SUBCUBE PARTITIONS OF THE VERTEX SET OF THE n-CUBE: A TWO-SLICE RECURSION, THE CUBICAL BELL NUMBERS, AND THE SEVENTH TERM OF A018926

KOREA SUPERINTELLIGENCE LABS

Abstract. Let $f(n)$ be the number of ways to partition the vertex set of the n-cube into subcubes — OEIS A018926, the function $f(d)$ of Alon, Balogh and Potapov, and the $(n+1)$-st cubical Bell number $B^{(c)}_{n+1}$ of Duarte and Solus, which counts the stagings of a level of a CStree. Six terms have been known since 2019, $f(0), \dots, f(5) = 1, 2, 8, 154, 89512, 71319425714$, and the sequence carries the OEIS keyword `more`; Alon, Balogh and Potapov ask for the function outright, and Duarte and Solus print a question mark in the corresponding cell of their table of CStree counts. We prove a two-slice splitting identity for the number of subcube partitions of an *arbitrary* vertex set of an arbitrary cube, together with the two-level form it generates, and we derive $f(0), \dots, f(5)$ from the definition through it; for $f(5) = 71319425714$ this is, as far as we can determine, the first derivation from the definition and the first independent confirmation of the value. The identity is what makes the next term reachable: it replaces an enumeration of $2.16 \cdot 10^{23}$ objects, and a table over the 2^{32} vertex sets of the 5-cube, by $2 \cdot 5^{16} \approx 3.05 \cdot 10^{11}$ integer operations over a 512 KB table. Carrying that computation out gives

$$f(6) \;=\; 216\,009\,178\,641\,968\,077\,129\,336,$$

the number of subcube partitions of the vertex set of the 6-cube and the seventh cubical Bell number, and hence $7! \prod_{k=1}^{7} B^{(c)}_k \approx 1.71251 \cdot 10^{46}$ CStrees on seven binary variables. The value rests on four independent exact evaluations and is reported here outside the formally verified development, which contains the identities, the values up to $f(5)$, and a machine-checked evaluation of the seventh term's defining sum in its bitmask form. All theorems below are machine-checked in Lean 4.

1. Introduction

The objects. Write Q_n for the n-dimensional cube and $V_n = \{0,1\}^n$ for its vertex set. A *subcube* (a face of Q_n) is obtained by fixing some coordinates to 0 or 1 and leaving the others free; it is specified by a word $c \in \{0, 1, *\}^n$, and its *cell* $[c] \subseteq V_n$ is the set of vertices agreeing with c on every fixed coordinate. There are 3^n subcubes, and every cell is nonempty. A *subcube partition* of a vertex set $X \subseteq V_n$ is a set Π of subcubes whose cells are pairwise disjoint and have union exactly X. Let

$$\mathrm{P}(X) \;=\; \#\{\text{subcube partitions of } X\}, \qquad f(n) \;=\; \mathrm{P}(V_n).$$

Since cells are nonempty and distinct subcubes have distinct cells, a set of subcubes is faithfully the same thing as a partition of X into faces, so no over-counting is hidden in the choice of model; both facts are part of the verified development (Section 2).

The sequence $f(n)$ is OEIS A018926 [7], whose name reads "Number of ways to partition an n-cube into subcubes (without reducing isomorphic cases); number of labeled squashed n-cubes; ways to make a tautology from disjoint terms with n Boolean variables; number of Rivest compression schemes for n-bit retrieval." The entry cites no source for any of the four descriptions in that name; the last of them we have not traced to a primary reference, and nothing below uses it. The sequence arises independently in at least three places.

Alon, Balogh and Potapov [1] let "$f(d)$ be the number of partitions (tilings) of the vertex set of Q_d in which each of the classes spans a subhypercube" — our $f(d)$ exactly — and pose, as their Problem 1.1, "Determine or estimate the functions (i) $m(d)$, (ii) $m'(d)$, (iii) $f(d)$, (iv) $f_{\leq 2}(d)$", recording that

"Problem (iii) was raised by Gadouleau at the 29th British Combinatorial Conference in 2022".[1] Their own results on f are asymptotic: the paper contains no exact value, no recursion and no generating function.

Duarte and Solus [4], in a paper on context-specific causal models, "define the $(p+1)^{st}$ cubical Bell number to be the number of ways to partition the vertices of $[0,1]^p$ into non-overlapping faces of the cube", so that $B^{(c)}_{p+1} = f(p)$, and prove that "the number of CStrees on p binary variables is $p!\prod_{k=1}^{p} B^{(c)}_k$". They note that "the cubical Bell numbers are known only for small values [7]. For $p = 1, \ldots, 6$ they are $1, 2, 8, 154, 89512, 71319425714$", print a question mark in the CStree column of their table of counts — Figure 11 of v4, captioned "Number of DAGs, CStrees and compatibly labeled staged trees on p binary variables" — at $p = 7$ and $p = 8$, and close the discussion with "it would be of interest to have a closed-form formula for $B^{(c)}_p$".

