s_7 = (0,1,0,0,0,1,1)$ [1, 2], $s_8 = (0,1,1,0,0,0,1)$ [1, 0], $s_9 = (0,1,1,0,0,1,2)$ [1, 2], $s_{10} = (1,0,0,0,0,1,1)$ [3, 0], and $s_{11} = (1,0,0,0,0,2,2)$ [3, 2]. Transitions $Ts_9 = 2s_9 + s_{11}$ ($Ts_{11} = 2s_{11}$) and $Ts_8 = 2s_8 + s_{10} + 2s_1$ ($Ts_{10} = 2s_{10} + s_1$, $Ts_1 = s_1$) yield the $\lambda = 2$ Jordan chains while the $\lambda = 1$ mode cancels on $\mathbf{u}_0$, giving $\langle (T^\top - 2I)^2 \mathbf{w}, T^m \mathbf{u}_0 \rangle = 0$ ($\forall m \geq 0$) and proving $a_3(n) = (n+6)2^n$ ($n \geq 1$).

Theorem 4.8 (Resolution of OEIS A179967 and family A322255)**.** *For $k = 9$, $\dim \mathcal{V}_{\text{reach}}(9) = 193$, and $a(n) = A179967(n)$ has generating function*

$$G(x) = \frac{\begin{aligned} &362880 - 23733610x + 574782136x^2 - 6066574082x^3 + 23063767496x^4 \\ &\quad + 1306804904x^5 + 19195078240x^6 + 99691196928x^7 + 5798373293056x^8 \\ &\quad + 140546260769280x^9 - 1838864889858048x^{10} \end{aligned}}{(1-8x)(1-14x)(1-18x)(1-20x)^2} \tag{9}$$

for all $n \geq 0$, and closed form for all $n > 5$:

$$\begin{aligned} a(n) = {} & \frac{154924827300000\,n + 5704752193340625}{31250000000} \cdot 20^n - \frac{3073238635}{354294} \cdot 18^n \\ & - \frac{13406475}{235298} \cdot 14^n - \frac{105}{8192} \cdot 8^n. \end{aligned} \tag{10}$$

Across $2 \leq k \leq 9$, the minimal recurrence operator for $n \geq k-1$ is $\prod_{C=0}^{\lfloor (k-1)/2 \rfloor} (S - (C+1)(k-1-C))$, yielding the exact closed forms in Table 3.

Proof. By Lemma 4.5 and Proposition 4.6, for $k = 9$ and $n \geq 8$, $a(n) = \langle \mathbf{w}_{\text{close}}, T^{n-8} \mathbf{u}_0 \rangle$ on $\mathbb{Q}^{193}$. For $P(\lambda) = (\lambda - 8)(\lambda - 14)(\lambda - 18)(\lambda - 20)^2$ and $\mathbf{z} := P(T^\top)\mathbf{w}_{\text{close}}$, Cayley–Hamilton implies $\mathcal{K}(\mathbf{u}_0) = \text{span}_{\mathbb{Q}}\{T^0 \mathbf{u}_0, \ldots, T^{192} \mathbf{u}_0\}$. Appendix A verifies Krylov annihilation $\langle \mathbf{z}, T^m \mathbf{u}_0 \rangle = 0$ across $0 \leq m \leq 194$ (and non-vanishing if any root factor is omitted), proving $P(S)a(n) = 0$ for all $n \geq 8$. All eight pre-bulk values $a(0), \ldots, a(7)$ are computed directly without transfer operators (subset DP for $0 \leq n \leq 4$ and vectorized meet-in-the-middle for $n = 5, 6, 7$, i.e., $N = 19, 21, 23$). Combining $a(0), \ldots, a(7)$ with $a(8), \ldots, a(12)$ yields zero coefficients at x^{11}, x^{12} in $(\sum_{n=0}^{12} a(n)x^n)(1-8x)(1-14x)(1-18x)(1-20x)^2$, proving (9) and (10). Analogous Krylov verification for $2 \leq k \leq 8$ (cross-checked against raw labeled Stage 1 enumeration for $2 \leq k \leq 5$ and subset DP across $2 \leq k \leq 6$, $2n + k \leq 16$) establishes Table 3. □

TABLE 2. Values of $A179967(n)$ for $15 \leq n \leq 20$ beyond the main OEIS display ($0 \leq n \leq 14$), confirming agreement between $\langle \mathbf{w}_{\text{close}}, T^{n-8} \mathbf{u}_0 \rangle$, (10), and Manyama's table [8].

