

# TREES AND GRAPHS WITH NON LOG-CONCAVE DOMINATING SET SEQUENCE VIA AI TOOLS

ALINA DU, STEVEN HEILMAN, GRETA PANOVA

ABSTRACT. We give new examples of graphs and trees with dominating set sequences that are not log-concave. These examples were generated by PatternBoost, a transformer-based reinforcement learning software developed by Charton-Ellenberg-Wagner-Williamson. We also show: for any positive integer  $m$ , there exists a tree whose dominating set sequence is not log-concave for at least  $m$  indices by modifying a similar construction of Bautista-Ramos for the independent set sequence. We show that a large class of caterpillar graphs has log-concave dominating set sequences. A continuous analogue of the sequence is also log-concave for all graphs.

## 1. INTRODUCTION

Let  $G$  be a finite undirected graph on  $n$  vertices. A dominating set in a graph is a set of vertices which, together with their neighbors, is equal to all vertices in the graph. In other words, every vertex of the graph is either in the set or is a neighbor of a vertex from that set. For any  $0 \leq j \leq n$ , let  $d_j(G)$  be the number of dominating sets in  $G$  with cardinality  $j$ .

The following was conjectured in [AP14].

**Conjecture 1.1.** *For any finite graph  $G$ , the dominating set sequence  $d_0(G), \dots, d_n(G)$  is unimodal. That is, there exists some  $0 \leq m \leq n$  such that*

$$d_0(G) \leq d_1(G) \leq \dots \leq d_m(G) \geq d_{m+1}(G) \geq \dots \geq d_n(G).$$

This conjecture remains widely open even for trees. Since such a basic question remains unanswered, one might reasonably question whether Conjecture 1.1 is true or not.

There are not many available techniques to prove unimodality, especially in general settings without algebraic structure. A stronger property, which implies unimodality, is log-concavity, which enjoys some better developed technology including properties like real-rootedness of the corresponding polynomial, Lorentzian properties, injective maps, etc.

The dominating set sequence is log-concave if

$$d_k(G)^2 \geq d_{k+1}(G)d_{k-1}(G), \quad \forall 1 \leq k \leq n-1.$$

In the attempt to disprove Conjecture 1.1, one may try to first disprove the stronger log-concavity property, find non log-concave families of graphs with patterns and try to amplify the structures. Alternatively, for certain families of graphs, one may try to prove log-concavity as a stronger but more technologically feasible property that implies unimodality.

In this paper we produce new families of trees and examples of more general graphs with non log-concave dominating set sequence. We produce these counterexamples using PatternBoost [CEWW24], which is a transformer-based reinforcement learning technology.

---

*Date:* May 3, 2026.

Email: alinadu@usc.edu, stevenmheilman@gmail.com, gpanova@usc.edu

S.H. is supported by NSF Grant CCF AF 2448108. G.P. is supported by NSF Grant CCF AF 2302174.

MSC Classification: 68T05, 05C69, 05C05

Keywords: Transformer, AI, Dominating set, graph, tree

Department of Mathematics, University of Southern California, Los Angeles, CA 90089.

We also show that legged caterpillars (caterpillars with at least one leaf attached to each spine vertex) have log-concave dominating set sequence in Theorem 3.3. We give a general polytope approach which shows that the continuous version of this problem is true in Theorem 3.2, showing that the failures are a discrete phenomenon related to integer point counts. We also find a parametrized family of trees that breaks log-concavity at arbitrarily many places in Theorem 1.2.

**1.1. Background and motivation.** The study of unimodality and log-concavity is especially compelling because such properties frequently reflect deep underlying structure: for example, log-concavity is closely tied to negative dependence (anti-correlation) and algebraic or geometric phenomena in a wide range of settings, including matroid theory and graph polynomials (see e.g. [AHK18]).

Very closely related to unimodality of dominating sets is the analogous problem for independent sets. However, already in their early work, Alavi, Erdős, Malde and Schwenk [AMSE87] observed that the sequence of independent sets of a graph can have an arbitrary ascent/descent pattern. They conjectured that for trees this sequence remains unimodal, perhaps since non-unimodal graphs are very dense, unlike trees. Independent sets of trees have received significant attention over the past 10 or so years, with strong evidence for unimodality [GH18], but also many breaking points of log-concavity recently discovered with PatternBoost [RS25].