Bridoux, Gadouleau and Theyssier [2] observe that "the number $A(n)$ of globally commutative, bijective Boolean networks is equal to the number of partitions of the cube $\{0,1\}^n$ into subcubes. This, in turn, is equal to the number of minimally unsatisfiable cnfs on n variables."

The published state of the sequence is six terms,

$$f(0), \ldots, f(5) \;=\; 1,\ 2,\ 8,\ 154,\ 89512,\ 71319425714,$$

with the OEIS keyword `more`. A018926 attributes only the last of the six: its single extensions line reads "a(5) from Sean A. Irvine, Feb 21 2019". For $f(0), \ldots, f(4)$ it names no computation at all — the entry's revision history shows it created in December 1996 already carrying exactly those five terms, and the only source it cites for them is "D. E. Knuth, personal communication". Its b-file is not a contributed table: it is headed "b-file synthesized from sequence entry", stops at $n = 5$, and is therefore no evidence beyond the entry itself.

What is settled here. *(1) A recursion.* Cutting Q_{n+1} along its last coordinate gives an identity for P on *every* vertex set, not only on the full cube (Theorem 3.1), and iterating it once gives a two-level form (Theorem 3.4) that expresses $f(n+2)$ entirely through P on subsets of V_n. Both hold for every n and are proved without any appeal to computation.

(2) The known values, from the definition. Evaluating the recursion gives $f(0), \ldots, f(4)$ (Theorem 4.1) and, through a table over the 65536 subsets of V_4 that is checked entry by entry against the recursion itself, $f(5) = 71319425714$ (Theorem 4.2). The values are not new. What appears to be new is that they are now derived from the definition of a subcube partition rather than enumerated, and that $f(5)$ — which had one contributor and no independent confirmation — has a second, structurally different derivation.

(3) The seventh term. Section 5 reports

$$f(6) \;=\; 216\,009\,178\,641\,968\,077\,129\,336 \;\approx\; 2.16009 \cdot 10^{23},$$

the first uncomputed case of Problem 1.1(iii) of [1] and the seventh cubical Bell number $B^{(c)}_7$, together with the resulting count of CStrees on seven binary variables,

$$7!\prod_{k=1}^{7} B^{(c)}_k \;=\; 17\,125\,077\,428\,476\,440\,507\,901\,823\,199\,935\,254\,770\,103\,418\,880 \;\approx\; 1.71251 \cdot 10^{46},$$

which fills the $p = 7$ cell of the table of [4]. These two values were obtained by computation and are stated as remarks lying outside the formally verified development; what *is* inside it, and machine-checked, is the exact bitmask sum whose value they are (Proposition 5.1), evaluated on a table that is itself certified against the recursion. Section 5 says precisely which link is missing.

[1] That conference was held at Lancaster, 11–15 July 2022. We have found no published problem list for it — the British Combinatorial Committee's archive links one only for the 30th conference — so the attribution rests on [1] alone, whose acknowledgements thank Maximilien Gadouleau by name.

Why the recursion is the point. The cost of $f(6)$ is what the recursion changes. A direct enumerator visits $2.16 \cdot 10^{23}$ partitions one at a time; the one-step recursion needs P on all 2^{32} subsets of V_5, a table of $4.3 \cdot 10^9$ entries, which at 64 bits an entry is 34 GB. The two-level form needs only the 2^{16}-entry table over V_4 — 512 KB — and $2 \cdot 5^{16} = 305\,175\,781\,250$ integer operations. That is a computation of a few CPU-hours, and it is the whole distance between the sixth term and the seventh.

Prior art, and how it was searched. The identities of Section 3 are elementary, and we claim only that we did not find them. The search was as follows, all of it carried out on 2026-08-29. A local corpus of 881 145 arXiv papers was swept for `A018926`, *cubical Bell*, *squashed cube*, *Rivest*, *CStree*, *staged tree*, and for the decimal strings 71319425714, 89512, 26466908160, 59136 and 11325628142378282557440: exactly three papers of the 881 145 mention A018926, namely [2], [4] and [1], and none of them states a recursion or an exact value beyond the published six. The live versions of [4] and [1] were downloaded and read in full rather than taken from the corpus copy. OEIS was queried live for the identifier, for the subsequences $1, 2, 8, 154, 89512$ and $154, 89512, 71319425714$, and for *squashed cube*, *Rivest* and *cubical Bell*; the first five queries return A018926 and nothing else, and A018926 carries no formula and no recursion. The forward citations of [4] were taken from OpenAlex (three work records, five citing papers in total, all in causal discovery or toric algebra; the nearest, arXiv:2402.07762, gives a closed formula only for the subclass of CStrees whose stages use at most two context variables).[2] Finally, the only program known to have produced a term of this sequence, Irvine's `A018926.java` in the *jOEIS* library [5], was read: it describes itself as "Uses the sub-cubes of n-cubes definition. Brute force, does every arrangement making no attempt to account for possible symmetries", it enumerates partitions one at a time, and it accumulates into a Java `long`, which $2.16 \cdot 10^{23}$ overflows.[3] A negative search result is not a proof of novelty, and we state these as "not found".