n	$A179967(n)$
15	8,360,092,086,071,998,742,691,840
16	170,567,948,261,566,724,975,493,120
17	3,478,446,688,452,290,960,163,471,360
18	70,906,470,173,375,431,565,055,098,880
19	1,444,804,224,341,790,235,429,754,634,240
20	29,428,214,824,670,023,287,111,753,400,320

TABLE 3. Reachable dimensions D_k, minimal recurrence roots of $a_k(n)$, and exact closed forms for $n \geq k-1$ across $2 \leq k \leq 9$ (diagonals of OEIS A322255).

k	OEIS / Seq.	D_k	Minimal recurrence roots	Exact proved formula $a_k(n)$ for all $n \geq k-1$
2	Const (2)	1	$\{1\}$	2
3	A129534	12	$\{2, 2\}$	$(n+6) \cdot 2^n$
4	A179962	31	$\{3, 4\}$	$\frac{49}{2} \cdot 4^n - \frac{8}{3} \cdot 3^n$
5	A179963	55	$\{4, 6, 6\}$	$\frac{182n+2633}{27} \cdot 6^n - \frac{3}{4} \cdot 4^n$
6	A179964	79	$\{5, 8, 9\}$	$\frac{462944}{729} \cdot 9^n - \frac{7883}{64} \cdot 8^n - \frac{32}{125} \cdot 5^n$
7	A179965	114	$\{6, 10, 12, 12\}$	$\frac{56571n+1394594}{432} \cdot 12^n - \frac{103701}{1250} \cdot 10^n - \frac{5}{54} \cdot 6^n$
8	A179966	147	$\{7, 12, 15, 16\}$	$\frac{978034131}{32768} \cdot 16^n - \frac{1954858304}{253125} \cdot 15^n - \frac{174619}{2592} \cdot 12^n - \frac{576}{16807} \cdot 7^n$
9	A179967	193	$\{8, 14, 18, 20, 20\}$	See closed form (10)

TABLE 4. Logical dependency and computational verification ledger.

Statement	Infinite-Range Proof Mechanism	Finite-Range / Annihilator Check in Appendix A
Lemmas 2.1–2.3	Hensel's lemma; Mahler [6]	`binom_theta` local valuation checks
Theorem 2.4 (A387569)	Dedekind $\mathcal{O}_K = \bigcap \mathcal{O}_{K,\mathfrak{P}}$; CRT	Suff. ($n \leq 27$), prime-del ($n \leq 16$), min-y ($n \leq 9$)
Theorem 2.5, Rem. 2.6	DVR completion $\widehat{\mathcal{O}}_{K,\mathfrak{p}} \cong \mathbb{Z}_p$; CRT	Full-period CRT $\mathbb{Z}[\sqrt{-5}]$ ($n \leq 12$); $\mathbb{Q}(\sqrt{-3})$
Lemma 3.1, Theorem 3.2	Chu–Vandermonde; $\mathbb{F}_2[[T]]$ Frobenius	`A392059_def` ($n \leq 15$; $x \in [-8, 9]$ for $n \leq 8$)
Lemma 4.5, Prop. 4.6	$\mathrm{Aut}(H_r)$ double-counting & bijective cuts	Labeled u_0 ($k \leq 5$); DP/MITM $n \leq 7$; bulk $2n + k \leq 16$
Example 4.7, Theorem 4.8	Cayley–Hamilton on $\mathbb{Q}^{D_k}$ ($D_9 = 193$)	Exact Krylov annihilator $0 \leq m \leq D_k + 1$ ($2 \leq k \leq 9$)

5. Proof-Dependency Ledger, Archiving, and AI Provenance Statement

AI Provenance and Public Availability. In accordance with transparent disclosure standards for AI-generated mathematical research, we state explicitly that this work is a joint AI–human collaborative study orchestrated by Seon-Woo Hong using a confidential prompting architecture. Primary mathematical derivations, Dedekind domain extensions, transfer-operator constructions, verification code, and manuscript prose (Versions 1.0–2.6) were generated by Google Gemini (model: `gemini-4-carbon`); independent adversarial mathematical auditing, the explicit $\mathbb{Q}(\sqrt{-3})$ counterexample to $2y$-periodicity, Chu–Vandermonde period-bound critiques, and Stage 1 prefix-graph completeness reviews across Versions 2.0–2.6 were contributed by OpenAI ChatGPT. No claim of unaided human mathematical derivation is made. Self-contained mathematical proofs and prompt-independent computational verification are provided in Sections 2–4 and Appendix A, archived publicly on CERN Zenodo.