However, for dominating sets the situation remains much less understood.

The smallest graph with a non log-concave dominating set sequence has 9 vertices [BB22].

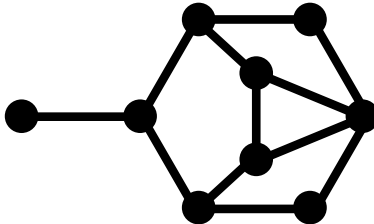


FIGURE 1. The unique graph on 9 vertices with non log-concave dominating set sequence: 0, 0, 1, 7, 50, 89, 75, 35, 9, 1

We are not aware of other published examples of such non-tree graphs. Alikhani and Peng [AP14] showed that for any  $n$ -vertex graph  $G$ ,

$$d_0(G) \leq \dots \leq d_{\lceil n/2 \rceil}(G).$$

Beaton showed [B21] that for any  $n$ -vertex graph without isolated vertices,

$$d_{\lfloor 3n/4 \rfloor - 1}(G) \geq \dots \geq d_n.$$

It was also shown in [BB22, Theorem 3.2] that, if a graph  $G$  on  $n$  vertices has all degrees larger than  $2 \log_2 n$ , then  $G$  has unimodal dominating set sequence with mode  $\lfloor n/2 \rfloor$ . That is, any counterexample to Conjecture 1.1 with  $n$  vertices must have a vertex with degree smaller than  $2 \log_2 n$ . In this sense trees could be considered a “difficult” case of Conjecture 1.1, since any tree has at least two degree one vertices.

One might ask whether random graphs exhibit log-concavity. Dense Erdős-Rényi graphs have unimodal dominating set sequence with probability tending to 1 as  $n \rightarrow \infty$  (when the probability  $p \in (0, 1)$  of any edge appearing is fixed) [BB22, Theorem 3.3]. This follows readily from Hoeffding’s inequality and the  $2 \log_2 n$  unimodality result, since such a random graph satisfies this degree lower bound with high probability. For this reason, the argument of [BB22] does not extend to sparse Erdős-Rényi graphs, where the probability of any edge appearing goes to zero as  $n \rightarrow \infty$ . In this

regime, perhaps the corresponding result for independent sets could be extended to dominating sets [Hei25].

For a while it was an open problem to find a tree whose dominating set sequence was not log-concave. Such an example was produced in [BS24], inspired by a tree whose independent set sequence is not log-concave [KL23]. Interestingly, PatternBoost also found these same tree examples (see Section 6).

It was also shown in [BS24, Theorems 6 and 16] that, if  $T$  is a tree with  $n$  vertices, then

$$d_{\gamma(T)} \leq \cdots \leq d_{\lfloor (n+2\gamma(T)+1)/3 \rfloor}, \quad d_{\lceil (n+2\Gamma(T)-2)/3 \rceil} \geq \cdots \geq d_n \quad (1.1)$$

where  $\Gamma(T)$  is the size of the largest minimal dominating set in  $T$ , and  $\gamma(T)$  is the size of the smallest dominating set in  $T$ .

It seems plausible that random trees would exhibit some log-concavity with high probability (by analogy with independent sets [Hei25, BG21]), though we are not aware of any results of this type.

**1.2. Proofs and counterexamples.** Here we explore aspects of Conjecture 1.1. In Section 3 we provide a geometric construction interpreting the numbers  $d_j(G)$  as integer point counts in polytopes and show that their approximation via volumes gives a log-concave sequence. We also show that a large class of caterpillar graphs we call *legged caterpillars* have log-concave domination polynomials in Theorem 3.3.

We produce new examples of trees and graphs with non log-concave dominating set sequence, using PatternBoost [CEWW24]. See Sections 5 and 6 for examples of these graphs and trees.