2. Preliminaries

Throughout, $n \geq 0$, $V_n = \{0,1\}^n$, and a subcube of Q_n is a word $c \in \{0,1,*\}^n$ with cell

$$[c] \;=\; \{v \in V_n \;:\; c_i \neq * \Rightarrow v_i = c_i \;\text{ for all } i\}.$$

Definition 2.1. For $X \subseteq V_n$, a *subcube partition* of X is a finite set Π of subcubes of Q_n such that $\bigcup_{c\in\Pi}[c] = X$ and $[c] \cap [d] = \emptyset$ for all distinct $c, d \in \Pi$. We write $\mathrm{P}(X)$ for the number of subcube partitions of X and $f(n) = \mathrm{P}(V_n)$.

Two elementary facts make Definition 2.1 the intended one. Every cell is nonempty — the word obtained by replacing each $*$ by 0 lies in it — so no block of a subcube partition is empty; and $c \mapsto [c]$ is injective, since a coordinate is fixed to b precisely when it is constant equal to b on the cell. Consequently counting *sets of subcubes* and counting *partitions of X into faces* give the same number, and $\mathrm{P}(\emptyset) = 1$. Both facts are proved in the verified development, and the injectivity is also checked directly at $n = 2$ as a control (Proposition 4.4).

Finally we fix the slicing notation used throughout. A vertex of Q_{n+1} is written (y, b) with $y \in V_n$ and $b \in \{0,1\}$ the last coordinate. For $X \subseteq V_{n+1}$ put

$$X_b \;=\; \{y \in V_n \;:\; (y,b) \in X\} \subseteq V_n \qquad (b = 0, 1),$$

the two *slices* of X, and write $\overline{A} = V_n \setminus A$ for complements inside V_n.

3. The two-slice recursion

A subcube c of Q_{n+1} either fixes the last coordinate — and then $[c]$ lies inside one slice, and c is a subcube of Q_n carried into that slice — or leaves it free, and then $[c] = [c'] \times \{0,1\}$ for a subcube c' of Q_n. Call the cubes of the second kind *spanning*. This dichotomy is the whole content of the following theorem.

[2] Semantic Scholar returned an empty citation list once and then refused four further attempts, so the forward-citation evidence here is OpenAlex's alone.

[3] The program is `src/irvine/oeis/a018/A018926.java` in that repository, read on 2026-08-29; the quoted sentence is the comment that describes its treatment of symmetry, and is not the file's only comment.

Theorem 3.1 (The two-slice recursion)**.** *For every $n \geq 0$ and every $X \subseteq V_{n+1}$,*

$$\mathrm{P}(X) \;=\; \sum_{A \,\subseteq\, X_0 \cap X_1} \mathrm{P}(A) \cdot \mathrm{P}(X_0 \setminus A) \cdot \mathrm{P}(X_1 \setminus A).$$

Proof sketch. Given a subcube partition Π of X, let $A(\Pi) \subseteq V_n$ be the union of the projections of the spanning cubes of Π. Every such projection is contained in both slices, so $A(\Pi) \subseteq X_0 \cap X_1$, and the map $\Pi \mapsto A(\Pi)$ fibres the set of subcube partitions of X over the subsets A of $X_0 \cap X_1$. For a fixed A, send Π to the triple consisting of the projections of its spanning cubes, of its cubes fixed at 0, and of its cubes fixed at 1. The three components are subcube partitions of A, of $X_0 \setminus A$ and of $X_1 \setminus A$ respectively, and the inverse map assembles a triple of partitions by lifting the three families with last coordinate free, 0 and 1. The two maps are mutually inverse for purely formal reasons — a subcube of Q_{n+1} is exactly a pair (subcube of Q_n, value in $\{0, 1, *\}$) — and what has to be checked is that assembly produces a partition of X and disassembly produces partitions of the three sets; both are direct verifications from Definition 2.1. Summing the fibre sizes gives the identity. No case distinction on n and no computation of any kind enters. □

Taking $X = V_{n+1}$, whose slices are both all of V_n, gives the recursion for the sequence itself.

Corollary 3.2. *For every $n \geq 0$,*

$$f(n+1) \;=\; \sum_{A \subseteq V_n} \mathrm{P}(A) \cdot \mathrm{P}(\overline{A})^2.$$

Corollary 3.2 alone still asks for P on all subsets of V_n, i.e. a table of 2^{2^n} entries; for $f(6)$ that is 2^{32} entries and out of reach. The useful form applies the recursion twice. The mechanism is that a vertex set of Q_{n+1} *is* its pair of slices.