References

[1] M. Bhargava, *The factorial function and generalizations*, Amer. Math. Monthly **107** (2000), no. 9, 783–799.

[2] P.-J. Cahen and J.-L. Chabert, *Integer-Valued Polynomials*, Math. Surveys Monogr., vol. 48, Amer. Math. Soc., Providence, RI, 1997.

[3] A. Chand, *Universal matrices for counting Fibonomial and C-nomial coefficients by their p-adic valuations*, preprint (2025), `arXiv:2508.18461`.

[4] W. K. Hale, *Frequency assignment: theory and applications*, Proc. IEEE **68** (1980), no. 12, 1497–1514.

[5] J. Y.-T. Leung, O. Vornberger, and J. D. Witthoff, *On some variants of the bandwidth minimization problem*, SIAM J. Comput. **13** (1984), no. 3, 650–667.

[6] K. Mahler, *An interpolation series for continuous functions of a p-adic variable*, J. Reine Angew. Math. **199** (1958), 23–34.

[7] Z. Miller and D. Pritikin, *On the separation number of a graph*, Networks **19** (1989), no. 6, 651–666.

[8] OEIS Foundation Inc., Entry A179967 (conjectures and table for $0 \leq n \leq 763$ by S. Manyama, August 2026), `https://oeis.org/A179967`.

[9] OEIS Foundation Inc., Entry A322255 (and diagonals A129534, A179962–A179967), https://oeis.org/A322255.
[10] OEIS Foundation Inc., Entry A387569 (R. Zhou and M. Marcus, 2025–2026), https://oeis.org/A387569.
[11] OEIS Foundation Inc., Entry A392059 (R. Zhou, 2025–2026), https://oeis.org/A392059.
[12] E. Rowland, *A matrix generalization of a theorem of Fine*, Integers **18A** (2018), Article A18, arXiv:1704.05872.

