hexagon:2611.00029v1

THE INVERSE PROBLEM FOR THE NUMBER OF CONNECTED INDUCED SUBGRAPHS

LUCAS ALVES DE ALMEIDA

Abstract. For a finite graph G, let $c(G)$ be the number of nonempty vertex sets that induce a connected subgraph. We show that a positive integer k equals $c(G)$ for some finite connected graph G if and only if $k \notin \{2, 4, 5, 8, 9, 16\}$. Every attainable value $k \neq 25$ is attained on $\lfloor \log_2 k \rfloor + 1$ vertices, the least possible number, and 25 needs six vertices. More precisely, for every $m \geq 6$ each integer from 2^{m-1} to $2^m - 1$ is the value of c on a connected graph with exactly m vertices. The proofs of these results are short and use no computer; computer calculations serve only as independent checks, and for two auxiliary results: the values on six to eight vertices, and the result on all connected subgraphs described below. The value 16 falls into a gap above the path: a connected graph on $n \geq 4$ vertices other than the path P_n has at least $n - 3$ more connected sets than P_n. The values $k \geq 17$ are realized by a recursive construction from cones, disjoint unions and a family of graphs that we call near-cones, starting from seven small graphs; it produces a graph with $c(G) = k$ in time polynomial in $\log k$. The question arose from a problem of Ordoñez in continuum theory, through a reduction by Camargo, Ordoñez and Ramírez, who found the values up to 255 by a computer search. The proof is formalized in Lean 4. We also observe that a theorem of Czabarka, Székely and Wagner on subtrees reduces the corresponding problem for the number of all connected subgraphs to a finite computation, and we report its outcome. We did not find these results in the literature, but we make no claim of novelty or priority.

1. Introduction

Let G be a finite simple graph. A *connected set* of G is a nonempty set of vertices that induces a connected subgraph, and $c(G)$ denotes the number of connected sets of G. Equivalently, $c(G)$ is the number of connected induced subgraphs of G, where subgraphs induced by different vertex sets are counted separately even when they are isomorphic. For instance,

$$c(P_n) = \frac{n(n+1)}{2}, \qquad c(K_n) = 2^n - 1, \qquad c(K_{1,d}) = 2^d + d, \qquad c(C_n) = n(n-1) + 1$$

for the path, the complete graph, the star and the cycle; these and several other closed formulas are given by Vince [23, Example 1]. For a tree, $c(G)$ is the number of subtrees. Write $T(n) = n(n+1)/2$. If G is connected with n vertices, then

$$T(n) \leq c(G) \leq 2^n - 1 \tag{1}$$

(Lemma 2.1(f)). The *inverse problem* for c asks which positive integers are of the form $c(G)$. The question is only interesting for connected graphs: c is additive over components, so every positive integer k is c of the edgeless graph on k vertices.

Theorem 1.1. *A positive integer k equals $c(G)$ for some finite connected graph G if and only if*

$$k \notin \{2, 4, 5, 8, 9, 16\}.$$

The upper bound in (1) shows that a connected graph with $c(G) = k$ has at least $\lfloor \log_2 k \rfloor + 1$ vertices. This bound is attained with a single exception.

Date: November 2026.

2020 *Mathematics Subject Classification.* Primary 05C30; Secondary 05C40, 54B20, 68V20.

Key words and phrases. Connected induced subgraph, connected set, inverse problem, graph invariant, subtrees, cone, formal verification, Lean.

Theorem 1.2. *Let k be a positive integer with $k \notin \{2,4,5,8,9,16\}$. The least number of vertices of a connected graph G with $c(G) = k$ is $\lfloor \log_2 k \rfloor + 1$ if $k \neq 25$, and 6 if $k = 25$.*

Both theorems follow from a statement about graphs with a prescribed number of vertices. For $m \geq 0$ let

$$V_m = \{c(G) : G \text{ a connected graph with exactly } m \text{ vertices}\},$$
$$\Lambda_m = \{c(G) : G \text{ a graph with exactly } m \text{ vertices}\}.$$

For integers $a \leq b$ we write $[a,b] = \{a, a+1, \ldots, b\}$. By (1), $V_m \subseteq [T(m), 2^m - 1]$.

Theorem 1.3. (a) *For every $m \geq 6$, $V_m \supseteq [2^{m-1}, 2^m - 1]$: every integer k with $2^{m-1} \leq k \leq 2^m - 1$ is $c(G)$ for a connected graph G with exactly m vertices.*
(b) *For every $m \geq 7$, $\Lambda_m \supseteq [m, 2^m - 1]$.*

Part (a) fails for $m = 5$, since $25 \notin V_5$ (Proposition 3.4). A computer search shows that part (b) fails for $m = 6$, since $26 \notin \Lambda_6$ (Section 9).

Where the exceptions come from. By (1), the values of c on connected graphs with $n = 1,2,3,4,5,6$ vertices lie in the intervals

$$[1,1], \quad [3,3], \quad [6,7], \quad [10,15], \quad [15,31], \quad [21,63],$$

and from $n = 4$ on consecutive intervals overlap. The five integers $2,4,5,8,9$ lie in none of these intervals. The integer 16 lies only in the interval for $n = 5$, and there it equals $T(5) + 1$. It falls into a gap just above the path. A connected graph on $n \geq 4$ vertices other than P_n has at least $T(n) + n - 3$ connected sets (Proposition 3.2), so $T(n) + 1, \ldots, T(n) + n - 4$ are never attained on n vertices. For $n \geq 6$ these gaps are covered by graphs with fewer vertices, but for $n = 5$ the gap is the single value 16, and no other order covers it. Theorems 1.1–1.3 say that nothing else is lost.

The method. The proofs of Theorems 1.1–1.3 use no computer. Computer calculations appear in this paper only as independent checks of explicit counts (Section 9), in the census of the values on six to eight vertices (Section 6), and in Proposition 7.3 on all connected subgraphs, which is labelled as computer-assisted. The exclusions come from (1) and Proposition 3.2. For the realizations, the basic observation is that adding a universal vertex to a graph K (taking the *cone* over K) adds exactly $2^{|K|}$ connected sets. So the cones over the graphs with $m-1$ vertices realize $2^{m-1} + \Lambda_{m-1}$ on m vertices. Every graph on $m-1$ vertices has at least $m-1$ connected sets, so cones miss the $m-1$ values at the bottom of $[2^{m-1}, 2^m - 1]$. To fill this gap we use *near-cones* (Definition 4.1). A near-cone is the cone over the disjoint union of a star and a graph R, with one edge at the apex removed; removing that edge destroys exactly $2^{|R|}$ connected sets. Choosing the size of the star near $\log_2 m$ makes the missing values reachable from $\Lambda_{|R|}$ by induction (Theorem 4.6). The induction starts at seven vertices and needs seven small graphs, which we call *sporadic*. The construction is explicit and yields a graph with $c(G) = k$ on $\lfloor \log_2 k \rfloor + 1$ vertices in time polynomial in $\log k$ (Remark 5.1).

1.1. **How the problem arose.** This work started from a question in continuum theory. A *continuum* is a nonempty compact connected metric space. A subcontinuum A of a continuum X is *regular* if $A = \mathrm{cl}_X \operatorname{int}_X A$, and Ordoñez [18] introduced the hyperspace $\mathcal{D}(X)$ of regular subcontinua of X. He asked for which n there is a continuum X with exactly n regular subcontinua [18, Problem 4.14]. Camargo, Ordoñez and Ramírez [7] showed that when $\mathcal{D}(X)$ is finite, it is determined by its minimal elements, which have pairwise disjoint interiors and cover X [7, Theorem 3.16]. This reduces the computation of $|\mathcal{D}(X)|$ to a count in a graph: $|\mathcal{D}(X)|$ is the number of connected sets of the intersection graph of these minimal elements, a finite connected graph [7], [2, Proposition 3.2]. They proved that $|\mathcal{D}(X)| \notin \{2,4,5,8,9\}$ [7, Proposition 3.18]. A computer search over all graphs with at most eight vertices showed that 16 does not occur as a value of c, while every integer from

17 to 255 does [7, Remark 3.20]. They asked whether every $k \geq 256$ is the cardinality of some $\mathcal{D}(X)$ [7, Question 3.21].

The author was working on this question. In a companion paper [2] we answer it: the finite cardinalities of $\mathcal{D}(X)$ are exactly the positive integers outside $\{2, 4, 5, 8, 9, 16\}$. That paper realizes every finite connected graph by copies of the Knaster continuum glued at points. Hence the topological question is equivalent to the graph-theoretic one, which is Theorem 1.1. The graph-theoretic result does not depend on continuum theory and may be of independent interest. So we present it here on its own, with complete proofs and a more detailed explanation of the construction. Compared with [2], this paper also contains a new and more conceptual treatment of five vertices (Propositions 3.2 and 3.4) and some related observations (Sections 6 and 7).

1.2. **Related work and the status of the result.** The invariant $c(G)$ has been studied mainly from the extremal point of view. For trees, c is the number of subtrees, studied by Székely and Wang [21], among others. For general graphs, Dossou-Olory [10, 11, 12], Andriantiana and Dossou-Olory [5] (Nordhaus–Gaddum inequalities), and Haslegrave [14] studied the extremal values of c and of related counts of connected sets. Vince [23] studied the growth of c, and of the number of partitions of the vertex set into connected sets, along infinite families of graphs; he computed c for several families and bounded it in terms of the number of edges. The invariant is a specialization of the subgraph component polynomial of Tittmann, Averbouch and Makowsky [22], which counts vertex sets by their size and by the number of components of the induced subgraph; $c(G)$ counts the vertex sets inducing exactly one component. Pandey and Patra [19] studied extremal problems for the number $F(G)$ of all connected subgraphs, not necessarily induced.

Inverse problems ask which values a graph invariant takes. Linek [15] showed that every positive integer is the number of independent sets of a bipartite graph. Wang and Yu [24] showed that all but 49 positive integers are Wiener indices of trees. Czabarka, Székely and Wagner [8] solved the inverse problem for several tree parameters; in particular, every positive integer except 34 explicit values, the largest being 89, is the number of subtrees of some tree. More recently, Alizadeh et al. [1] showed that every nonnegative integer except 1 is the Mostar index of a graph, and Bernert and Shaw [6] showed that, for each fixed $k \geq 2$, all sufficiently large integers are Steiner–Wiener k-indices of graphs.

Status. We searched the literature for a classification of the values of c on general connected graphs. The search used web and arXiv searches on connected sets, connected induced subgraphs, the subgraph component polynomial and inverse problems for graph invariants, and the references of the works above. We did not find Theorems 1.1–1.3. The only previous determination of these values that we know is the computational one of [7, Remark 3.20] for graphs with at most eight vertices. We also note that Theorem 1.1 can be deduced from the theorem of [8] on subtrees by a finite computation (Remark 7.1); that route uses a computer and gives neither Theorem 1.2 nor Theorem 1.3. The question is natural and the proof is elementary, so the result may be known, or folklore, in a form that we did not find. We therefore make no claim of novelty or priority, and we would be grateful for references.