Lemma 3.3. *The slice map $X \mapsto (X_0, X_1)$ is a bijection from the subsets of V_{n+1} to the pairs of subsets of V_n; writing $\langle X_0, X_1 \rangle$ for its inverse and*

$$g(X_0, X_1) \;=\; \sum_{A \subseteq X_0 \cap X_1} \mathrm{P}(A) \cdot \mathrm{P}(X_0 \setminus A) \cdot \mathrm{P}(X_1 \setminus A),$$

one has $\mathrm{P}(\langle X_0, X_1 \rangle) = g(X_0, X_1)$ and $\overline{\langle X_0, X_1 \rangle} = \langle \overline{X_0}, \overline{X_1} \rangle$, the first complement taken in V_{n+1} and the other two in V_n.

Theorem 3.4 (The two-level form)**.** *For every $n \geq 0$,*

$$f(n+2) \;=\; \sum_{X_0 \subseteq V_n} \; \sum_{X_1 \subseteq V_n} g(X_0, X_1) \cdot g(\overline{X_0}, \overline{X_1})^2,$$

where g is as in Lemma 3.3; in particular $f(n+2)$ is determined by the values of P *on the subsets of V_n alone.*

Proof sketch. Apply Corollary 3.2 at level $n+1$ and reindex the sum over $A \subseteq V_{n+1}$ along the bijection of Lemma 3.3; the summand $\mathrm{P}(A)\mathrm{P}(\overline{A})^2$ becomes $g(X_0, X_1)\, g(\overline{X_0}, \overline{X_1})^2$ by the two displayed identities of that lemma. As with Theorem 3.1, nothing is computed. □

Remark 3.5 (Cost). Theorem 3.4 at $n = 4$ is the exact sum evaluated in Section 5. Its cost is easy to read off. Identify subsets of V_4 with 16-bit masks; for a fixed pair (X_0, X_1) the inner sum of g has $2^{|X_0 \cap X_1|}$ terms, so $\sum_{X_0, X_1} 2^{|X_0 \cap X_1|} = 5^{16}$, and the two evaluations of g needed per pair bring the total to $2 \cdot 5^{16} \approx 3.05 \cdot 10^{11}$ operations on a table of 2^{16} entries. The one-step Corollary 3.2 at $n = 5$ would instead need the 2^{32}-entry table, and a direct enumeration would visit $f(6) \approx 2.16 \cdot 10^{23}$ objects.

4. The values up to $f(5)$, from the definition

Theorem 3.1 turns into a computable function by recursion on the dimension: P on V_0 is identically 1, and P at level $n+1$ is the displayed sum over the subsets of $X_0 \cap X_1$. Unmemoised, the cost of level $n+1$ is about $2^{2^n} \cdot 3$ times the cost of level n, which is a tower: about $2.6 \cdot 10^6$ products at $n = 4$ and about $5 \cdot 10^{11}$ at $n = 5$. So the naive evaluation reaches exactly $f(4)$.

Theorem 4.1. *$f(0) = 1$, $f(1) = 2$, $f(2) = 8$, $f(3) = 154$ and $f(4) = 89512$; equivalently $B_1^{(c)}, \dots, B_5^{(c)} = 1, 2, 8, 154, 89512$.*

Proof sketch. Each value is the evaluation of the recursion of Theorem 3.1 at the full vertex set, the recursion having first been proved equal to the definitional count for all n and all X. The values at $n \le 2$ are obtained by kernel reduction; those at $n = 3$ and $n = 4$ by compiled evaluation of the same function, the finite object traversed being the tower of powerset sums described above, some $2.6 \cdot 10^6$ products at $n = 4$. No table and no transcribed constant enters: the only inputs are Definition 2.1 and Theorem 3.1. □

For $f(5)$ the recursion has to be tabulated, and then the table becomes the thing to be trusted. We avoid trusting it. Encode a subset $A \subseteq V_k$ by the 2^k-bit mask $m(A) = \sum_{v \in A} 2^{\mathrm{code}(v)}$, where $\mathrm{code}(v) = \sum_i v_i 2^i$, and let T_k be an array of 2^{2^k} natural numbers, built by fast mask arithmetic whose correctness is *not* asserted. Define the predicate

$$\mathrm{ok}(T_{k+1}, T_k) \ : \ \text{ for every } A \subseteq V_{k+1}, \quad T_{k+1}[m(A)] = \sum_{B \subseteq A_0 \cap A_1} T_k[m(B)] \cdot T_k[m(A_0 \setminus B)] \cdot T_k[m(A_1 \setminus B)].$$

If $T_k[m(A)] = \mathrm{P}(A)$ for all $A \subseteq V_k$ and $\mathrm{ok}(T_{k+1}, T_k)$ holds, then $T_{k+1}[m(A)] = \mathrm{P}(A)$ for all $A \subseteq V_{k+1}$, by Theorem 3.1. The point of this shape is that no property of the encoding is needed — not injectivity, not surjectivity, not any relation between masks and set operations. The check quantifies over vertex *sets* directly, so a wrong encoding convention, a wrong array layout or a wrong submask enumeration makes the check fail rather than the conclusion false.