Appendix A. Standalone Exact-Arithmetic Verification Script

Tested under Python 3.12.5, NumPy 2.1.3, and SymPy 1.13.3 (and cross-tested under Python 3.13.5, NumPy 2.3.5, SymPy 1.14.0; save as `verify.py` and run `python3 verify.py`). The script executes finite-range sufficiency and full-period CRT necessity checks for Theorems 2.4–3.2, pure-integer Stage 1 contraction cross-checked against raw labeled backtracking (`build_u0_labeled`, where `dg[x]` $\in \{0,1,2\}$ records consumed slot capacity $\deg_C(x)+\mathbf{1}_{\mathrm{endpoint}}(x)$, so 2 marks a closed vertex) for $2 \le k \le 5$, direct path enumeration for $0 \le n \le 7$ ($N \le 23$) of OEIS A179967 (in `mitm_odd_vec`, each path on odd $N = 2L-1$ splits uniquely at midpoint $u = p(L)$ into two L-vertex half-paths with masks $M_1 \cap M_2 = \{u\}$ and $M_2 = (\mathrm{full} \oplus M_1) \,|\, 2^u$, using graph reflection $v \mapsto N-1-v$ to halve the midpoint loop), subset-DP bulk checks ($2 \le k \le 6$, $2n+k \le 16$), and exact Krylov annihilator verification over $0 \le m \le D_k + 1$.

```
import math
from collections import defaultdict
from fractions import Fraction
import numpy as np
import sympy as sp

def binom_theta(x, y, d, n):
    c0, c1 = 1, 0
    kc = (d - 1) // 4 if (d % 4 == 1) else None
    for j in range(n):
        u = x - j
        if kc is None: c0, c1 = c0 * u + d * c1 * y, c0 * y + c1 * u
        else: c0, c1 = c0 * u + kc * c1 * y, c0 * y + c1 * u + c1 * y
    den = math.factorial(n)
    return Fraction(c0, den), Fraction(c1, den)

def vp_fact(n, p):
    s = 0
    while n >= p: n //= p; s += n
    return s

def is_y_feasible_crt(y, d, n):
    kc = (d - 1) // 4 if (d % 4 == 1) else None
    for p in sp.primerange(2, n + 1):
        pe = p ** vp_fact(n, p); ok_p = False
        for xp in range(pe):
            c0, c1 = 1, 0
            for j in range(n):
                u = (xp - j) % pe
                if kc is None: c0, c1 = (c0 * u + d * c1 * y) % pe, (c0 * y + c1 * u) % pe
                else: c0, c1 = (c0 * u + kc * c1 * y) % pe, (c0 * y + c1 * u + c1 * y) % pe
            if c0 == 0 and c1 == 0: ok_p = True; break
        if not ok_p: return False
    return True

def A387569_formula(n):
    req2, t2 = 2 * vp_fact(n, 2), 0
    while (2 * sum((n + (1 << s) - 1) >> s for s in range(1, t2 + 1))
           + ((n + (1 << (t2 + 1)) - 1) >> (t2 + 1)) < req2):
        t2 += 1
    res = 1 << t2
    for p in sp.primerange(3, n + 1):
        if p & 3 == 3:
```

```
            reqp, acc, pk = vp_fact(n, p), 0, 1
            while acc < reqp: pk *= p; acc += (n + pk - 1) // pk
            res *= pk
    return res

def A392059_def(n):
    res = 1
    for p in [2] + [q for q in sp.primerange(3, n + 1) if q % 4 == 3]:
        pk = 1
        while pk * p <= n: pk *= p
        res *= pk
    return res

def verify_number_theory():
    oeis_23 = [1, 2, 3, 12, 6, 6, 42, 168, 252, 84, 924, 2772, 924, 924,
               2772, 3696, 924, 5544, 52668, 105336, 52668, 105336, 1211364]
    assert [A387569_formula(n) for n in range(1, 24)] == oeis_23
    for n in range(1, 28):
        y = A387569_formula(n); r0, r1 = binom_theta(y, y, -1, n)
        assert r0.denominator == 1 and r1.denominator == 1
        if n <= 16:
            assert is_y_feasible_crt(y, -1, n)
            for q in sp.factorint(y): assert not is_y_feasible_crt(y // q, -1, n)
        if n <= 9: assert all(not is_y_feasible_crt(b, -1, n) for b in range(1, y))
    yk_d5 = [next(y for y in range(1, 500) if is_y_feasible_crt(y, -5, n))
             for n in range(1, 13)]
    assert yk_d5 == [1, 2, 1, 4, 10, 2, 2, 8, 4, 20, 220, 220]
    assert binom_theta(-10, 4, -3, 4) == (Fraction(931, 3), Fraction(-1463, 3))
    assert binom_theta(-2, 4, -3, 4) == (Fraction(-5, 1), Fraction(11, 1))
    for n in range(1, 16):