1.3. **Formalization, computations and organization.** Theorems 1.1–1.3, the other results of Sections 2–5 and 6, and Lemma 7.2 are formalized in Lean 4 with Mathlib (Section 8). All explicit finite counts in this paper were also checked by computer (Section 9). The LaTeX source, the Lean formalization, the programs and their outputs are in the repository [3]. Section 2 collects the counting rules. Section 3 treats graphs with at most five vertices and proves the gap above the path. Section 4 contains the construction, and Section 5 the proofs of the main theorems. Section 6 discusses the values on a fixed number of vertices. Section 7 discusses subtrees, all connected subgraphs, and the topological interpretation.

2. Counting rules

Graphs are finite and simple; they need not be connected, and the graph with no vertices is allowed, with $c = 0$. For a graph G we write $|G|$ for its number of vertices, and $G[F]$ for the subgraph induced by $F \subseteq V(G)$. We write $c_v(G)$ for the number of connected sets containing a vertex v. For sets A, B of integers and an integer t we write $t + A = \{t + a : a \in A\}$ and $A + B = \{a + b : a \in A,\ b \in B\}$.

A vertex is *universal* if it is adjacent to every other vertex. The *cone* over a graph H is obtained by adding one new vertex adjacent to every vertex of H. A *pendant vertex* is a vertex of degree one. A *spider* consists of a vertex, its *centre*, and paths, its *legs*, each joined to the centre by an edge at one of its ends and otherwise disjoint from each other. The *length* of a leg is its number of vertices, not counting the centre. The *paw* is a triangle with one pendant vertex, and the *diamond* is K_4 minus an edge.

Lemma 2.1. *Let G, H and K be graphs.*

(a) (Disjoint union.) *If G is the disjoint union of H and K, then $c(G) = c(H) + c(K)$.*

(b) (Monotonicity.) *If H is a spanning subgraph of G, then $c(H) \le c(G)$. If moreover some edge of G is not an edge of H, then $c(H) < c(G)$.*

(c) (Gluing at a vertex.) *Suppose that $G = G_1 \cup G_2$ for subgraphs with $V(G_1) \cap V(G_2) = \{v\}$ and $E(G) = E(G_1) \cup E(G_2)$. Then*

$$c(G) = c_v(G_1)\, c_v(G_2) + c(G_1 - v) + c(G_2 - v).$$

(d) (Cone.) *If G is the cone over K, then $c(G) = 2^{|K|} + c(K)$.*

(e) (Almost-cone.) *Let u be a vertex of K of degree d in K. If G is obtained from K by adding a vertex w adjacent to every vertex of K except u, then*

$$c(G) = c(K) + 2^{|K|} - 2^{|K|-1-d}.$$

(f) (Range.) *If G is connected with $n \ge 1$ vertices, then $T(n) \le c(G) \le 2^n - 1$. The lower bound is attained by the path P_n and the upper bound by K_n.*

Proof. (a) A connected set lies in one component.

(b) Every connected set of H is connected in G (compare [23, Lemma 1]). If uv is an edge of G but not of H, then $\{u, v\}$ is connected in G but not in H.

(c) No edge joins $G_1 - v$ and $G_2 - v$, so a connected set avoiding v lies in $G_1 - v$ or in $G_2 - v$. Let F be a connected set containing v, and put $F_i = F \cap V(G_i)$. A path in $G[F]$ between two vertices of F_i stays in $V(G_i)$: it could leave $V(G_i)$ only through v, and it would have to return through v, which a path cannot do. So each F_i is a connected set of G_i containing v. Conversely, the union of a connected set of G_1 and one of G_2, both containing v, is connected. This gives a bijection between the connected sets of G containing v and such pairs (F_1, F_2).

(d) and (e) Let w be the added vertex. The connected sets of G avoiding w are exactly those of K. The sets containing w are the sets $\{w\} \cup U$ with $U \subseteq V(K)$, and distinct sets U give distinct sets $\{w\} \cup U$. Hence $c(G) = c(K) + c_w(G)$, where $c_w(G)$ is the number of sets $U \subseteq V(K)$ such that $\{w\} \cup U$ is connected. This happens if and only if every component of $K[U]$ contains a neighbour of w: such components are joined to w, and a component of $K[U]$ without a neighbour of w would be a component of $G[\{w\} \cup U]$ not containing w. In (d) the condition holds for all $2^{|K|}$ sets U. In (e) the neighbours of w are all vertices of K except u, so the condition fails exactly when $\{u\}$ is a component of $K[U]$, that is, when $u \in U$ and U contains no neighbour of u. Such a set U consists of u and an arbitrary subset of the $|K| - 1 - d$ vertices that are neither u nor neighbours of u, so there are $2^{|K|-1-d}$ of them.

(f) Let S be a spanning tree of G. For vertices x, y, possibly equal, let $P_{x,y}$ be the x–y path in S. Its vertex set is connected in S, hence in G. Since S is a tree, $S[V(P_{x,y})] = P_{x,y}$: an edge of S joining two vertices of $P_{x,y}$ that are not consecutive on $P_{x,y}$ would close a cycle with a segment

of $P_{x,y}$. If $x \neq y$, then x and y are the only vertices of degree one in $P_{x,y}$, and if $x = y$, then $V(P_{x,y}) = \{x\}$. So the set $V(P_{x,y})$ determines the unordered pair $\{x, y\}$, and distinct pairs give distinct sets. This gives $n + \binom{n}{2} = T(n)$ connected sets. The connected sets of P_n are its $T(n)$ intervals, since a set of vertices of P_n that is not an interval misses a vertex lying between two of its vertices, and is then disconnected. There are only $2^n - 1$ nonempty sets, and all of them are connected in K_n. For trees, the lower bound is also part of [23, Theorem 4], where c is bounded in terms of the number of edges. □

Example 2.2. (i) A spider whose legs have lengths $\ell_1, \ldots, \ell_s$ satisfies

$$c = \prod_i (\ell_i + 1) + \sum_i T(\ell_i) :$$

a connected set containing the centre takes an initial segment of each leg, possibly empty, and a connected set avoiding the centre is an interval of one leg. For the star $K_{1,d}$ this gives $2^d + d$.

(ii) Attach $t_v \geq 0$ pendant vertices at each vertex v of a complete graph K_a. A connected set is either a single pendant vertex, or meets the clique in a nonempty set Q, together with any pendant vertices attached at Q. Hence $c = \prod_v (1 + 2^{t_v}) - 1 + \sum_v t_v$. For the paw this gives $3 \cdot 2 \cdot 2 - 1 + 1 = 12$.

(iii) Let Φ be a disjoint union of stars $K_{1,s_1}, \ldots, K_{1,s_p}$ with $s_i \geq 1$ and at most n vertices in all, and let $K_n - \Phi$ be the complete graph on n vertices with the edges of a copy of Φ removed. A set U is disconnected in $K_n - \Phi$ if and only if $U = A \cup B$ with A, B nonempty and disjoint and all pairs in $A \times B$ edges of Φ. This complete bipartite subgraph of Φ is connected, so it lies in one star, and then A or B is the centre alone. Hence

$$c(K_n - \Phi) = 2^n - 1 - \sum_i (2^{s_i} - 1).$$

For the diamond $K_4 - K_{1,1}$ this gives 14.

(iv) The connected sets of the cycle C_n, $n \geq 3$, are the whole vertex set and the $n(n-1)$ proper cyclic intervals, so $c(C_n) = n(n-1) + 1$.

3. At most five vertices, and the gap above the path

Lemma 3.1. (a) *$V_1 = \{1\}$, $V_2 = \{3\}$, $V_3 = \{6, 7\}$ and $V_4 = [10, 15]$.*
(b) *$\Lambda_2 = \{2, 3\}$, $\Lambda_3 = \{3, 4, 6, 7\}$ and $\Lambda_4 = [4, 8] \cup [10, 15]$.*

Proof. (a) The connected graphs on at most three vertices are K_1, K_2, P_3 and K_3. The six connected graphs on four vertices are the path, the star, the paw, the cycle, the diamond and K_4, with $c = 10, 11, 12, 13, 14, 15$ by Lemma 2.1 and Example 2.2.

(b) By Lemma 2.1(a), the value of c on a disconnected graph is the sum of its values on the components. A disconnected graph on four vertices has components with $1+3$, $2+2$, $1+1+2$ or $1+1+1+1$ vertices, giving $\{7, 8\}$, $\{6\}$, $\{5\}$ and $\{4\}$. Together with V_4 this gives Λ_4. The sets Λ_2 and Λ_3 are found in the same way. □

Proposition 3.2 (The gap above the path)**.** *Let $n \geq 4$ and let G be a connected graph with n vertices other than the path P_n. Then*

$$c(G) \geq T(n) + n - 3,$$

with equality for the spider with legs of lengths 1, 1 and $n - 3$.

Proof. Case 1: G has a spanning tree S that is not a path. By Lemma 2.1(b), $c(G) \geq c(S)$, so it suffices to find $T(n) + n - 3$ connected sets of S.

Path sets. By the proof of Lemma 2.1(f), the vertex sets $V(P_{x,y})$ of the paths of S are $T(n)$ distinct connected sets, and each of them induces a path in S, so every vertex of $S[V(P_{x,y})]$ has degree at most 2 there.

Branching sets. A tree whose vertices all have degree at most 2 is a path, so S has a vertex v of degree at least 3. Choose three distinct neighbours a, b, q of v and put $Q = \{v, a, b, q\}$. For each of the $n-4$ vertices $y \notin Q$, put $Q_y = Q \cup V(P_{v,y})$. Let H_y be the union of the star with centre v and leaves a, b, q and the path $P_{v,y}$. It is a connected subgraph of S with vertex set Q_y, so Q_y is a connected set, and so is Q. Moreover $S[Q_y] = H_y$, because $S[Q_y]$ is a tree and a tree has no connected spanning subgraph other than itself. In $S[Q]$ and in $S[Q_y]$ the vertex v has degree at least 3, so none of these sets is a path set.

Distinctness. The set Q differs from every Q_y, since $y \in Q_y \setminus Q$. In H_y the vertices outside Q are the vertices of $P_{v,y}$ outside Q. The edges of the star meet only Q, so such a vertex has degree 2 in H_y if it is an inner vertex of $P_{v,y}$, and degree 1 if it is the end y. Hence y is the only vertex of degree one of $S[Q_y]$ outside Q, and Q_y determines y. So Q and the sets Q_y are $n-3$ distinct connected sets of S, none of them a path set, and $c(S) \geq T(n) + n - 3$.