Theorem 4.2. *$f(5) = 71319425714$, that is, $B_6^{(c)} = 71319425714$.*

Proof sketch. The check $\mathrm{ok}(T_{k+1}, T_k)$ is verified by exhaustive compiled evaluation for $k = 0, 1, 2, 3$, quantifying over all 4, 16, 256 and 65536 vertex sets of V_1, V_2, V_3 and V_4 respectively; the base $T_0[m(A)] = \mathrm{P}(A)$ for the two subsets of V_0 is decided by kernel reduction. Chaining the four steps certifies T_4, the 65536-entry table of P over the subsets of V_4. Then Corollary 3.2 at $n = 4$ is evaluated on T_4: the sum $\sum_{A \subseteq V_4} T_4[m(A)] \cdot T_4[m(\overline{A})]^2$ over all 65536 subsets is again an exhaustive compiled evaluation, and it returns 71319425714. Total cost 231 seconds. The exhaustive parts are therefore exactly these: four table checks over $4 + 16 + 256 + 65536$ vertex sets, and one outer sum of 65536 terms. □

Remark 4.3. The value $f(5) = 71319425714$ is published, and Theorem 4.2 is not a new number. What it adds is a derivation from Definition 2.1 rather than from an enumeration, and an independent confirmation: A018926 credits the term to a single contributor, its b-file is synthesized from the entry itself, and we found no second computation of it in the literature. The route here shares nothing with a partition enumerator except the definition.

Proposition 4.4 (Controls)**.** *The following are verified, each guarding a specific way the development could be vacuous or misaligned.*

(i) *$f(3) \notin \{153, 155\}$, and $f(3) \neq 4140$; the last is the Bell number B_8, so the count is not the number of set partitions of the eight vertices of Q_3.*

(ii) *$f(5) \notin \{71319425713, 71319425715\}$.*

(iii) *Being a subcube partition is neither vacuous nor trivial: the single free cube $**$ is a subcube partition of V_2; the empty family is not; and the pair $\{0*, \ **\}$, whose cells cover V_2 but overlap, is not.*

(iv) *$c \mapsto [c]$ is injective at $n = 2$, so distinct subcubes really are distinct blocks.*
(v) *The table check of Section 4 is not satisfied by an arbitrary array: perturbing a single entry of T_3 makes $\mathrm{ok}(T_3, T_2)$ fail.*
(vi) *The certified table reproduces the independently proved value of Theorem 4.1 at the full vertex set: $T_4[m(V_4)] = 89512$.*
(vii) *The two-level identity of Theorem 3.4 reproduces $f(3) = 154$ at $n = 1$, so the pairing of slices, the complement and the inner sum are oriented as the computation of Section 5 has them; and its summand g is not constant, so the identity is not collapsing to a single term.*
(viii) *The chunked sum of Proposition 5.1 is glued from its five pieces by an induction on ranges, not asserted.*

5. The seventh term

The sum, on masks. Fix the level-4 table T_4 certified in Section 4, and identify subsets of V_4 with 16-bit masks through $m(\cdot)$, writing $\wedge$, $\oplus$ for bitwise and, exclusive or, and $\overline{x} = x \oplus (2^{16} - 1)$. Put

$$F(x,y) \;=\; \sum_{a \,\subseteq\, x\wedge y} T_4[a]\cdot T_4[x\oplus a]\cdot T_4[y\oplus a], \qquad S \;=\; \sum_{x=0}^{2^{16}-1}\sum_{y=0}^{2^{16}-1} F(x,y)\cdot F(\overline{x},\overline{y})^2,$$

the sum over a running over the submasks of $x \wedge y$. This is Theorem 3.4 at $n = 4$ transported along the encoding.

Proposition 5.1. $S = 216\,009\,178\,641\,968\,077\,129\,336$. *More precisely, splitting the outer sum at $x \in \{0,\ 12128,\ 25967,\ 39570,\ 53409,\ 65536\}$, the five partial sums are*

$$\begin{array}{ll} 0 \le x < 12128: & 153\,521\,523\,175\,882\,770\,138\,007 \\ 12128 \le x < 25967: & 27\,918\,756\,004\,743\,293\,992\,314 \\ 25967 \le x < 39570: & 19\,407\,284\,860\,293\,318\,718\,588 \\ 39570 \le x < 53409: & 12\,267\,567\,235\,815\,939\,212\,710 \\ 53409 \le x < 65536: & 2\,894\,047\,365\,232\,755\,067\,717 \end{array}$$

and their sum is S.