        k, sh = A392059_def(n), 1 << (n.bit_length() - 1)
        xs = n - sh; r0, r1 = binom_theta(xs, k, -1, n)
        assert r0.denominator == 1 and r1.denominator == 1
        assert (int(r0) - math.comb(xs, n) + int(r1)) % 2 == 1
        for m in range(1, n + 1):
            u0, u1 = binom_theta(0, 2 * k, -1, m)
            assert u0.denominator == 1 and u1.denominator == 1
            assert (int(u0) + int(u1)) % 2 == 0
        if n <= 8:
            for c in range(n + 1):
                for x in range(-8, 10):
                    rc, ic = binom_theta(x, k, -1, c)
                    assert rc.denominator == 1 and ic.denominator == 1
                    assert ((int(rc) + int(ic)) % 2) == (int(sp.binomial(x + sh, c)) % 2)
    print("[PASS] A387569 suff n<=27, prime-del n<=16, min-y n<=9; "
          "Z[sqrt(-5)] n<=12; A392059 n<=15.")

def verify_combinatorics():
    def held_karp(N, th):
        M = 1 << N; dp = np.zeros((M, N), dtype=np.int64)
        for u in range(N): dp[1 << u, u] = 1
        adj = [[v for v in range(N) if abs(u - v) > th] for u in range(N)]
        pop = np.array([bin(i).count("1") for i in range(M)], dtype=np.uint8)
        for s in range(1, N):
            masks = np.nonzero(pop == s)[0]
            for u in range(N):
                col = dp[masks, u]; nz = np.nonzero(col)[0]
                if not len(nz): continue
                ms, ws = masks[nz], col[nz]
                for v in adj[u]:
                    b = 1 << v; ok = (ms & b) == 0
                    if np.any(ok): dp[ms[ok] | b, v] += ws[ok]
        return int(dp[M - 1].sum())

    def mitm_odd_vec(N, th):
```

```
    # Split odd N = 2L-1 at midpoint u = p(L); reflection v <-> N-1-v halves midpoint loop
    L = N // 2 + 1
    adj = [[v for v in range(N) if abs(u - v) > th] for u in range(N)]
    cur_m = [np.array([1 << u], dtype=np.uint32) for u in range(N)]
    cur_w = [np.array([1], dtype=np.int64) for u in range(N)]
    for _ in range(2, L + 1):
        in_m, in_w = [[] for _ in range(N)], [[] for _ in range(N)]
        for u in range(N):
            ms = cur_m[u]
            if not len(ms): continue
            ws = cur_w[u]
            for v in adj[u]:
                bit = np.uint32(1 << v); ok = (ms & bit) == 0
                if np.any(ok):
                    in_m[v].append(ms[ok] | bit); in_w[v].append(ws[ok])
        for v in range(N):
            if not in_m[v]:
                cur_m[v] = np.empty(0, dtype=np.uint32)
                cur_w[v] = np.empty(0, dtype=np.int64)
            else:
                am, aw = np.concatenate(in_m[v]), np.concatenate(in_w[v])
                od = np.argsort(am, kind='mergesort'); am, aw = am[od], aw[od]
                df = np.empty(len(am), dtype=bool); df[0] = True; df[1:] = am[1:] != am[:-1]
                ix = np.nonzero(df)[0]; cur_m[v] = am[ix]; cur_w[v] = np.add.reduceat(aw, ix)
    full = np.uint32((1 << N) - 1); tot = 0
    for u in range((N + 1) // 2):
        mult = 2 if u < N - 1 - u else 1; ms, ws = cur_m[u], cur_w[u]
        if not len(ms): continue
        cp = (full ^ ms) | np.uint32(1 << u)
        ps = np.clip(np.searchsorted(ms, cp), 0, len(ms) - 1); ht = (ms[ps] == cp)
        tot += mult * int(np.sum(ws[ht] * ws[ps[ht]]))
    return tot

imap = {('U','U'):3, ('U','Va'):4, ('U','Vf'):5, ('E','U'):6,
        ('Va','Va'):7, ('Va','Vf'):8, ('E','Va'):9, ('Vf','Vf'):10, ('E','Vf'):11}
def add_c(a, t1, t2):
    p = (t1, t2) if t1 <= t2 else (t2, t1)
    if p == ('E', 'E'): return False
    a[imap[p]] += 1; return True

def build_u0(k):
    states = {(0, k - 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0): 1}
    for r in range(1, k + 1):
        if r >= 2:
            nu = defaultdict(int)
            for st, w in states.items():
                nu[(st[0]+1,) + st[1:]] += w