Case 2: every spanning tree of G is a path. Fix one, with vertices $v_1, \ldots, v_n$ in this order, and call it P_n. Since G is not a path, it has an edge $v_a v_b$ that is not an edge of P_n, with $1 \leq a$ and $a + 2 \leq b \leq n$. By Lemma 2.1(b), $c(G) \geq c(P_n + v_a v_b)$. The connected sets of P_n are its $T(n)$ intervals, and they remain connected in $P_n + v_a v_b$, so it suffices to count the connected sets of $P_n + v_a v_b$ that are not intervals.

Let U be such a set. As U is not an interval, it is the union of at least two maximal intervals contained in U, and no edge of P_n joins two of them. The single extra edge $v_a v_b$ can join only two of them, so U is the union of exactly two maximal intervals, one containing v_a and the other containing v_b. Thus

$$U = \{v_\alpha, \ldots, v_x\} \cup \{v_y, \ldots, v_\beta\} \quad \text{with} \quad 1 \leq \alpha \leq a \leq x, \quad x + 2 \leq y \leq b \leq \beta \leq n.$$

Conversely, each such set is connected in $P_n + v_a v_b$ and is not an interval. The quadruple (α, x, y, β) is determined by U: the indices α and β are the least and the greatest index in U, the index $x+1$ is the least index above α missing from U, and y is the least index above x in U. So the sets U correspond bijectively to these quadruples. Put $t = b - a - 1$; the conditions $1 \leq a$ and $a + 2 \leq b \leq n$ give $1 \leq t \leq n - 2$. There are a choices of α and $n + 1 - b$ choices of β. For $x = a, \ldots, b - 2$ there are $b - x - 1$ choices of y, so there are $\sum_{s=1}^{t} s = T(t)$ choices of (x, y). Hence the chord adds exactly $a(n+1-b)T(t)$ connected sets. The factors satisfy $a \geq 1$, $n + 1 - b \geq 1$ and $a + (n + 1 - b) = n - t$. Since $pq \geq p + q - 1$ for positive integers p, q, and $T(t) \geq t$,

$$a(n + 1 - b)\, T(t) \;\geq\; (n - t - 1)\, t \;\geq\; n - 2 \qquad (1 \leq t \leq n - 2),$$

where the last inequality holds because $(n - t - 1)t$ is a concave function of t that equals $n - 2$ at $t = 1$ and at $t = n - 2$. Hence $c(G) \geq c(P_n + v_a v_b) \geq T(n) + n - 2$.

Finally, by Example 2.2(i), the spider with legs 1, 1, $n - 3$ has $c = 2 \cdot 2 \cdot (n - 2) + T(1) + T(1) + T(n - 3) = \frac{n^2 + 3n - 6}{2} = T(n) + n - 3$. □

Section 8 sketches a second proof of Proposition 3.2, by induction on connected sets, which is the one formalized in Lean.

Corollary 3.3. *For $n \geq 5$, none of the $n - 4$ integers $T(n) + 1, \ldots, T(n) + n - 4$ lies in V_n. In particular $16 = T(5) + 1 \notin V_5$, and 16 is not a value of c on any connected graph.*

Proof. The first assertion is Proposition 3.2, since $c(P_n) = T(n)$. A connected graph with $c = 16$ has n vertices with $T(n) \leq 16 \leq 2^n - 1$, by Lemma 2.1(f). This forces $n = 5$. □

Proposition 3.4. $V_5 = \{15\} \cup [17, 24] \cup [26, 31]$.

Proof. The inclusion $\supseteq$ holds because $c(P_5) = 15$ and $V_5 \supseteq [17, 24] \cup [26, 31]$ by Lemma 4.4(b) below, whose proof does not use this proposition. For $\subseteq$, let G be connected with five vertices. By Lemma 2.1(f), $c(G) \in [15, 31]$, and $c(G) \neq 16$ by Corollary 3.3. It remains to show that $c(G) \neq 25$. We distinguish cases by the maximum degree Δ of G; the table at the end of the proof summarizes them.

If $\Delta = 4$, then G is the cone over a graph H with four vertices, and $c(G) = 16 + c(H)$ by Lemma 2.1(d). Since $9 \notin \Lambda_4$ (Lemma 3.1), $c(G) \neq 25$.

If $\Delta = 3$, let w be a vertex of degree 3, let u be the vertex not adjacent to w, and let $K = G - w$. The degree d of u in K is at least 1, since G is connected, and at most 3. By Lemma 2.1(e) with $|K| = 4$, $c(G) = c(K) + 16 - 2^{3-d}$. If $c(G) = 25$, then $c(K) = 9 + 2^{3-d}$, that is, $c(K) = 13$, 11 or 10 for $d = 1$, 2 or 3. A disconnected graph on four vertices has $c \leq 8$ (Lemma 3.1(b)), so K is connected. The six connected graphs on four vertices have the distinct values $10, \ldots, 15$ (proof of Lemma 3.1), so K is the cycle C_4, the star $K_{1,3}$ or the path P_4, respectively. But u has degree d in K, and C_4 has no vertex of degree 1, $K_{1,3}$ has no vertex of degree 2, and P_4 has no vertex of degree 3. So $c(G) \neq 25$.

If $\Delta \leq 2$, then G is P_5 or C_5, with $c = 15$ or 21. In summary, where for $\Delta = 3$ the vertex w has degree 3, $K = G - w$, and d is the degree in K of the vertex u not adjacent to w:

case	$c(G)$	$c(G) = 25$ forces	impossible because
$\Delta = 4$	$16 + c(H)$, $\lvert H\rvert = 4$	$c(H) = 9$	$9 \notin \Lambda_4$
$\Delta = 3$, $d = 1$	$c(K) + 12$	$c(K) = 13$, so $K = C_4$	C_4 has no vertex of degree 1
$\Delta = 3$, $d = 2$	$c(K) + 14$	$c(K) = 11$, so $K = K_{1,3}$	$K_{1,3}$ has no vertex of degree 2
$\Delta = 3$, $d = 3$	$c(K) + 15$	$c(K) = 10$, so $K = P_4$	P_4 has no vertex of degree 3
$\Delta \leq 2$	15 or 21	—	G is P_5 or C_5

□

The case analysis also gives a second proof that $16 \notin V_5$: a cone has $c \geq 16 + 4$, and in the case $\Delta = 3$, $c(G) = 16$ would force $c(K) = 2^{3-d} \leq 4$, hence K edgeless, contradicting $d \geq 1$.

Corollary 3.5. *The values of c on connected graphs that are at most* 16 *are* 1, 3, 6, 7 *and the integers from* 10 *to* 15*.*

Proof. Graphs with at least six vertices have $c \geq 21$, and graphs with five vertices have $c \in \{15\} \cup [17, 31]$, by Lemma 2.1(f) and Corollary 3.3. Apply Lemma 3.1. □

4. Near-cones and the dyadic intervals

The construction works with all graphs, not only connected ones, because of the cone. If H has $m - 1$ vertices and $c(H) = x$, then the cone over H is a connected graph with m vertices and $c = 2^{m-1} + x$. So the natural auxiliary object is the set Λ_{m-1} of values of c on all graphs with $m - 1$ vertices. Cones alone do not suffice. Every graph on $m - 1$ vertices has at least $m - 1$ connected sets, its singletons, so cones give only values $2^{m-1} + x$ with $x \geq m - 1$. They miss the $m - 1$ values from 2^{m-1} to $2^{m-1} + m - 2$, at the bottom of the interval $[2^{m-1}, 2^m - 1]$. Near-cones fill this gap.

Throughout this section we keep the two families of sets apart: V_m consists of values on *connected* graphs with m vertices, the sets we are after, and $\Lambda_m \supseteq V_m$ of values on *all* graphs with m vertices, the auxiliary sets. The rules (C) and (N) below produce elements of V_m from elements of Λ_{m-1} and Λ_r, and the rule (U) produces elements of Λ_m.

4.1. **Near-cones.**

Definition 4.1. Let $d \geq 1$, $r \geq 0$, and let R be a graph with r vertices. The *near-cone* $N(d, R)$ is obtained as follows. Take a star $K_{1,d}$ with centre u and leaves $\ell_1, \ldots, \ell_d$ and a disjoint copy of R,

and add a vertex w adjacent to every vertex except u (Figure 1). Thus $N(d, R)$ is the cone with apex w over $K_{1,d} \sqcup R$, with the edge wu removed. It is connected and has $m = d + r + 2$ vertices.

For example, $N(1, K_1)$ is the path P_4, and $N(2, K_0)$, where K_0 is the graph with no vertices, is the cycle C_4.

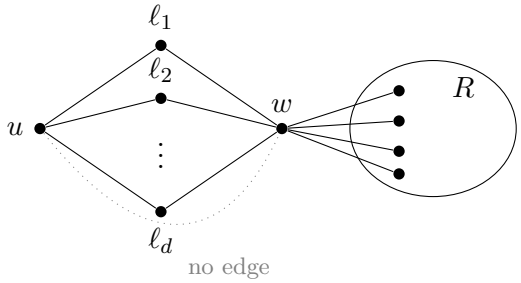


FIGURE 1. The near-cone $N(d, R)$. The vertex w is joined to $\ell_1, \ldots, \ell_d$ and to every vertex of R; the centre u of the star is joined only to the leaves ℓ_t.

Lemma 4.2. *Let $m \geq 1$.*

(U) *If $x \in V_a$ and $y \in \Lambda_{m-a}$ with $1 \leq a \leq m$, then $x + y \in \Lambda_m$. In particular $1 + \Lambda_{m-1} \subseteq \Lambda_m$.*
(C) *$2^{m-1} + \Lambda_{m-1} \subseteq V_m$.*
(N) *If $d \geq 1$, $r \geq 0$ and $d + r + 2 = m$, then $2^{m-1} - 2^r + 2^d + d + \Lambda_r \subseteq V_m$. More precisely,*

$$c(N(d, R)) = 2^{m-1} - 2^r + 2^d + d + c(R).$$

Proof. (U) is Lemma 2.1(a); for the second statement add an isolated vertex. (C) is Lemma 2.1(d).

For (N), let G' be the cone with apex w over $K_{1,d} \sqcup R$. By Lemma 2.1(a),(d) and Example 2.2(i), $c(G') = 2^{m-1} + 2^d + d + c(R)$. Now remove the edge wu to obtain $G = N(d, R)$. Sets avoiding w or u are connected in G exactly when they are connected in G'. A set $\{w\} \cup U$ with $u \in U$ is connected in G'. It is disconnected in G exactly when U contains no leaf ℓ_t, since u is adjacent only to the leaves and every other vertex of U is adjacent to w. There are 2^r such sets U: they contain u, no leaf, and an arbitrary set of vertices of R. Hence $c(G) = c(G') - 2^r$. This is also the case $K = K_{1,d} \sqcup R$ of Lemma 2.1(e). □

4.2. **Seven sporadic graphs.**

Lemma 4.3. *The seven graphs of Table 1 and Figure 2 have the stated values. Hence $19 \in V_5$; $25, 32, 35, 57 \in V_6$; and $67, 90 \in V_7$.*

Proof. The values 19, 25, 32 and 57 follow from Example 2.2; for 57, Φ consists of two copies of $K_{1,2}$.