Proof sketch. Each of the five partial sums is an exhaustive compiled evaluation of the displayed expression over its range of x; the finite computation performed is precisely $2 \cdot 5^{16} = 305\,175\,781\,250$ submask operations in total, in exact integer arithmetic with no bound on word size. Each summand $F(x,y)F(\overline{x},\overline{y})^2$ is one of the nonnegative terms of S, so it and every partial sum are at most $S < 2^{78}$ — well past the 2^{63} at which Lean's `Nat` becomes a heap-allocated bignum, so a substantial part of the running time below is bignum arithmetic in the outer sum. The boundaries were chosen to equalise the per-x cost $3^k 2^{16-k} + 3^{16-k} 2^k$, k the number of set bits of x, so each piece is about $6.1 \cdot 10^{10}$ operations; each took about 38 minutes, 3.17 CPU-hours in all, in 1.54 GB of memory. The five pieces are combined into S by an induction on ranges — that a sum over $[l, l+p+q)$ splits as a sum over $[l, l+p)$ plus a sum over $[l+p, l+p+q)$ — and not by asserting the total. $\square$

What Proposition 5.1 is, and what it is not. It is a verified evaluation, on certified input data, of the sum that Theorem 3.4 equates with $f(6)$ — but transported by hand onto bitmasks. The transport is the missing link. Making Proposition 5.1 into a proof that $f(6) = S$ requires, in addition to what is proved above, the level-4 bridge between sets and masks: that $m(X \cap Y) = m(X) \wedge m(Y)$ and $m(X \setminus Y) = m(X) \wedge \overline{m(Y)}$, that summing over subsets of V_4 is summing over $[0, 2^{16})$, and that the inner submask enumeration computes the sum over the subsets of $X_0 \cap X_1$. All of these follow from the single statement that the bit of $m(A)$ at position $\mathrm{code}(v)$ is set exactly when $v \in A$, together with the bijectivity of code onto $\{0, \dots, 15\}$; the work is bounded and specified, and it has not been carried out here. Until it is, the equality $f(6) = S$ is an equality between a proved sum and an evaluated one whose common value we have checked in four ways, and not a theorem. We are explicit about this because the alternative — quietly labelling S as $f(6)$ — is exactly the kind of step this sequence's history does not need.

Remark 5.2 (Outside the formal development: the value of $f(6)$). Four independent exact evaluations agree that

$$f(6) \;=\; 216\,009\,178\,641\,968\,077\,129\,336,$$

the number of subcube partitions of the vertex set of Q_6, the seventh cubical Bell number $B_7^{(c)}$, and the first uncomputed case of Problem 1.1(iii) of [1]. The four are:

(a) Theorem 3.4 at $n = 4$, evaluated in C with 128-bit integers: the table of P over the 65536 subsets of V_4 is built by the lowest-vertex recursion, and the double sum is then formed. About 22 CPU-minutes in under 20 MB. Run at base dimensions 1, 2 and 3 instead of 4, the identical code path prints 154, 89512 and 71319425714.

(b) A structurally independent route, sharing with (a) only the identity of Theorem 3.1: P on subsets of V_5 by the lowest-vertex recursion $\mathrm{P}(X) = \sum_{c\colon \min X\in[c]\subseteq X} \mathrm{P}(X \setminus [c])$, memoised on canonical forms under the hyperoctahedral group B_5 of order 3840, and then Corollary 3.2 at $n = 5$ evaluated as a sum over the 1 228 158 B_5-orbits of subsets of V_5, the orbits being found by an early-exit canonicality scan of all 2^{32} masks. About 1.2 CPU-hours. Its intermediate checkpoint $\mathrm{P}(V_5) = 71319425714$ is itself an independent confirmation of the sixth term.

(c) A sampled cross-check of the level-5 half of route (a) against a plain memoised lowest-vertex recursion over the subsets of the sampled set, with no group theory and nothing shared but the level-4 table: 3400 random vertex sets of V_5 of up to 22 vertices, zero mismatches.

(d) Proposition 5.1, in a different language and compiler from (a) and (b), on the certified table T_4.

This remark is a report of computation and is not part of the machine-checked development; the identity all four routes implement is, however, exactly Theorem 3.4 or Corollary 3.2. For orientation, $f(4)/f(3)^2 = 3.77$, $f(5)/f(4)^2 = 8.90$ and $f(6)/f(5)^2 = 42.5$.