                if st[12] < 2: a = list(st); a[6] += 1; a[12] += 1; nu[tuple(a)] += w
            states = nu
        nxt = defaultdict(int)
        for st, ways in states.items():
            qU, qVa, qVf, UU, UVa, UVf, UE, VaVa, VaVf, VaE, VfVf, VfE, et = st
            if et < 2:
                for mu, di, oe in [(qU,0,'U'),(qVa,1,'Va'),(2*UU,3,'U'),(UVa,4,'Va'),
                                   (UVf,5,'Vf'),(UVa,4,'U'),(2*VaVa,7,'Va'),(VaVf,8,'Vf')]:
                    if mu: a = list(st); a[di] -= 1; a[12] += 1; add_c(a,'E',oe); nxt[tuple(a)] += ways*mu
            cats = [(0,qU,0,'U'),(1,qVa,1,'Va'),(2,2*UU,3,'U'),(3,UVa,4,'Va'),(4,UVf,5,'Vf'),
                    (5,UE,6,'E'),(6,UVa,4,'U'),(7,2*VaVa,7,'Va'),(8,VaVf,8,'Vf'),(9,VaE,9,'E')]
            for i in range(10):
                c1, n1, i1, o1 = cats[i]
                if not n1: continue
                if n1 >= 2 and o1 != 'E':
                    w2 = 2*UU*(UU-1) if c1==2 else (2*VaVa*(VaVa-1) if c1==7 else n1*(n1-1)//2)
                    if w2:
                        a = list(st); a[i1] -= 2
```

```
                        if add_c(a, o1, o1): nxt[tuple(a)] += ways * w2
                for j in range(i + 1, 10):
                    c2, n2, i2, o2 = cats[j]
                    if not n2 or (o1 == 'E' and o2 == 'E'): continue
                    w12 = UVa*(UVa-1) if (c1==3 and c2==6) else n1*n2
                    if w12:
                        a = list(st); a[i1] -= 1; a[i2] -= 1
                        if add_c(a, o1, o2): nxt[tuple(a)] += ways * w12
        states = nxt
        NVa = k - r
        if NVa >= 1:
            ne = defaultdict(int)
            for st, ways in states.items():
                qU, qVa, qVf, UU, UVa, UVf, UE, VaVa, VaVf, VaE, VfVf, VfE, _ = st
                sVa = NVa - qVa - (UVa + 2*VaVa + VaVf + VaE)
                if sVa > 0:
                    assert (ways * sVa) % NVa == 0
                    ne[st] += (ways * sVa) // NVa
                for cnt, d1, d2 in [(qVa,1,2),(UVa,4,5),(2*VaVa,7,8),(VaVf,8,10),(VaE,9,11)]:
                    if cnt > 0:
                        assert (ways * cnt) % NVa == 0
                        a = list(st); a[d1] -= 1; a[d2] += 1
                        ne[tuple(a)] += (ways * cnt) // NVa
            states = ne
    out = defaultdict(int)
    for st, ways in states.items():
        qU, qVa, qVf, UU, UVa, UVf, UE, VaVa, VaVf, VaE, VfVf, VfE, et = st
        assert qVa == 0 and UVa == 0 and VaVa == 0 and VaVf == 0 and VaE == 0
        cu = qU + UU; hd = (2*cu+UVf+UE) - (2*qVf+2*VfVf+UVf+VfE); re = 2 - et
        if abs(hd) <= re and (hd - re) % 2 == 0:
            out[(cu, UVf, UE, qVf, VfVf, VfE, et)] += ways * (1 << UU)
    return dict(out)

def build_u0_labeled(k):
    # dg[x] in {0,1,2}: consumed slot capacity deg_C(x) + is_endpoint(x); 2 = closed
    nb = 2 * k - 2
    st = {(tuple([0]*nb), tuple(range(nb)), 0): 1}
    for r in range(1, k + 1):
        nb_r = list(range(r - 1)) + list(range(k - 2 + r, nb))
        nx = defaultdict(int)
        for (dg, mt, ec), w in st.items():
            if ec < 2:
                for x in nb_r:
                    if dg[x] == 0:
                        d2, m2 = list(dg), list(mt); d2[x] = 1; m2[x] = -1
                        nx[(tuple(d2), tuple(m2), ec + 1)] += w
                    elif dg[x] == 1 and mt[x] != -1:
                        ox = mt[x]; d2, m2 = list(dg), list(mt)
                        d2[x] = 2; m2[x] = -2; m2[ox] = -1
                        nx[(tuple(d2), tuple(m2), ec + 1)] += w
            for i, x in enumerate(nb_r):
                if dg[x] == 2: continue
                for y in nb_r[i+1:]:
                    if dg[y] == 2 or (dg[x] == 1 and mt[x] == y): continue
                    ex = x if dg[x] == 0 else mt[x]
                    ey = y if dg[y] == 0 else mt[y]