For $c = 35$: a connected set is either one of the two pendant vertices, or a connected set Q of the diamond $K_4 - ab$ together with any of the pendant vertices attached at Q. The connected sets of $K_4 - ab$ are all nonempty subsets of its vertex set other than $\{a, b\}$. Summing $2^{|Q \cap \{a,z\}|}$ over all nonempty Q gives $3 \cdot 3 \cdot 2 \cdot 2 - 1 = 35$. Removing $Q = \{a, b\}$ subtracts 2, and the pendant vertices add 2, giving 35.

For $c = 67$: glue $K_{2,3}$ and the star $K_{1,2}$ at x (Lemma 2.1(c)). Let $\{p, q\}$ be the smaller side and $\{x, y, z\}$ the larger one. The connected sets of $K_{2,3}$ containing x are $\{x\}$ and the sets $\{x\} \cup S \cup S'$ with $\emptyset \neq S \subseteq \{p, q\}$ and $S' \subseteq \{y, z\}$; there are $1 + 3 \cdot 4 = 13$. Also $K_{2,3} - x = C_4$ has $c = 13$, and the star contributes $c_x = 4$ and $c(K_{1,2} - x) = 2$.

For $c = 90$: let $H = K_6 - va - vb$ and glue H with an edge at v. In H, the sets containing v that are disconnected are $\{v, a\}$, $\{v, b\}$ and $\{v, a, b\}$, so $c_v(H) = 32 - 3 = 29$; and $H - v = K_5$ has $c = 31$. The edge contributes $c_v = 2$ and 1. □

c	m	graph	count
19	5	triangle xyz with one pendant vertex at x and one at y	Ex. 2.2(ii): $3 \cdot 3 \cdot 2 - 1 + 2$
32	6	triangle xyz with two pendant vertices at x and one at y	Ex. 2.2(ii): $5 \cdot 3 \cdot 2 - 1 + 3$
35	6	diamond $K_4 - ab$, with one pendant vertex at a and one at a vertex z of degree 3	see the proof
57	6	K_6 minus two disjoint paths with three vertices each	Ex. 2.2(iii): $63 - 3 - 3$
67	7	$K_{2,3}$ with two pendant vertices at a vertex x of the larger side	Lem. 2.1(c): $13 \cdot 4 + 13 + 2$
90	7	K_6 minus two edges va, vb, with a pendant vertex at v	Lem. 2.1(c): $29 \cdot 2 + 31 + 1$
25	6	spider with legs of lengths 1, 2, 2	Ex. 2.2(i): $2 \cdot 3 \cdot 3 + 1 + 3 + 3$

TABLE 1. The sporadic graphs, drawn in Figure 2. The six graphs above the line fill the gaps left by the rules (U), (C) and (N) in Lemma 4.4 (Remark 4.5). The spider below the line realizes 25 on six vertices.

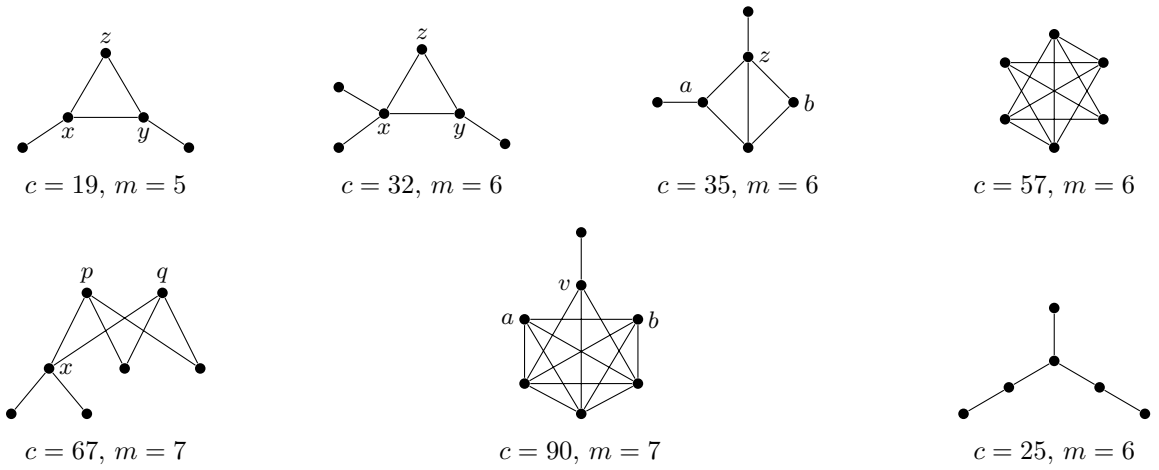


FIGURE 2. The seven sporadic graphs of Table 1, with their values of c and their numbers m of vertices. In the graph with $c = 57$, the hexagon is drawn without four of its sides; these form the two removed paths with three vertices each.

4.3. **The base.**

Lemma 4.4. (a) $\Lambda_5 \supseteq [5, 24] \cup [26, 31]$, $\Lambda_6 \supseteq [6, 25] \cup [27, 63]$ *and* $\Lambda_7 \supseteq [7, 127]$.
(b) $V_5 \supseteq [17, 24] \cup [26, 31]$, $V_6 \supseteq [32, 63]$ *and* $V_7 \supseteq [64, 127]$.

Proof. We apply the rules of Lemma 4.2 one order at a time, starting from Lemma 3.1 and $\Lambda_0 = \{0\}$. At each order we use only the inclusions already established at smaller orders, and we record only inclusions. We give the order $m = 5$ in full; Table 2 lists every order, including $m \le 4$, where the rules reproduce Lemma 3.1.

For $m = 5$, the rule (C) applied to $\Lambda_4 = [4, 8] \cup [10, 15]$ gives $16 + \Lambda_4 = [20, 24] \cup [26, 31]$. The rule (N) with $(d, r) = (1, 2)$ gives $2^4 - 2^2 + 2^1 + 1 + \Lambda_2 = 15 + \{2, 3\} = \{17, 18\}$. The sporadic graph with $c = 19$ (Lemma 4.3) supplies the remaining value, so $V_5 \supseteq [17, 24] \cup [26, 31]$. By (U), $1 + \Lambda_4 = [5, 9] \cup [11, 16]$ and $7 + 3 = 10 \in V_3 + \Lambda_2$ lie in Λ_5. Together with V_5, this gives $\Lambda_5 \supseteq [5, 24] \cup [26, 31]$. □

m	cones (C)	near-cones (N), with (d, r)	sporadic	$V_m \supseteq$	$\Lambda_m \supseteq$ by (U) and V_m
2	$2 + \{1\} = \{3\}$	—	—	$\{3\}$	$1 + \{1\}$; so $\{2, 3\}$
3	$4 + \{2, 3\} = \{6, 7\}$	—	—	$\{6, 7\}$	$1 + \{2, 3\}$; so $\{3, 4, 6, 7\}$
4	$8 + \{3, 4, 6, 7\} = \{11, 12, 14, 15\}$	$(1, 1)$: 10; $(2, 0)$: 13	—	$[10, 15]$	$1 + \{3, 4, 6, 7\}$, $3 + 3$; so $[4, 8] \cup [10, 15]$
5	$16 + \Lambda_4$: $[20, 24] \cup [26, 31]$	$(1, 2)$: $15 + \{2, 3\} = \{17, 18\}$	19	$[17, 24] \cup [26, 31]$	$[5, 9] \cup [11, 16]$, $7 + 3$; so $[5, 24] \cup [26, 31]$
6	$32 + \Lambda_5$: $[37, 56] \cup [58, 63]$	$(1, 3)$: $27 + \{3, 4, 6, 7\} = \{30, 31, 33, 34\}$; $(2, 2)$: $34 + \{2, 3\} = \{36, 37\}$	$32, 35, 57$	$[32, 63]$	$[6, 25] \cup [27, 32]$; so $[6, 25] \cup [27, 63]$
7	$64 + \Lambda_6$: $[70, 89] \cup [91, 127]$	$(1, 4)$: $51 + [13, 15] = [64, 66]$; $(2, 3)$: $62 + \{6, 7\} = \{68, 69\}$	$67, 90$	$[64, 127]$	$[7, 26] \cup [28, 64]$, $24 + 3$; so $[7, 127]$

TABLE 2. The steps of the proof of Lemma 4.4. In the cone and near-cone columns, Λ_r stands for the part of Λ_r established in the previous rows. The last column lists $1 + \Lambda_{m-1}$ and the sums $x + y$ with $x \in V_a$, $y \in \Lambda_{m-a}$ that are needed, for instance $3 + 3 \in V_2 + \Lambda_2$ and $24 + 3 \in V_5 + \Lambda_2$.

Remark 4.5. Two statements must be distinguished here: a value may fail to be produced by the rules (C) and (N), and a value may fail to be realized by any graph. Lemma 4.4 is sharp in the second sense. Indeed $25 \notin \Lambda_5$, because $25 \notin V_5$ (Proposition 3.4) and a disconnected graph with five vertices has $c \leq 15 + 1$. Also $26 \notin \Lambda_6$, by the computation of Section 9. In the first sense, even with every sporadic graph of smaller order available, the rules (C) and (N) produce on five vertices every value in $[17, 31]$ except 19 and 25, on six vertices every value in $[32, 63]$ except 32, 35 and 57, and on seven vertices every value in $[64, 127]$ except 67 and 90 (Section 9). All these values except 25 do lie in V_m, and the six sporadic graphs above the line in Table 1 realize them.

4.4. **The induction.**

Theorem 4.6. *For every $m \geq 8$:*

(G_m) $V_m \supseteq [2^{m-1}, 2^m - 1]$*;*
(H_m) $\Lambda_m \supseteq [m, 2^m - 1]$.

The two assertions play different roles. The assertion (H_m) provides auxiliary graphs, possibly disconnected, with every count from m to $2^m - 1$. The assertion (G_m) provides connected graphs for the upper half of the range. We prove (G_m) from earlier (H)-statements, and then obtain (H_m) by combining (G_m) with the isolated-vertex rule (U). In the proof of (G_m) the interval splits as

$$[2^{m-1}, 2^m - 1] = \underbrace{[2^{m-1},\, 2^{m-1} + m - 2]}_{\text{near-cones}} \cup \underbrace{[2^{m-1} + m - 1,\, 2^m - 1]}_{\text{cones}}.$$

Proof. We use induction on $m \geq 8$. The induction hypothesis is ($\mathrm{H}_{m'}$) for every m' with $7 \leq m' < m$; for $m' = 7$ this is Lemma 4.4(a). The statements ($\mathrm{G}_{m'}$) are not used.