The application of this transformer-based reinforcement learning technology is significant since it elaborates upon known counterexamples for a difficult graph theoretic problem (Conjecture 1.1) with an automated method that does not use any explicit knowledge of the mathematical literature. Unlike approaches to mathematics based on LLMs that use vast amounts of training data (including research papers, websites with mathematical content, etc.), PatternBoost is a reinforcement learning optimization algorithm that generates its own training data with no access to the internet or existing mathematical literature. These examples are thus generated de novo, although some thought is needed to initialize the algorithm correctly.

Our code is available at the following GitHub link:

[https://github.com/sheilman77/dominating\\_sets\\_PatternBoost](https://github.com/sheilman77/dominating_sets_PatternBoost)

Lastly, without using PatternBoost, we modify a construction of [B25] to produce a family of trees whose dominating set sequence can fail log-concavity at an arbitrarily large number of indices.

**Theorem 1.2.** *For any positive integer  $m \geq 1$ , there exists a finite tree  $T = W_m$  such that*

$$d_{\gamma+2j+1}^2(T) < d_{\gamma+2j}(T)d_{\gamma+2j+2}(T), \quad \forall 0 \leq j \leq m-1.$$

For a picture of this tree, see Figure 2. This example therefore elaborates upon the example produced in [BS24] of a tree whose dominating set sequence is not log-concave at one index. However, Theorem 1.2 does not provide a path toward disproving Conjecture 1.1 due to (1.1).

## 2. COMPUTATIONAL METHODS

**2.1. Transformer Definition.** PatternBoost is a transformer-based neural network. In this section, we therefore define a transformer.

Let  $\beta > 0$ . For any  $x \in \mathbb{R}^n$ , define the **softmax** function  $p = p^{(\beta)}: \mathbb{R}^n \rightarrow \mathbb{R}^n$  by

$$p_i(x) := \frac{e^{\beta x_i}}{\sum_{j=1}^n e^{\beta x_j}}, \quad \forall x \in \mathbb{R}^n, \quad \forall 1 \leq i \leq n.$$

The quantity  $1/\beta$  is called the **temperature** of the softmax function.

Below, we consider a function  $f: \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ , i.e.  $f$  is a function from  $m \times n$  real matrices to  $m \times n$  real matrices. We denote the  $i^{\text{th}}$  row of an  $m \times n$  matrix  $M$  as  $M^{(i)}$ , for each  $1 \leq i \leq m$ .

**Definition 2.1 (Transformer).** Let  $A_1, \dots, A_k, B_1, \dots, B_k$  be  $n \times n$  real matrices. A **multi-head attention layer** or **transformer** with  $k$  **attention heads** is a function  $f: \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$  defined by

$$[f(Y)]^{(i)} := \sum_{j=1}^k p([Y A_j Y^T]^{(i)}) Y B_j, \quad \forall Y \in \mathbb{R}^{m \times n}.$$

The matrices  $A_1, \dots, A_k$  are called **attention matrices**, and the matrices  $B_1, \dots, B_k$  are thought of as projection matrices, though they will in general not be actual projection matrices. Each of the  $k$  terms in the sum in Definition 2.1 is called an attention head.

**2.2. Background on PatternBoost.** PatternBoost works in the following way. The input is: a reward function, a greedy local search algorithm, and a choice of representation of the object being maximized. The output is a set of candidate maximizing objects. In more detail:

- Initialize a reward function of graphs that we should maximize. In our case, this function could be the following function of a graph  $G$ :

$$d_{k+2}(G)d_k(G) - d_{k+1}(G)^2, \quad (2.1)$$

where  $k$  is the size of the smallest dominating set in  $G$ . (For a general graph  $G$ , the coefficients  $d_j(G)$  cannot be computed efficiently, but they can be computed efficiently for trees. See Section 4.1 for more details.)

It is generally preferable that the function to maximize can be computed efficiently.

- Initialize a greedy local search algorithm. In our case, if we are optimizing over graphs, this means: given any graph, consider all ways to add or delete an edge, and choose one with the largest reward function (including the original graph). If we are instead optimizing over trees, we use the algorithm of [RS25]: connect two random vertices with an edge, consider all ways to subtract an edge from the resulting cycle, and among those choose a tree with the largest reward.
- We represent graphs in our optimization as binary vectors containing the entries of the graph's adjacency matrix above the diagonal. We represent trees in our optimization by Prüfer codes.