Remark 5.3 (Outside the formal development: CStrees on seven binary variables). By the proposition of [4] quoted in Section 1, the number of CStrees on p binary variables is $p!\prod_{k=1}^{p} B_k^{(c)}$ with $B_k^{(c)} = f(k-1)$. At $p = 7$, Remark 5.2 gives

$$7!\prod_{k=1}^{7} B_k^{(c)} \;=\; 17\,125\,077\,428\,476\,440\,507\,901\,823\,199\,935\,254\,770\,103\,418\,880 \;\approx\; 1.71251\cdot 10^{46},$$

the cell at $p = 7$ that [4] has left unfilled in every version from January 2021 to the present — blank in v1, and printed as "?" from v2 (March 2021) onwards. This is arithmetic on Remark 5.2 through a proposition of [4] that is about staged trees rather than about cubes and is not formalised here; we state it as a report. The same formula was evaluated at $p = 1, \ldots, 6$ before the new cell was filled, reproducing the printed column of [4] exactly: 1, 4, 96, 59136, 26466908160 and 11 325 628 142 378 282 557 440, which v4 of [4] prints in the rounded form $1.1326 \cdot 10^{22}$ and v1 prints exactly.

6. Two by-products

Remark 6.1 (Outside the formal development: enumeration data for Q_4). The companion repository of [4] ships files of subcube partitions of Q_3 and Q_4 [3]. Read as comma-separated blocks of space-separated vertex labels with singleton blocks omitted — a reading forced by the 3-cube file, which under it yields 153 distinct valid records, so $154 = f(3)$ once the all-singleton partition is restored — the 4-cube file yields 61711 distinct records. Every one of them is a genuine partition of V_4 into genuine faces; the file plus the restored all-singleton partition is simply 61712 partitions against $f(4) = 89512$, so 27800 of them are absent. This is neither an erratum nor an undocumented gap. [4] makes no completeness claim about the repository, and the repository itself says of this file, in `CStrees.R`, "We also produce a subset of the partitions of the 4-dimensional cube. These partitions are those that can be recursively constructed by first partitioning the whole cube into two disjoint facets and then partitioning these two facets." We record the numbers here for one reason. That recursive construction — which is what the repository's notebook implements for the 4-cube file, extending and joining pairs of partitions of Q_{d-1} coordinate by coordinate — is the enumeration analogue of the $A = \emptyset$ term of Theorem 3.1, the term in which no cube spans the cut. What the identity adds is the sum over the other A, and a construction restricted to that one term does not reach $f(4)$; that the file stops well short of it is also consistent with [4] taking 89512 and 71319425714 from OEIS rather than from its own code.

Remark 6.2 (Outside the formal development: partitions up to symmetry). The name of A018926 says "without reducing isomorphic cases", which names a companion sequence. Enumerating every subcube partition of Q_n for $n \le 4$ and canonicalising the multiset of blocks under the hyperoctahedral group B_n of order $2^n n!$ gives

n	0	1	2	3	4
$f(n)$	1	2	8	154	89512
up to B_n	1	2	4	15	434

and $1, 2, 4, 15, 434$ is, as of 2026-08-29, not in OEIS (nor are its tails $2, 4, 15, 434$ and $4, 15, 434$). The computation is a direct enumeration plus canonicalisation, seconds of work once the partitions are listed, and it validates itself by reproducing $f(0), \ldots, f(4)$ in the same run; we claim no more for it than that nobody appears to have written the numbers down. The next term is the interesting one and is not a direct enumeration: $n = 5$ has $7.1 \cdot 10^{10}$ partitions and would need Burnside's lemma with a per-group-element count of fixed partitions.

7. Verification

Every theorem, corollary, lemma and proposition stated above has been formally verified in Lean 4, in a development built on *Mathlib* [6]; that development contains no `sorry` and no floating-point arithmetic anywhere, and the two remarks of Section 6, Remark 5.2 and Remark 5.3 are labelled in the text as lying outside it. Theorems 3.1 and 3.4, Corollary 3.2 and Lemma 3.3 hold for every n and depend on no axioms beyond `propext`, `Classical.choice` and `Quot.sound`: no step of them is decided by evaluation. The numerical statements are where computation enters, and it enters through Lean's `native_decide`, which delegates a decidable proposition to compiled code and adds one axiom per invocation; the exhaustive searches so delegated are exactly those named in the proof sketches — the tower evaluation behind $f(3)$ and $f(4)$, the four table checks over $4 + 16 + 256 + 65536$ vertex sets and the 65536-term outer sum behind $f(5)$ (231 seconds), the corresponding control evaluations, and the five range sums of Proposition 5.1 (3.17 CPU-hours in total, 1.54 GB each). Nothing else is delegated: the table-building arithmetic of Section 4 is deliberately left unverified and is forced to be correct by the check, and the range-splitting that assembles Proposition 5.1 from its five pieces is an ordinary induction. The computational core evaluated by the five range sums imports nothing, so that it can be compiled to a shared library and run as native code rather than through the interpreter; the two regimes were measured on one machine at $5.15 \cdot 10^7$ against $2.2 \cdot 10^6$ inner operations per second, a factor of 23, which for this sum is the difference between hours and days. The formal development is available in the authors' repository; repository reference to be added at submission.

8. What is not settled

A proof that $f(6) = S$. As explained after Proposition 5.1, what separates S from $f(6)$ is the level-4 bridge between vertex sets and bitmasks. It is a bounded, fully specified piece of work, and the expensive half — the evaluation — is already done.