Proof of (G_m). By (C) and (H_{m-1}), $V_m \supseteq 2^{m-1} + [m - 1, 2^{m-1} - 1] = [2^{m-1} + m - 1, 2^m - 1]$. It remains to realize $2^{m-1} + j$ for $0 \leq j \leq m - 2$. By (N), a pair (d, r) realizes $2^{m-1} + j$ if it satisfies the following conditions:

(P1) $d \geq 1$, $r \geq 0$ and $d + r + 2 = m$;
(P2) $x_j \in \Lambda_r$, where

$$x_j := 2^r - 2^d - d + j. \tag{2}$$

Indeed, if R is a graph with r vertices and $c(R) = x_j$, then $c(N(d,R)) = 2^{m-1} - 2^r + 2^d + d + x_j = 2^{m-1} + j$. When $r \geq 7$ and $r < m$, the hypothesis (H_r) gives $[r, 2^r - 1] \subseteq \Lambda_r$. Then (P2) holds for every $j \in [0, m-2]$ as soon as it holds at the two endpoints, that is, as soon as

(P3) $x_{m-2} \leq 2^r - 1$, which is equivalent to $2^d + d \geq m - 1$, and
(P4) $x_0 \geq r$, which is $2^r - 2^d - d \geq r$.

Condition (P3) asks for d large and (P4) for d small compared with r. For $m \geq 12$ we take the least d satisfying (P3). This keeps d close to $\log_2 m$, so that $r = m - 2 - d$ is large. For $8 \leq m \leq 11$ this choice leaves $r \leq 6$, where (H_r) is not available, and we check (P2) directly from Lemmas 3.1(b) and 4.4(a), with the parameters of Table 3.

m	(d, r)	j	$x_j = 2^r - 2^d - d + j$	$x_j \in \Lambda_r$ by	
8	$(2,4)$	$0, \ldots, 5$	$10, \ldots, 15$	$\Lambda_4 \supseteq [10, 15]$	Lemma 3.1(b)
8	$(3,3)$	6	3	$3 \in \Lambda_3$	Lemma 3.1(b)
9	$(2,5)$	$0, \ldots, 5$	$26, \ldots, 31$	$\Lambda_5 \supseteq [26, 31]$	Lemma 4.4(a)
9	$(3,4)$	$6, 7$	$11, 12$	$\Lambda_4 \supseteq [10, 15]$	Lemma 3.1(b)
10	$(3,5)$	$0, \ldots, 8$, $j \neq 4$	$21, \ldots, 29$, $\neq 25$	$\Lambda_5 \supseteq [5, 24] \cup [26, 31]$	Lemma 4.4(a)
10	$(2,6)$	4	62	$\Lambda_6 \supseteq [27, 63]$	Lemma 4.4(a)
11	$(3,6)$	$0, \ldots, 9$	$53, \ldots, 62$	$\Lambda_6 \supseteq [27, 63]$	Lemma 4.4(a)

TABLE 3. The parameters of the near-cones in the proof of Theorem 4.6 for $8 \leq m \leq 11$, covering $j = 0, \ldots, m-2$. For $m = 10$ the choice $(3,5)$ fails only at $j = 4$, since $25 \notin \Lambda_5$. The graphs R with $c(R) = x_j$ are those given by the proofs of Lemmas 3.1 and 4.4.

Now let $m \geq 12$, let d be the least integer $d \geq 1$ with $2^d + d \geq m - 1$, and put $r = m - 2 - d$, so that (P1) and (P3) hold. Then $d \geq 3$, since $2^2 + 2 < 11$. If $d = 3$, then $m = 12$ and $r = 7$. If $d \geq 4$, minimality of d gives $2^{d-1} + d - 1 < m - 1$, so $r = m - 2 - d \geq 2^{d-1} - 1 \geq 7$ and $r \geq d + 1$. In both cases $7 \leq r \leq m - 3$ and $r \geq d + 1$. For (P4),

$$x_0 = 2^r - 2^d - d \geq 2^r - 2^{r-1} - (r-1) = 2^{r-1} - r + 1 \geq r,$$

because $d \leq r - 1$ and $2^{r-1} \geq 2r - 1$ for $r \geq 4$. Hence, for every j with $0 \leq j \leq m - 2$,

$$r \leq x_0 \leq x_j \leq x_{m-2} = 2^r - 2^d - d + m - 2 \leq 2^r - 1,$$

and $x_j \in [r, 2^r - 1] \subseteq \Lambda_r$ by the induction hypothesis (H_r), which applies since $7 \leq r < m$.

Proof of (H_m). By (U) and (H_{m-1}), $\Lambda_m \supseteq 1 + [m-1, 2^{m-1} - 1] = [m, 2^{m-1}]$. By ($\mathrm{G}_m$), $\Lambda_m \supseteq V_m \supseteq [2^{m-1}, 2^m - 1]$. □

Example 4.7. Take $k = 128 = 2^7$. Then $m = 8$ and $j = 0$, and the proof uses $(d, r) = (2, 4)$, with $x_0 = 2^4 - 2^2 - 2 = 10 = c(P_4)$. So the near-cone $N(2, P_4)$, formed by the path P_4 and the vertices u, ℓ_1, ℓ_2, w, is a connected graph with 8 vertices and

$$c(N(2, P_4)) = 2^7 - 2^4 + 2^2 + 2 + 10 = 128.$$

Seven vertices are not enough, since then $c \leq 2^7 - 1$.

5. Proofs of the main theorems

Proof of Theorem 1.3. (a) For $m = 6, 7$ this is Lemma 4.4(b), and for $m \geq 8$ it is Theorem 4.6. (b) For $m = 7$ this is Lemma 4.4(a), and for $m \geq 8$ it is Theorem 4.6. □

Proof of Theorem 1.2. Let $k \notin \{2, 4, 5, 8, 9, 16\}$ and put $m = \lfloor \log_2 k \rfloor + 1$, so that $2^{m-1} \leq k \leq 2^m - 1$. A connected graph with fewer than m vertices has $c \leq 2^{m-1} - 1 < k$, by Lemma 2.1(f). It remains to show that $k \in V_m$ when $k \neq 25$, and that $25 \in V_6$. If $k \leq 15$, then $k \in \{1, 3, 6, 7\} \cup [10, 15]$ and

$k \in V_m$ by Lemma 3.1. If $17 \le k \le 31$ and $k \ne 25$, then $m = 5$ and $k \in V_5$ by Proposition 3.4. If $k \ge 32$, then $m \ge 6$ and $k \in V_m$ by Theorem 1.3(a). Finally $25 \notin V_5$ by Proposition 3.4, and the spider with legs 1, 2, 2 shows $25 \in V_6$ (Lemma 4.3). Table 4 lists the witnesses for the values below 32, where the order is given by the finite base rather than by the induction. □

k	least order	a connected graph with $c = k$ of least order
1; 3	1; 2	K_1; K_2
6; 7	3	P_3; K_3
$10, 11, 12, 13, 14, 15$	4	P_4, $K_{1,3}$, paw, C_4, diamond, K_4 (Lemma 3.1)
$[17, 24] \cup [26, 31]$	5	Table 2, row $m = 5$, and Appendix A; 19 is sporadic
25	6	the spider with legs $1, 2, 2$ (Lemma 4.3)
$k \ge 32$	$\lfloor \log_2 k \rfloor + 1$	Lemma 4.4(b) for $m = 6, 7$; Theorem 4.6 and Remark 5.1 for $m \ge 8$

TABLE 4. Least orders and witnesses (Theorem 1.2). For each $k \le 31$ the least order is $\lfloor \log_2 k \rfloor + 1$, except for $k = 25$.

Proof of Theorem 1.1. If $k = c(G)$ for a connected graph G, then $k \notin \{2, 4, 5, 8, 9, 16\}$ by Corollary 3.5. Conversely, every positive integer outside this set is a value of c on a connected graph by Theorem 1.2. □

Remark 5.1 (An algorithm). The proofs are constructive. For $m \ge 7$ and $v \in [m, 2^m - 1]$, the following recursive procedure returns a graph $\mathrm{Re}(m, v)$ with m vertices and $c = v$, which is connected when $v \ge 2^{m-1}$.

Input: integers $m \ge 7$ and $v \in [m, 2^m - 1]$, in binary. *Output:* the edge list of $\mathrm{Re}(m, v)$ on the vertices $0, \dots, m - 1$.

(1) If $m = 7$, return the graph given by the proof of Lemma 4.4, from a fixed table.
(2) If $v < 2^{m-1}$, return $\mathrm{Re}(m - 1, v - 1)$ together with an isolated vertex.
(3) If $v \ge 2^{m-1} + m - 1$, return the cone over $\mathrm{Re}(m - 1, v - 2^{m-1})$.
(4) Otherwise put $j = v - 2^{m-1}$, choose (d, r) as in the proof of Theorem 4.6 (Table 3 for $8 \le m \le 11$, the least d with $2^d + d \ge m - 1$ for $m \ge 12$), and return $N(d, R)$. Here $R = \mathrm{Re}(r, x_j)$ if $r \ge 7$, and R is the graph given by the proof of Lemma 3.1 or Lemma 4.4, from a fixed table, if $r \le 6$.

The arguments of the recursive calls are admissible: in (2), $v - 1 \in [m - 1, 2^{m-1} - 1]$; in (3), $v - 2^{m-1} \in [m - 1, 2^{m-1} - 1]$; and in (4), $x_j \in [r, 2^r - 1]$ by the proof of Theorem 4.6. Each call makes at most one recursive call, with fewer vertices, so there are at most $m - 6$ calls. We count bit operations. Apart from its recursive call, a call compares and subtracts integers of at most $m + 1$ bits, finds d by at most $\log_2 m + 1$ trials on integers of $O(\log m)$ bits, and computes x_j, using $O(m)$ bit operations in all. It also adds at most $2m$ edges: none in (2), $m - 1$ in (3), and $2d + r = m - 2 + d$ in (4). The graph returned by the recursive call keeps its vertices $0, \dots, m' - 1$, and the new vertices receive the next labels, so nothing is relabelled. With vertex labels of $O(\log m)$ bits, the new edges take $O(m \log m)$ bits to write. The tables used in (1) and (4) have constant size. Hence $\mathrm{Re}(m, v)$ is computed with $O(m^2 \log m)$ bit operations, and its output, a graph with at most $\binom{m}{2}$ edges, has $O(m^2 \log m)$ bits. For $k \ge 64$, the graph $\mathrm{Re}(m, k)$ with $m = \lfloor \log_2 k \rfloor + 1 \ge 7$ is connected, has the least possible number of vertices, and is found with $O(\log^2 k \cdot \log \log k)$ bit operations, in particular in time polynomial in $\log k$. For $32 \le k \le 63$ the procedure does not apply, since $m = 6$; a connected graph with six vertices and $c = k$ is given by the proof of Lemma 4.4(b). The procedure cannot simply be extended to $m = 6$, because $26 \notin \Lambda_6$. For example, $c = 10^{100}$ is realized on 333 vertices. Section 9 reports a check of this procedure.