After these initialization steps, the algorithm proceeds as follows.

- Generate an initial list (dataset) of e.g. 50,000 graphs (or trees). This list is made via a greedy local search, possibly with some (pseudo)random choices.
- Train a transformer on the list of graphs (or trees). More specifically, optimize the weights of the transformer using the Adam optimization method (or a variant thereof) with a loss function that fits the transformer to the initial dataset. (We briefly define a transformer in Section 2.1.)
- Query the transformer to generate a new list (dataset) of e.g. 100,000 graphs (or trees).
- Run the greedy local search on the new dataset.
- Retain the top 10% of examples; iterate the previous three steps several times.

**2.3. Related Results.** We use the code of [RS25] who achieved similar results for the related problem for independence polynomials of trees. Nevertheless, we made some changes to the codes of [RS25]. Most notably, [RS25] chooses a reward function (2.1) (for the independent set sequence of a tree) where  $k = n/2$  is a fixed function of the number of vertices of the graph. We however found this was not necessary either for independent sets or for dominating sets. That is, it suffices to choose the index  $k$  to depend on the first nonzero index of the sequence.

Also, for trees we needed to adapt some results in the literature (albeit using standard techniques) to efficiently compute their dominating set sequences. We describe this algorithm in Section 4.1.

**2.4. Comparison with Exhaustive Search.** There are approximately  $1.6 \cdot 10^{11}$  nonisomorphic simple graphs on 12 vertices. Moreover, the computation of the number of dominating sets of fixed size is computationally intractable [KPT17]. So, an exhaustive search over such graphs together with computation of the dominating sets is infeasible.

The number of nonisomorphic unlabelled trees on  $n$  vertices is asymptotically larger than  $2.9^n$ . For example, there are approximately  $1.4 \cdot 10^{10}$  such trees on  $n = 30$  vertices. The number of dominating sets of a fixed size can be computed efficiently, but an exhaustive search over trees becomes intractable when  $n$  is much larger than 20.

In other words, PatternBoost does not seem to be performing a convoluted exhaustive search.

**2.5. Limitations.** Although PatternBoost is able to generate interesting examples for Conjecture 1.1 (as it did in a similar problem for independent sets [RS25]), there are many problems for which PatternBoost fails to find interesting examples. In the case of our problem, PatternBoost was unable to find a tree where log-concavity of the dominating set sequence breaks in at least two indices. However, we know such examples exist due to Theorem 1.2. Typically an epoch for Patternboost running on an NVIDIA RTX Ada 5000 would be complete in 5 to 15 minutes. So, running ten to twenty epochs, which was sufficient for one experimental run of PatternBoost for a graph or tree with a fixed number of vertices, could take one to several hours. The full project, including testing runs, used about 24 to 48 hours of compute on this single GPU. Since finding dominating sets on general graphs is inefficient, it is doubtful we could search for graphs with more than 15 to 20 vertices. Similarly, it is doubtful we could search for trees with more than a few hundred vertices, even though their dominating set polynomials can be found efficiently.

**2.6. Conclusions from the Computations.** PatternBoost was able to generate examples of graphs and trees with non log-concave dominating set sequences de novo, replicating and extending the result of [BS24] for trees. This result is interesting since PatternBoost generates its own data while not using the internet, any external training data, or any innate knowledge. However, PatternBoost was unable to find a graph or tree with more than one log-concavity violation, which we were able to do “by hand” by adapting an argument of [B25]. Perhaps it is difficult in our implementation of PatternBoost to find such a highly structured tree as in Theorem 1.2 (and Figure 2). So, our work does not rule out the possibility that PatternBoost could discover such trees, but at the present time this automated tool serves as an assistant rather than a replacement for the mathematician.

### 3. LOG-CONCAVE INSTANCES

**3.1. A Continuous Analogue.** Using the Brunn-Minkowski inequality, we demonstrate that a continuous version of the dominating set sequence of a graph is log-concave.