$f(7)$. Out of reach by this route, and by a wide margin. The same two-level identity from a level-5 table needs $2 \cdot 5^{32} \approx 4.66 \cdot 10^{22}$ operations and a 2^{32}-entry table; a three-level version removes the table but not the exponent; reduction under B_6, of order 46080, removes at most that factor.

A closed form. Neither Theorem 3.1 nor Theorem 3.4 is a formula: both are recursions whose cost still grows doubly exponentially, and they say nothing about the asymptotics studied in [1]. The request of [4] for a closed form for $B_p^{(c)}$ is untouched.

The restricted counts. Problem 1.1(iv) of [1] asks for $f_{\leq 2}(d)$, the number of partitions using only subcubes of dimension at most 2, and their Problem 1.10 asks whether $f(d)/f_{\leq 2}(d)$ is subexponential; exact values at $d \leq 6$ would be data that, as far as we know, nobody has. Restricting the admissible dimensions does not simply restrict Theorem 3.1, because the dimension of a spanning cube drops by one on projection while that of a fixed cube does not, so a bound on dimensions propagates to a different bound in the first of the three factors. The dynamic programming behind route (b) of Remark 5.2, which enumerates admissible subcubes directly, does restrict. We have computed nothing in this direction. One restriction of this kind is settled in the literature, in the opposite direction: Rios, Markham and Solus (arXiv:2402.07762v2) give a closed formula for the number of partitions of Q_d into faces of dimension *at least* $d-2$. That count and $f_{\leq 2}$ do not meet.

Partitions up to symmetry. Remark 6.2 stops at $n = 4$ because the method is enumeration. The value at $n = 5$ is open.

Question 8.1. Is there a method of computing $f(n)$ that does not tabulate a function on the subsets of a vertex set? Enumeration, the lowest-vertex recursion, Theorem 3.1 and its two-level form all do; the cheapest of them is the two-level form, and it still stores $2^{2^{n-2}}$ values and performs $2 \cdot 5^{2^{n-2}}$ operations. Reduction under the symmetry group of the cube divides that by at most $2^n n!$, which against a doubly exponential cost buys one term at best.

References

[1] N. Alon, J. Balogh and V. N. Potapov, *Partitioning the hypercube into smaller hypercubes*, Illinois J. Math. **69** (2025), no. 1, 109–122; `doi:10.1215/00192082-11792788`; arXiv:2401.00299. All quotations above, and the problem numbers we cite, are from the e-print version v3 (7 November 2024), read on 2026-08-29; we have not checked whether the journal version renumbers them.

[2] F. Bridoux, M. Gadouleau and G. Theyssier, *Commutative automata networks*, in: H. Zenil (ed.), *Cellular Automata and Discrete Complex Systems* (AUTOMATA 2020), Lecture Notes in Computer Science **12286**, Springer, 2020, pp. 43–58; `doi:10.1007/978-3-030-61588-8_4`. The sentence quoted above was read in the e-print, arXiv:2004.09806v1 (21 April 2020), which is its only arXiv version; we have not seen the proceedings text.

[3] `soluslab/CStrees`, the companion code repository of [4], files `threeDpartitions.txt`, `fourDpartitions.txt` and `CStrees.R` at commit `7b54326` of 2 January 2022, which is its current head; downloaded and counted 2026-08-29, re-downloaded and re-counted 2026-08-30.

[4] E. Duarte and L. Solus, *Representation of context-specific causal models with observational and interventional data*, J. R. Stat. Soc. Ser. B Stat. Methodol. **88** (2026), no. 2, 567–610, published online 4 October 2025; `doi:10.1093/jrsssb/qkaf059`; arXiv:2101.09271. All quotations above are from the e-print version v4 (16 July 2024), read on 2026-08-29, and are version-sensitive: v1 asks for a formula rather than a "closed-form formula", and its figure of counts is differently numbered, differently captioned and prints the $p = 6$ CStree count as an exact integer. The results quoted are unnumbered here because we do not use that paper's internal numbering, which also moved between versions.

[5] S. A. Irvine, *jOEIS*: Java implementations of sequences in the OEIS, `https://github.com/archmageirvine/joeis`, program `src/irvine/oeis/a018/A018926.java`, source read 2026-08-29. The project states no citation form of its own; this follows the link A018926 itself carries, "Sean A. Irvine, Java program (github)".

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

[7] OEIS Foundation Inc., entry **A018926**, *The On-Line Encyclopedia of Integer Sequences*, `https://oeis.org/A018926`, fetched 2026-08-29 and again 2026-08-30, unchanged between the two: page version #27 of 5 January 2024; offset 0; six terms; keyword `more`; one extensions line, "a(5) from Sean A. Irvine, Feb 21 2019"; one reference, "D. E. Knuth, personal communication"; no formula and no recursion. The entry's revision history was read on 2026-08-30.