6. Values on a fixed number of vertices

Theorem 1.3 describes the upper half $[2^{n-1}, 2^n - 1]$ of the range (1). The lower half behaves less regularly. Table 5 gives V_n for $n \leq 8$. For $n \leq 5$ the table follows from Lemma 3.1 and Proposition 3.4; for $n = 6, 7, 8$ it comes from the computer census of Section 9, which confirms [7, Remark 3.20].

n	$T(n)$	$2^n - 1$	integers in $[T(n), 2^n - 1]$ not in V_n
1	1	1	—
2	3	3	—
3	6	7	—
4	10	15	—
5	15	31	16, 25
6	21	63	22, 23, 26
7	28	127	29, 30, 31, 35
8	36	255	37, 38, 39, 40, 43, 50, 55

Table 5. The sets V_n for $n \leq 8$. For $n \geq 5$, the first $n - 4$ missing values are $T(n) + 1, \ldots, T(n) + n - 4$ (Corollary 3.3).

Proposition 6.1. *Let $n \geq 6$. Then $T(n) \in V_n$ and $T(n) + n - 3 \in V_n$, none of the $n - 4$ integers $T(n) + 1, \ldots, T(n) + n - 4$ lies in V_n, and every integer in $[2^{n-1}, 2^n - 1]$ lies in V_n. In particular, at least $n - 4$ integers of $[T(n), 2^n - 1]$ are not values of c on connected graphs with n vertices, and all of them are smaller than 2^{n-1}.*

Proof. Combine Proposition 3.2, Corollary 3.3 and Theorem 1.3(a). □

So the number of missing values in $[T(n), 2^n - 1]$ grows at least linearly. For $5 \leq n \leq 8$ the missing values are the gap above the path together with one or three further values.

Question 6.2. Determine V_n for all n. In particular, is the number of integers in $[T(n), 2^n - 1] \setminus V_n$ bounded by a linear function of n?

7. Related counts

7.1. **Subtrees.** For a tree T, $c(T)$ is the number of subtrees of T, and Czabarka, Székely and Wagner [8] showed that every positive integer outside the set

$$E_T = \{2, 4, 5, 7, 8, 9, 12, 13, 14, 16, 18, 19, 22, 23, 26, 27, 29,\\ 31, 33, 35, 38, 39, 42, 43, 46, 50, 52, 54, 60, 65, 68, 72, 77, 89\}$$

of 34 exceptions is the number of subtrees of some tree. Thus allowing cycles reduces the 34 exceptions for trees to the six exceptions of Theorem 1.1.

Remark 7.1 (A computer-assisted route). The theorem of [8] gives a second proof of Theorem 1.1. Every positive integer outside E_T is a value of c on a tree. The elements of E_T other than $2, 4, 5, 8, 9, 16$ are values of c on connected graphs with at most seven vertices, by Lemma 3.1 and Appendix A. With Corollary 3.5, this gives Theorem 1.1. This route relies on the computer-assisted parts of [8] and on a census of small graphs, and it gives no information on the number of vertices.

7.2. **All connected subgraphs.** A *connected subgraph* of G is a connected graph (V', E') with $\emptyset \neq V' \subseteq V(G)$ and $E' \subseteq E(G[V'])$. Let $F(G)$ be the number of connected subgraphs of G; Pandey and Patra [19] call it the subgraph index and study its extremal values. Let $L(G)$ denote the line graph of G.

Lemma 7.2. *Let G be a graph with n vertices.*

(a) $F(G) = n + c(L(G))$.
(b) $c(G) \leq F(G)$, *with equality if and only if G is a forest.*
(c) *If G' is obtained from G by adding an edge, then $F(G') > F(G)$.*

Proof. (a) The connected subgraphs without edges are the n single vertices. A connected subgraph with a nonempty edge set E' has vertex set $V(E')$, the set of ends of edges of E', and the graph $(V(E'), E')$ is connected if and only if E' is a connected set of $L(G)$.

(b) Each connected set S gives the connected subgraph $G[S]$. If G is a forest, then $G[V']$ is a forest, so a connected spanning subgraph of $G[V']$ must be $G[V']$ itself. If G has a cycle, the cycle minus one of its edges is a connected subgraph that is not induced.

(c) All connected subgraphs of G remain, and the new edge with its two ends is a new one. □

Proposition 7.3. *A positive integer k equals $F(G)$ for some finite connected graph G if and only if*

$$k \notin E_F = \{2, 4, 5, 7, 8, 9, 12, 13, 14, 16, 19, 22, 23, 35, 38, 39, 46\}.$$

Proof (computer-assisted). By Lemma 7.2(b) and [8], every positive integer outside E_T is a value of F on a tree, and $\max E_T = 89$. It remains to decide which elements of E_T are values of F. We separate the reduction, which is proved, from the finite search, which is a computation.

Reduction. Let G be connected with n vertices and $F(G) \leq 89$. By Lemma 7.2(b) and Lemma 2.1(f), $T(n) \leq c(G) \leq F(G) \leq 89$, so $n \leq 12$. Let S be a spanning tree of G, and add the edges of $E(G) \setminus E(S)$ to S one at a time. This gives connected graphs $S = G_0, G_1, \ldots, G_s = G$, each obtained from the previous one by adding one edge. By Lemma 7.2(c), $F(G_0) < F(G_1) < \cdots < F(G_s) \leq 89$, and $F(G_0) = c(S)$ by Lemma 7.2(b). So every connected graph with $F \leq 89$ is reached from a tree with $c \leq 89$ and at most 12 vertices by adding one edge at a time, and every intermediate graph also has $F \leq 89$. Being reachable in this way does not depend on the labeling, so the search may be carried out up to isomorphism.

Search. Level 0 consists of the trees with at most 12 vertices and $c \leq 89$, one from each isomorphism class; all trees are generated by NetworkX [13]. There are 94 of them: 1, 1, 1, 2, 3, 6, 11, 20, 24, 17, 6 and 2 trees on $1, 2, \ldots, 12$ vertices, respectively. Level $i+1$ consists of the graphs obtained from a graph of level i by adding one missing edge. A graph is discarded if $F > 89$, or if it is isomorphic to a graph found before. The isomorphism test compares Weisfeiler–Lehman hashes and then tests exact isomorphism. Levels $1, 2, 3, 4$ contain 69, 13, 1 and 0 graphs, so the search stops. It finds 177 pairwise non-isomorphic connected graphs with $F \leq 89$. By the reduction, every connected graph with $F \leq 89$ is isomorphic to one of them, and their values of F are all integers in $[1, 89]$ except the elements of E_F. The 17 elements of $E_T \setminus E_F$ are realized by the graphs of Table 6. The values of F were computed in two independent ways, by Lemma 7.2(a) and by testing all edge subsets; Section 9 describes the programs and the machine-readable list of the 177 graphs. □

For instance, $F(C_5) = 5 + c(C_5) = 26$, the paw has $F = 4 + c(\text{diamond}) = 18$, and the diamond has $F = 4 + c(W_4) = 4 + 29 = 33$, where W_4, the cone over C_4, is the line graph of the diamond. Unlike Theorem 1.1, Proposition 7.3 depends on computer calculations, both through [8] and through our search. A proof in the spirit of Section 4 would be of interest.

7.3. **The topological interpretation.** As explained in Section 1.1, Theorem 1.1 answers [7, Question 3.21]. Combined with the reduction of [7] and a realization of every finite connected

F	n	edges	F	n	edges
18	4	01 02 03 13 (paw)	54	5	01 02 03 13 24 34
26	5	01 02 13 24 34 (C_5)	60	7	01 02 12 14 23 45 56
27	5	01 02 12 13 34	65	8	01 02 15 23 34 47 56 67
29	5	01 02 04 13 34	68	7	01 02 05 14 23 45 46
31	5	01 02 03 13 34	72	7	01 02 12 13 15 34 56
33	4	01 02 03 12 13 (diamond)	77	7	01 02 03 04 15 16 56
42	6	01 02 05 14 23 45	89	8	01 02 13 15 23 24 56 67
43	6	01 02 14 23 25 45			
50	7	01 02 14 23 36 45 56			
52	6	01 02 03 05 14 45			

TABLE 6. Connected graphs realizing, as values of F, the 17 subtree exceptions of [8] that are values of F. Each graph has the least possible number of vertices and, among those, the least number of edges. Vertices are $0, \dots, n-1$, and ij denotes the edge joining i and j.

graph by copies of the Knaster continuum glued at points, it gives the following result of the companion paper.

Theorem 7.4 ([2, Theorem A])**.** *A positive integer k is the number of regular subcontinua of some metric continuum if and only if $k \notin \{2, 4, 5, 8, 9, 16\}$. For every $k \geq 32$ such a continuum can be chosen as the union of $\lfloor \log_2 k \rfloor + 1$ copies of the Knaster continuum, any two of which meet in at most one point, and no continuum with exactly k regular subcontinua has fewer minimal regular subcontinua.*

8. Formal verification in Lean

The results of Sections 2–5 and 6 and Lemma 7.2 are formalized in Lean 4 [9] with Mathlib [16], in the library `ConnectedSets` in the directory `lean` of the repository [3] of this paper. The file `MANIFEST.txt` there lists the SHA-256 checksums of all files of this version, including the Lean sources and the scripts of Section 9. The toolchain is Lean `v4.35.0` with Mathlib commit `6015dbbfc399f4548bafd46f9560ff61595fd328`, both fixed in the files `lean-toolchain`, `lakefile.toml` and `lake-manifest.json`. In the directory `lean`, the commands

```
lake exe cache get    lake build    lake env lean CheckAxioms.lean
```

download a compiled Mathlib (this step is optional), build the library, and print the axioms of every statement of Table 7. The library has 20 modules and 4623 lines. Eleven of them (`Basic`, `Defs`, `Rules`, `Transport`, `Checker`, `Five`, `Sporadic`, `Base`, `Intervals`, `Induction` and `Main`; 2201 lines) are the graph-theoretic part of the formalization written for the companion paper [2], which is in the directory `lean` of [4]. They are copied with the namespace renamed and the comments renumbered, and the scripts in `lean/gen` regenerate the modules `Sporadic`, `Base` and `Intervals` byte for byte. The other modules are new: `Gap` and `Path` (Proposition 3.2), `Spider`, `Small` (Lemmas 2.1(b) and 3.1, Corollaries 3.3 and 3.5, Propositions 3.4 and 6.1, and Theorem 1.2 for every k), `Glue` (Lemma 2.1(c)), `LineGraph` (Lemma 7.2), `Extras`, and `Paper`, which states the results with the numbering of this paper.