Let  $G = (V, E)$  be an undirected graph with vertex set  $V = \{1, \dots, n\}$  and edge set  $E$ . Let  $A$  be its adjacency matrix, i.e.  $A_{i,j} = 1$  if  $\{i, j\} \in E$ . Let  $\mathbf{x} := [x_1, \dots, x_n]^T$  and define the polytope  $P(G)$  as

$$P(G) := \{\mathbf{x} \in [0, 1]^n : (A\mathbf{x})_i + x_i \geq 1, i = 1, \dots, n\}.$$

In other words, it consists of the points in the hypercube, such that  $x_i + \sum_{j \in V: \{i,j\} \in E} x_j \geq 1$  for every  $i = 1, \dots, n$ .

**Lemma 3.1.** *We have that  $d_k(G) = |P(G) \cap \{\mathbf{x} \in \{0, 1\}^n : x_1 + \dots + x_n = k\}|$ .*

*Proof.* This follows immediately by treating  $\mathbf{x} \in \{0, 1\}^n$  as an indicator vector of a set  $S \subset V$  with  $x_i = 1$  iff  $i \in S$ . The conditions of  $P(G)$  are equivalent to  $S$  being a dominating set: since  $x_i + \sum_{j \in V: \{i, j\} \in E} x_j \geq 1$  we have that either  $i \in S$  or else there must exist at least one  $j$  with  $x_j = 1$  and  $\{i, j\} \in E$ , so  $j \in S$  is adjacent to  $i$ . The last restriction on the sum implies that  $|S| = k$ .  $\square$

A not-so-close approximation to the number of integer points in a polytope is given by the volume. Set

$$\begin{aligned} D_k(G) &:= \{x \in P(G) : \sum_{v \in V} x_v = k\} \\ \tilde{d}_k(G) &:= \text{Vol}_{n-1}(D_k(G)) \end{aligned} \tag{3.1}$$

for any real number  $k \in [0, n]$ , where  $\text{Vol}_{n-1}$  denotes  $(n-1)$ -dimensional Lebesgue measure.

**Theorem 3.2.** *The numbers  $\tilde{d}_k(G)$  form a log-concave sequence, i.e.*

$$\tilde{d}_k(G)^2 \geq \tilde{d}_{k-1}(G)\tilde{d}_{k+1}(G)$$

for  $k = 1, \dots, n-1$  and every graph  $G$ .

*Proof.* The log-concavity of these numbers follows directly from the Brunn-Minkowski inequality. For any real number  $k$  in  $[1, n-1]$ , we have

$$D_k(G) \supseteq \frac{1}{2} \left( D_{k+1}(G) + D_{k-1}(G) \right),$$

where  $+$  denotes the Minkowski sum, and we consider  $D_{k-1}(G), D_k(G), D_{k+1}(G)$  all as subsets of  $\mathbb{R}^{n-1}$  by translating in the direction  $(1, \dots, 1)$  as needed. So, using this containment, (3.1) and the multiplicative form of Brunn-Minkowski that  $\mu(\lambda A + (1-\lambda)B) \geq \mu(A)^\lambda \mu(B)^{1-\lambda}$  for a Lebesgue measure  $\mu$ , for any  $0 < \lambda < 1$ , we have (using  $\lambda = 1/2$ )

$$\begin{aligned} \tilde{d}_k(G)^2 &\geq \left( \text{Vol}_{n-1} \left( \frac{1}{2} (D_{k+1}(G) + D_{k-1}(G)) \right) \right)^2 \\ &\geq \text{Vol}_{n-1} D_{k+1}(G) \cdot \text{Vol}_{n-1} D_{k-1}(G) \stackrel{(3.1)}{=} \tilde{d}_{k-1}(G)\tilde{d}_{k+1}(G). \end{aligned}$$

That is, the sequence  $\tilde{d}_0(G), \dots, \tilde{d}_n(G)$  is log-concave.  $\square$

It is tempting to deduce a similar property for  $d_0(G), \dots, d_n(G)$  from this log-concavity property using e.g. the main result of [HKM10], but it would be wrong.

**3.2. Log-concave families of trees.** Paths and caterpillars are a fundamental class of trees. They were seriously studied in the context of the independence set problem, see e.g. [BES18, GH18, WZ11, Zhu07].