                    if ex == -1 and ey == -1: continue
                    d2, m2 = list(dg), list(mt); d2[x] += 1; d2[y] += 1
                    if dg[x] == 1: m2[x] = -2
                    if dg[y] == 1: m2[y] = -2
                    if ex >= 0: m2[ex] = ey
                    if ey >= 0: m2[ey] = ex
                    nx[(tuple(d2), tuple(m2), ec)] += w
        st = nx
    out = defaultdict(int)
```

```
        for (dg, mt, ec), w in st.items():
            cs = [i for i in range(k - 1) if dg[i] < 2]
            eps = [()] + ([(x,) for x in cs] if ec <= 1 else [])
            if ec == 0:
                eps += [(cs[a], cs[b]) for a in range(len(cs)) for b in range(a+1, len(cs))]
            for ep in eps:
                d2, m2, e2, ok = list(dg), list(mt), ec, True
                for x in ep:
                    if d2[x] == 0: d2[x] = 1; m2[x] = -1; e2 += 1
                    elif m2[x] == -1: ok = False; break
                    else: ox = m2[x]; d2[x] = 2; m2[x] = -2; m2[ox] = -1; e2 += 1
                if not ok: continue
                qU = sum(1 for i in range(k - 1) if d2[i] == 0)
                qV = sum(1 for j in range(k - 1, nb) if d2[j] == 0)
                uu = sum(1 for i in range(k - 1) if d2[i] == 1 and 0 <= m2[i] < k - 1) // 2
                vv = sum(1 for j in range(k - 1, nb) if d2[j] == 1 and m2[j] >= k - 1) // 2
                uv = sum(1 for i in range(k - 1) if d2[i] == 1 and m2[i] >= k - 1)
                eu = sum(1 for i in range(k - 1) if d2[i] == 1 and m2[i] == -1)
                ev = sum(1 for j in range(k - 1, nb) if d2[j] == 1 and m2[j] == -1)
                cu = qU + uu; hd = (2*cu + uv + eu) - (2*qV + 2*vv + uv + ev); re = 2 - e2
                if abs(hd) <= re and (hd - re) % 2 == 0:
                    out[(cu, uv, eu, qV, vv, ev, e2)] += w * (1 << uu)
        return dict(out)

    def v_sat(cu, uvf, ue, qvf, vfvf, vfe, et, w, nxt, dec_q, can_cl):
        cl = 0; qn = qvf - (1 if dec_q else 0)
        if et < 2:
            if cu > 0: nxt[(cu-1, uvf, ue+1, qn, vfvf, vfe, et+1)] += w * cu
            if uvf > 0: nxt[(cu, uvf-1, ue, qn, vfvf, vfe+1, et+1)] += w * uvf
            if can_cl and ue == 1 and cu == 0 and uvf == 0 and et == 1: cl += w
        if cu >= 2: nxt[(cu-1, uvf, ue, qn, vfvf, vfe, et)] += w * cu * (cu - 1)
        if uvf >= 2: nxt[(cu, uvf-2, ue, qn, vfvf+1, vfe, et)] += w * (uvf*(uvf-1)//2)
        if cu >= 1 and uvf >= 1: nxt[(cu-1, uvf, ue, qn, vfvf, vfe, et)] += w * cu * uvf
        if cu >= 1 and ue >= 1: nxt[(cu-1, uvf, ue, qn, vfvf, vfe, et)] += w * cu * ue
        if uvf >= 1 and ue >= 1: nxt[(cu, uvf-1, ue-1, qn, vfvf, vfe+1, et)] += w * uvf * ue
        if can_cl and ue == 2 and cu == 0 and uvf == 0 and et == 2: cl += w
        return cl

    def close_vr(work):
        tot = 0
        while work:
            nxt = defaultdict(int)
            for (cu, uvf, ue, qvf, vfvf, vfe, et), w in work.items():
                rem = qvf + 2*vfvf + uvf + vfe
                if not rem: continue
                last = (rem == 1)
                if qvf > 0: tot += v_sat(cu, uvf, ue, qvf, vfvf, vfe, et, w, nxt, True, last)
                elif vfe > 0:
                    if last and et == 1 and cu == 0 and uvf == 0 and ue == 0: tot += w
                    if cu > 0: nxt[(cu-1, uvf, ue+1, 0, vfvf, vfe-1, et)] += w * cu
                    if uvf > 0: nxt[(cu, uvf-1, ue, 0, vfvf, vfe, et)] += w * uvf
                    if last and ue == 1 and cu == 0 and uvf == 0 and et == 2: tot += w
                elif vfvf > 0:
                    if et < 2: nxt[(cu, uvf, ue, 0, vfvf-1, vfe+1, et+1)] += w
                    if cu > 0: nxt[(cu-1, uvf+1, ue, 0, vfvf-1, vfe, et)] += w * cu
                    if uvf > 0: nxt[(cu, uvf-1, ue, 0, vfvf, vfe, et)] += w * uvf
                    if ue > 0: nxt[(cu, uvf, ue-1, 0, vfvf-1, vfe+1, et)] += w * ue