For a graph `G` on a finite type `V`, the number `c G` is the number of finite sets `S` of vertices such that the induced graph `G.induce S` is connected; Mathlib's notion of a connected graph includes nonemptiness. The statement `InV m k` says that some connected graph on `Fin m` has `c = k`, that is, $k \in V_m$, and `InLam m k` says $k \in \Lambda_m$. Paths, forests and line graphs are Mathlib's `pathGraph`, `IsAcyclic` and `lineGraph`. The number `F G` counts the pairs (S, E) of a finite set of vertices and

a finite set of edges of `G` with both ends in S such that the graph with vertex set S and edge set E is connected. Two of the statements of the module `Paper` read as follows; there `T n` is $T(n)$.

```
theorem theorem_1_1 (k : ℕ) (hk : 0 < k) :
    (∃ n, ∃ G : SimpleGraph (Fin n), G.Connected ∧ c G = k) ↔
      k ∉ ({2, 4, 5, 8, 9, 16} : Finset ℕ)

theorem proposition_3_2 (G : SimpleGraph V) (hG : G.Connected) :
    (IsEmpty (pathGraph (Fintype.card V) ≃g G) →
      T (Fintype.card V) + (Fintype.card V - 3) ≤ c G) ∧
    (c G = T (Fintype.card V) ↔ Nonempty (pathGraph (Fintype.card V) ≃g G))
```

Table 7 lists the correspondence between this paper and the module `Paper`; all its declarations are in the namespace `ConnectedSets.Paper`.

this paper	declarations in `ConnectedSets/Paper.lean`
Theorems 1.1, 1.2, 1.3	`theorem_1_1`, `theorem_1_1_type`, `theorem_1_2`, `theorem_1_3`, `theorem_1_3_sharp`
formulas of Section 1	`section_1_formulas` (path, complete graph, star)
Lemma 2.1(a)–(f)	`lemma_2_1_a`, ..., `lemma_2_1_f`
Lemma 3.1	`lemma_3_1_a`, `lemma_3_1_b`
Proposition 3.2	`proposition_3_2`, `proposition_3_2_spider`
Corollaries 3.3, 3.5; Proposition 3.4	`corollary_3_3`, `corollary_3_5`, `proposition_3_4`
Lemmas 4.2, 4.3, 4.4	`lemma_4_2`, `lemma_4_3`, `lemma_4_4_a`, `lemma_4_4_b`
Remark 4.5 ($25 \notin \Lambda_5$)	`remark_4_5`
Theorem 4.6, Example 4.7	`theorem_4_6`, `example_4_7`
Proposition 6.1	`proposition_6_1`
Lemma 7.2(a)–(c)	`lemma_7_2_a`, `lemma_7_2_b`, `lemma_7_2_c`

TABLE 7. Correspondence between the results of this paper and the Lean formalization.

The formal proofs follow the paper, with four differences.

(1) Proposition 3.2 is proved by a different argument, an induction on connected sets that avoids spanning trees. Call a vertex y of a connected set S an *end* of S if S has exactly one connected subset containing y of each size. An end forces S to have exactly $T(|S|)$ connected subsets, and it gives an enumeration of S in which two vertices are adjacent exactly when they are consecutive. Conversely, if S has exactly $T(|S|)$ connected subsets, then every vertex whose removal leaves S connected is an end. In the induction, remove such a vertex w. If $S \setminus \{w\}$ has the minimum number of connected subsets, it is a path, and the intervals of this path that form a connected set together with w give at least $2|S| - 3$ connected sets containing w, unless S itself has the minimum. Otherwise the induction hypothesis applies to $S \setminus \{w\}$. This gives $c(G) = T(n)$ or $c(G) \geq T(n) + n - 3$ for every connected graph G with n vertices, and $c(G) = T(n)$ exactly when G is isomorphic to `pathGraph n`.

(2) In Proposition 3.4 the values 16 and 25 are excluded by a kernel computation (`decide +kernel`) over all 2^{10} labeled graphs on five vertices, not by the case analysis of the paper; for 16 the formalized Corollary 3.3 gives a second proof. The computation uses a bitmask counting procedure proved equal to `c`. The same procedure checks the counts of the sporadic graphs, the 353 derivations of Lemma 4.4, the exclusions $5 \notin \Lambda_3$ and $9 \notin \Lambda_4$ of Lemma 3.1 (over all labeled graphs on three and four vertices), and $25 \notin \Lambda_5$ in Remark 4.5.

(3) Lemma 2.1(c) is formalized inside one graph: the vertex set is covered by sets A and B with $A \cap B = \{v\}$ and no edge between $A \setminus \{v\}$ and $B \setminus \{v\}$, and $G[A]$ and $G[B]$ play the roles of G_1 and G_2.

(4) Lemma 2.1(e) and the near-cone formula of Lemma 4.2 are stated without subtraction, for instance as $c(G) + 2^{|K|-1-d} = c(K) + 2^{|K|}$, where d is the size of a finite set listing the neighbours of u.

Not formalized are the general formulas of Example 2.2, whose instances used in the proofs are checked by the kernel; the computer census behind Table 5 for $n = 6, 7, 8$, and with it $26 \notin \Lambda_6$; the statements of Remark 4.5 about which values the rules produce; the running time in Remark 5.1; Remark 7.1 and Proposition 7.3, which rely on the computer-assisted theorem of [8] and on our search; and Appendix A. Theorem 7.4 is formalized in the companion formalization [2].

The library uses neither `sorry` nor `native_decide` and declares no axioms. Its 31 statements in the module `Paper` depend only on the axioms `propext`, `Classical.choice` and `Quot.sound`, and one of them, `theorem_1_2_minOrder`, depends on none. We checked the library in four ways, recorded in the file `CHECKS.txt` of the formalization. It builds from an empty build directory, and a copy of the sources in a fresh directory builds with the same output. The declarations of all modules were replayed through the kernel with `leanchecker`, and the whole import closure, including Lean core and Mathlib, with `leanchecker --fresh`. Finally, the statements of `Paper` were exported together with everything they depend on, 10538 declarations in all, and checked by the independent type checker `nanoda`, with only the three axioms above permitted; no errors were reported.

9. Computational verification

The proofs of Theorems 1.1–1.3 do not use a computer. We checked every explicit finite count in this paper, and the computer-assisted result of [7], with programs written for this purpose. The programs for the census of small graphs, the construction and the subtrees, with their outputs, are in the repository [4] of the companion paper; the file `CHECKS.md` there maps each computational statement of [2] to the program that checks it. The scripts for Proposition 7.3 and Appendix A, the construction program `construct.py` with its certificate for Lemma 4.4, and the outputs are in the directories `code` and `data` of the repository [3] of this paper.

Counting. We compute $c(G)$ in three ways: by testing all $2^n - 1$ nonempty vertex sets; by the ESU enumeration scheme [25], which lists each connected set once; and by a recurrence over vertex sets. In the recurrence, $g(S) = 1$ if $G[S]$ is connected and $g(S) = 0$ otherwise. The component of $G[S]$ containing $\min S$ is the only connected set $T \subseteq S$ with $\min S \in T$ and no edge between T and $S \setminus T$. Hence $g(S) = 1 - \sum g(T)$, the sum running over sets $T \neq S$ with $\min S \in T \subseteq S$ and no edge between T and $S \setminus T$.

Graphs on at most eight vertices. Program A takes the graphs on at most seven vertices from the atlas of Read and Wilson [20], as distributed with NetworkX [13]. It extends each of the 1044 graphs on seven vertices, connected or not, by every nonempty neighbourhood of a new vertex, and removes isomorphic copies in two independent ways. Both give the 11117 connected graphs on eight vertices of OEIS A001349 [17]. Program B, written in C, uses no atlas and no isomorphism test. It runs over all labeled graphs on at most seven vertices, and over the graphs on eight vertices obtained by extending the labeled graphs on seven vertices with nonincreasing degree sequence. Its numbers of labeled connected graphs agree with OEIS A001187. The two programs give the same sets V_n for $n \leq 8$, as in Table 5. Every value in $[1, 255]$ except $2, 4, 5, 8, 9, 16$ occurs, and every value in $[17, 127]$ occurs on at most seven vertices. This confirms [7, Remark 3.20]. Program B also rederives each witness of Appendix A by brute force over all labeled graphs.

The construction. We implemented Lemma 4.4 and the procedure of Remark 5.1 literally, and checked the gaps stated in Remark 4.5. An independently written program re-checked each of the 353 derivations for $m \leq 7$. For every $k \in [17, 70000]$, and for 300 random $k < 2^{22}$, we built the resulting graph and counted its connected sets. For every $8 \leq m \leq 16$ and every $v \in [m, 2^m - 1]$,

the graph $\mathrm{Re}(m, v)$ has exactly m vertices, is connected when $v \geq 2^{m-1}$, and has $c = v$; in total 130708 graphs were checked.

Subtrees. We re-verified the list E_T of [8] from all trees with at most 18 vertices, generated and counted in two independent ways. A tree with $n \geq 19$ vertices has at least $T(19) = 190$ subtrees, so this determines the subtree counts below 190.

All connected subgraphs. The script `code/f_values.py` of [3] carries out the search in the proof of Proposition 7.3; the command `python3 f_values.py 89` runs it, with Python 3.13 and NetworkX 3.7, in about two seconds. During the search it computes F through Lemma 7.2(a), counting the connected sets of the line graph with the ESU scheme and stopping as soon as the count exceeds 89. For every graph found it then recomputes F by testing all nonempty edge subsets with a union-find structure; the two methods agree on all 177 graphs. The script prints the number of trees of each order at level 0, the size of each level, and the missing values, as stated in the proof (output file `f_values_output.txt`). It also writes the file `f_graphs_le_89.csv`, which lists the 177 graphs with F, c, the numbers of vertices and edges, and the edge list. Together with the reduction in the proof, this list certifies both that the values in $[1, 89] \setminus E_F$ occur and that the elements of E_F do not. A separate program in C, `f_edge_subsets.c`, recomputed c by vertex subsets and F by edge subsets for all 996 connected graphs with at most seven vertices of the atlas, independently of the search. It agrees with the values below 36, where those graphs suffice, since $F \geq T(n) \geq 36$ for $n \geq 8$.

AI disclosure

Artificial intelligence was used extensively in this work, both to obtain the results and to check them.

Solving and proving. The main model used to solve the problem and to prove the theorems was Claude Opus 5.5 (Anthropic). In an iterative process directed by the author, it carried out most of the exploratory proof development, including the near-cones of Section 4 and the arguments of Section 3. It also wrote and ran the programs of Section 9, carried out the literature search of Section 1.2, and generated most of the first draft of this manuscript and of the Lean 4 formalization of Section 8.

Reading and checking. ChatGPT-Astra (OpenAI) was used, together with the author, to read the manuscript and check the arguments. AI was used extensively for proofreading and proof checking, and for repeated rounds of checking the arguments and comparing the proofs with one another, for instance the two proofs of Proposition 3.2 and the two exclusions of 16 on five vertices. In each round the author, with ChatGPT-Astra, reviewed the text, the proofs and the supplementary material and sent feedback, and Claude revised them.

The author's role. The author formulated the problem, which arose from Questions 3.5 and 3.21 of [7], provided the mathematical context and constraints, and steered the iterations through prompts and feedback. The author also supplied ideas and suggested strategies, selected and revised the outputs, and reviewed the final arguments.

The explicit finite counts are checked by programs (Section 9), and the results listed in Table 7 are formally verified in Lean (Section 8). No independent expert has yet verified the proofs. The author takes full responsibility for the content.

References

[1] Y. Alizadeh, N. Bašić, I. Damnjanović, T. Došlić, T. Pisanski, D. Stevanović, and K. Xu, *Solving the Mostar index inverse problem*, preprint, arXiv:2312.13083.

[2] L. Alves de Almeida, *All finite cardinalities of the hyperspace of regular subcontinua*, preprint, version 2.3, 2026, available in [4].

[3] L. Alves de Almeida, *The inverse problem for the number of connected induced subgraphs: paper, Lean formalization and programs*, GitHub repository, https://github.com/LucasAlves8Sp/THE-INVERSE-PROBLEM-FOR-THE-NUMBER-OF-CONNECTED-INDUCED-SUBGRAPHS.
[4] L. Alves de Almeida, *All finite cardinalities of the hyperspace of regular subcontinua: paper, programs and Lean formalization*, GitHub repository, https://github.com/LucasAlves8Sp/ALL-FINITE-CARDINALITIES-OF-THE-HYPERSPACE-OF-REGULAR-SUBCONTINUA.
[5] E. O. D. Andriantiana and A. A. V. Dossou-Olory, *Nordhaus–Gaddum inequalities for the number of connected induced subgraphs in graphs*, preprint, arXiv:2006.01187.
[6] C. Bernert and J. Shaw, *The inverse problem for the Steiner–Wiener index via additive number theory*, preprint, 2026, arXiv:2603.17559.
[7] J. Camargo, N. Ordoñez, and D. Ramírez, *On the hyperspaces of meager and regular continua*, Appl. Gen. Topol. **25** (2024), no. 2, 385–406. doi:10.4995/agt.2024.20116.
[8] É. Czabarka, L. A. Székely, and S. Wagner, *The inverse problem for certain tree parameters*, Discrete Appl. Math. **157** (2009), no. 15, 3314–3319. doi:10.1016/j.dam.2009.07.004.
[9] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction – CADE 28, Lecture Notes in Comput. Sci. **12699**, Springer, Cham, 2021, 625–635. doi:10.1007/978-3-030-79876-5_37.
[10] A. A. V. Dossou-Olory, *Graphs and unicyclic graphs with extremal number of connected induced subgraphs*, preprint, arXiv:1812.02422.
[11] A. A. V. Dossou-Olory, *Cut and pendant vertices and the number of connected induced subgraphs of a graph*, preprint, arXiv:1910.04552.
[12] A. A. V. Dossou-Olory, *On the maximum number of connected induced subgraphs of a graph*, preprint, 2023, arXiv:2303.01964.
[13] A. A. Hagberg, D. A. Schult, and P. J. Swart, *Exploring network structure, dynamics, and function using NetworkX*, in: Proceedings of the 7th Python in Science Conference (SciPy 2008), 11–15.
[14] J. Haslegrave, *The number and average size of connected sets in graphs with degree constraints*, preprint, arXiv:2105.13332.
[15] V. Linek, *Bipartite graphs can have any number of independent sets*, Discrete Math. **76** (1989), 131–136. doi:10.1016/0012-365X(89)90306-3.
[16] The mathlib Community, *The Lean mathematical library*, in: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020), 367–381. doi:10.1145/3372885.3373824.
[17] OEIS Foundation Inc., *The On-Line Encyclopedia of Integer Sequences*, sequences A001187 and A001349, https://oeis.org.
[18] N. Ordoñez, *The hyperspace of regular subcontinua*, Topology Appl. **234** (2018), 415–427. doi:10.1016/j.topol.2017.11.038.
[19] D. Pandey and K. L. Patra, *On the number of connected subgraphs of graphs*, Indian J. Pure Appl. Math. **52** (2021), no. 2, 571–583. doi:10.1007/s13226-021-00061-4.
[20] R. C. Read and R. J. Wilson, *An Atlas of Graphs*, Oxford University Press, Oxford, 1998.
[21] L. A. Székely and H. Wang, *On subtrees of trees*, Adv. in Appl. Math. **34** (2005), 138–155. doi:10.1016/j.aam.2004.07.002.
[22] P. Tittmann, I. Averbouch, and J. A. Makowsky, *The enumeration of vertex induced subgraphs with respect to the number of components*, European J. Combin. **32** (2011), no. 7, 954–974. doi:10.1016/j.ejc.2011.03.017.
[23] A. Vince, *Counting connected sets and connected partitions of a graph*, Australas. J. Combin. **67** (2017), no. 2, 281–293.
[24] H. Wang and G. Yu, *All but 49 numbers are Wiener indices of trees*, Acta Appl. Math. **92** (2006), 15–20. doi:10.1007/s10440-006-9037-2.
[25] S. Wernicke, *Efficient detection of network motifs*, IEEE/ACM Trans. Comput. Biol. Bioinform. **3** (2006), no. 4, 347–359. doi:10.1109/TCBB.2006.51.

Appendix A. Witnesses for the values from 17 to 127

A *witness* for an integer k is a connected graph G with $c(G) = k$. We record witnesses as graph6 strings. A *graph6 string* is a standard printable encoding of a finite labeled graph with vertex set $\{0, \dots, n-1\}$. For $n \leq 62$ its first character is the one with ASCII code $n + 63$. The remaining characters encode the entries above the diagonal of the adjacency matrix, read column by column, that is, in the order $(0,1), (0,2), (1,2), (0,3), (1,3), (2,3), \dots$. These bits are padded with zeros to a multiple of six, and each group of six bits, read as a binary number b, is written as the character

with ASCII code $b+63$. For example, `D?{` encodes the star $K_{1,4}$ with centre 4: the bits of `?` and `{` are 000000 and 111100, so the edges are $(0,4),(1,4),(2,4),(3,4)$.

Canonical strings. A graph6 string depends on the labeling. For a fixed n all graph6 strings have the same length, and we compare them lexicographically by their ASCII characters. The *canonical graph6 string* of a graph is the least graph6 string among all its labelings with vertex set $\{0,\dots,n-1\}$; two graphs are isomorphic if and only if their canonical strings coincide.

Witness rule. For each $k \in [17,127]$ the table gives a witness of least possible order n; among those, of least number of edges; and among those, the one with the least canonical graph6 string. The table lists k, n and the canonical string. By Theorem 1.2, $n = \lfloor \log_2 k \rfloor + 1$ except for $k = 25$. Program B of Section 9 rederives every entry by brute force over all labeled graphs on at most seven vertices. Edge lists are in the file `data/witnesses_small.csv` of [3]. The script `code/check_witnesses.py` of [3] reads this table from the LaTeX source of the paper, decodes each string with the rule above, and checks that it gives a connected graph with n vertices and exactly k connected sets; all 111 rows pass.

k	n	graph6	k	n	graph6	k	n	graph6
17	5	`D@s`	54	6	`EB~o`	91	7	`F?Nv_`
18	5	`DJc`	55	6	`EFz_`	92	7	`F?Fno`
19	5	`DBk`	56	6	`EFzg`	93	7	`F@Q~o`
20	5	`D?{`	57	6	`EL~o`	94	7	`F?Fnw`
21	5	`D@{`	58	6	`EFzw`	95	7	`F?]~_`
22	5	`DJk`	59	6	`EF~w`	96	7	`F@Vn_`
23	5	`DB{`	60	6	`E]~o`	97	7	`F@N^W`
24	5	`DLs`	61	6	`E]~w`	98	7	`F?Nvo`
25	6	`E@QW`	62	6	`E^~w`	99	7	`F@vf_`
26	5	`DFw`	63	6	`E~~w`	100	7	`F?B~o`
27	5	`DF{`	64	7	`F?DnO`	101	7	`F?B~w`
28	5	`DN{`	65	7	`F??^o`	102	7	`F?F~w`
29	5	`D]{`	66	7	`F?C^W`	103	7	`F?N~o`
30	5	`D^{`	67	7	`F?KuW`	104	7	`F?^vo`
31	5	`D~{`	68	7	`F?Cng`	105	7	`F@^vo`
32	6	`E?NW`	69	7	`F?@|w`	106	7	`F@vv_`
33	6	`E@Ug`	70	7	`F??Fw`	107	7	`F?^~o`
34	6	`EJ]G`	71	7	`F??Nw`	108	7	`FBnnO`
35	6	`E@Uw`	72	7	`F?Cfw`	109	7	`F@vvo`
36	6	`E?No`	73	7	`F??^w`	110	7	`FBzv_`
37	6	`E?Bw`	74	7	`F?C^w`	111	7	`F@vvw`
38	6	`E?Fw`	75	7	`F@Q^G`	112	7	`F?~v_`
39	6	`E@Rw`	76	7	`F?Cnw`	113	7	`F?~vg`
40	6	`E?Nw`	77	7	`F??~w`	114	7	`F@~vg`
41	6	`E@Nw`	78	7	`F?Dno`	115	7	`F?~vw`
42	6	`E?^o`	79	7	`F@Um_`	116	7	`F?~~w`
43	6	`E@Vw`	80	7	`F?LVw`	117	7	`F@~~w`
44	6	`E?^w`	81	7	`F@Q^W`	118	7	`FFzng`
45	6	`E@vg`	82	7	`F?Dnw`	119	7	`FFzfw`
46	6	`EBn_`	83	7	`F?@~o`	120	7	`FFznw`
47	6	`E@vo`	84	7	`F?D~o`	121	7	`FFz~o`
48	6	`EBz_`	85	7	`F?@~w`	122	7	`FFz~w`
49	6	`E@vw`	86	7	`F?Fng`	123	7	`FF~~w`
50	6	`EJfw`	87	7	`F@Umg`	124	7	`F]~vw`
51	6	`E?~o`	88	7	`F?NNg`	125	7	`F]~~w`
52	6	`E?~w`	89	7	`F?N^_`	126	7	`F^~~w`
53	6	`E@~w`	90	7	`F@]v_`	127	7	`F~~~w`

Independent researcher
Email address: lucasderiva8x@gmail.com