                else:
                    if et < 2: nxt[(cu, uvf-1, ue+1, 0, 0, 0, et+1)] += w
                    if cu > 0: nxt[(cu, uvf-1, ue, 0, 0, 0, et)] += w * cu * 2
                    if uvf >= 2: nxt[(cu, uvf-1, ue, 0, 0, 0, et)] += w * (uvf - 1)
                    if ue > 0: nxt[(cu, uvf-1, ue, 0, 0, 0, et)] += w * ue
            work = nxt
        return 2 * tot
```

```
def trans(st):
    if st == ('CLOSED',): return {}
    cu0, uvf0, ue0, qvf0, vfvf0, vfe0, et0 = st
    au = [(cu0+1, uvf0, ue0, qvf0, vfvf0, vfe0, et0, 1)]
    if et0 < 2: au.append((cu0, uvf0, ue0+1, qvf0, vfvf0, vfe0, et0+1, 1))
    out = defaultdict(int)
    for cu, uvf, ue, qvf, vfvf, vfe, et, wu in au:
        vr = qvf + 2*vfvf + uvf + vfe
        cl = v_sat(cu, uvf, ue, qvf, vfvf, vfe, et, wu, out, False, vr == 0)
        if cl: out[('CLOSED',)] += cl
    return {ns: w for ns, w in out.items() if w and (ns == ('CLOSED',) or
            (abs((2*ns[0]+ns[1]+ns[2]) - (2*ns[3]+2*ns[4]+ns[1]+ns[5])) <= 2-ns[6]
             and ((2*ns[0]+ns[1]+ns[2]) - (2*ns[3]+2*ns[4]+ns[1]+ns[5]) - (2-ns[6])) % 2 == 0))}

dims_exp = {2:1, 3:12, 4:31, 5:55, 6:79, 7:114, 8:147, 9:193}
for k in range(2, 10):
    u0 = build_u0(k)
    if k <= 5: assert u0 == build_u0_labeled(k)
    vis, q, adj = set(u0.keys()), list(u0.keys()), {}
    while q:
        s = q.pop(); tr = trans(s); adj[s] = tr
        for ns in tr:
            if ns not in vis: vis.add(ns); q.append(ns)
    sl = sorted(vis, key=str); idx = {s: i for i, s in enumerate(sl)}; D = len(sl)
    assert D == dims_exp[k]
    wv = [2 if s == ('CLOSED',) else close_vr({s: 1}) for s in sl]
    uv = [u0.get(s, 0) for s in sl]
    spT = [[(idx[ns], wt) for ns, wt in adj[s].items()] for s in sl]
    roots = [(C + 1) * (k - 1 - C) for C in range((k - 1) // 2 + 1)]
    z = list(wv)
    for lam in roots:
        Tz = [sum(wt * z[j] for j, wt in spT[i]) for i in range(D)]
        z = [Tz[i] - lam * z[i] for i in range(D)]
    cu_v = list(uv); seq_k = []
    for m_s in range(max(D + 2, 13)):
        assert sum(z[i] * cu_v[i] for i in range(D)) == 0
        if m_s <= 12: seq_k.append(sum(wv[i] * cu_v[i] for i in range(D)))
        nx = [0] * D
        for i, val in enumerate(cu_v):
            if val:
                for j, wt in spT[i]: nx[j] += wt * val
        cu_v = nx
    if k <= 6:
        for m_s in range(len(seq_k)):
            nv = k - 1 + m_s
            if 2 * nv + k <= 16:
                assert seq_k[m_s] == held_karp(2 * nv + k, nv)
    for di in range(len(roots)):
        zs = list(wv)
        for lam in (roots[:di] + roots[di+1:]):
            Tz = [sum(wt * zs[j] for j, wt in spT[i]) for i in range(D)]
            zs = [Tz[i] - lam * zs[i] for i in range(D)]
        ct = list(uv); van = True
        for _ in range(10):
            if sum(zs[i] * ct[i] for i in range(D)) != 0: van = False; break
            nx = [0] * D
            for i, val in enumerate(ct):
                if val:
                    for j, wt in spT[i]: nx[j] += wt * val
            ct = nx
        assert not van
    if k == 9:
        a_seq = ([held_karp(2*n+9, n) for n in range(5)]
                 + [mitm_odd_vec(2*n+9, n) for n in (5, 6, 7)])
        a_seq.extend(seq_k)
```

```
            den = [1, -80, 2508, -38336, 283840, -806400]
            num_exp = [362880, -23733610, 574782136, -6066574082, 23063767496, 1306804904,
                       19195078240, 99691196928, 5798373293056, 140546260769280,
                       -1838864889858048, 0, 0]
            assert [sum(a_seq[n-j]*den[j] for j in range(min(n+1, 6))) for n in range(13)] == num_exp
    print("[PASS] Stage-1 labeled check k<=5, DP/MITM n<=7, bulk DP 2n+k<=16, & Krylov D_k<=193.")

if __name__ == "__main__":
    verify_number_theory()
    verify_combinatorics()
```