A caterpillar tree  $T(a_1, \dots, a_k)$  consists of a spine of vertices  $v_1, \dots, v_k$ , with edges  $v_i - v_{i+1}$ , and to vertex  $v_i$  there are  $a_i$  many leaves attached.

**Theorem 3.3.** *Let  $T := T(a_1, \dots, a_k)$  be a caterpillar tree with  $a_i \geq 1$  for all  $i$ . Then the domination polynomial is equal to*

$$D(T; x) := \sum_{j \geq 0} d_j(T) x^j = \prod_{i=1}^k (x^{a_i} + x(1+x)^{a_i})$$

and its coefficients form a log-concave sequence, i.e.  $d_j(T)^2 \geq d_{j-1}(T)d_{j+1}(T)$ ,  $\forall j \geq 1$ .

*Proof.* Let  $S$  be a dominating set of  $T$ . If  $v_i \in S$ , then each of the leaves attached to  $v_i$  may or may not be in  $S$  independently of the rest of the graph. This contributes  $x(1+x)^{a_i}$  to the domination polynomial. If  $v_i \notin S$ , then each of the leaves attached to  $v_i$  has to be in  $S$ , otherwise it would not be “covered” by a vertex from  $S$ . This gives a term  $x^{a_i}$  in the domination polynomial. In that case the vertex  $v_i$  is “covered” by its leaves (here we use  $a_i \geq 1$ ) and the local configuration does not depend on which vertices from the rest of the graph are in  $S$ . Thus the choice of which vertices among  $v_i$  and its attached leaves lie in  $S$  is independent for each  $i$ , so

$$D(T; x) = \prod_{i=1}^k (x(1+x)^{a_i} + x^{a_i}).$$

Notice that

$$x(1+x)^{a_i} + x^{a_i} = x(1 + a_i x + \binom{a_i}{2} x^2 + \cdots + (a_i + 1)x^{a_i-1} + x^{a_i})$$

is a log-concave polynomial since most of its coefficients are binomials which satisfy

$$\binom{a_i}{j}^2 \geq \binom{a_i}{j-1} \binom{a_i}{j+1}$$

thereby verifying log-concavity for the coefficients where  $j \neq a_i - 2$  and

$$\binom{a_i}{2}^2 \geq \binom{a_i}{3} (a_i + 1).$$

Finally, the product of log-concave polynomials is log-concave and the claim follows.  $\square$

**Remark 3.4.** *We believe that also legless (when some  $a_i = 0$ ) caterpillars have log-concave domination polynomials. This result shows a different story to the independence polynomial log-concavity investigations. Only regular (all  $a_i$  being equal) or periodic regular ( $a_{2i} = a, a_{2i+1} = b$ ) caterpillars are known to have unimodal independence polynomials [GH18, BES18], whereas the general case remains open.*

#### 4. TREES WITH ARBITRARY BREAKS

In this section we prove that an arbitrary number of log-concavity failures can occur for a family built from the Bautista–Ramos trees [B25] by adding a leaf to each existing leaf of the trees, as in [BS24].

We work with the following rooted gadgets (see Figure 2.)

- Let  $L$  be the rooted path on three vertices  $(u, v, w)$ , rooted at  $u$ .
- Let  $F_t$  be the rooted tree obtained by taking a root  $c$  with  $t$  children, each a copy of  $L$ .
- Let  $H_t$  be the rooted tree obtained by taking a root  $v$  with three children, each a copy of  $F_t$ .
- Let  $X$  be the rooted path on two vertices  $(a, b)$ , rooted at  $a$ .
- Let  $W_{m,t}$  be the rooted tree obtained by taking a new root  $r$ , attaching  $m$  children each a copy of  $H_t$ , and attaching one additional child which is a copy of  $X$ .

This is exactly the tree obtained from the Bautista–Ramos tree  $TG_{m,t}$  [B25] by attaching one new leaf to every leaf.

**4.1. Rooted domination states.** Let  $S$  be a rooted tree with root  $\rho$  which we view as a subtree in a larger structure. Let  $x$  be an indeterminate. The dominating set polynomial can be split recursively into the following three parts.

- $\mathcal{A}_S(x)$ : generating polynomial for dominating sets of  $S$  in which  $\rho$  is chosen.

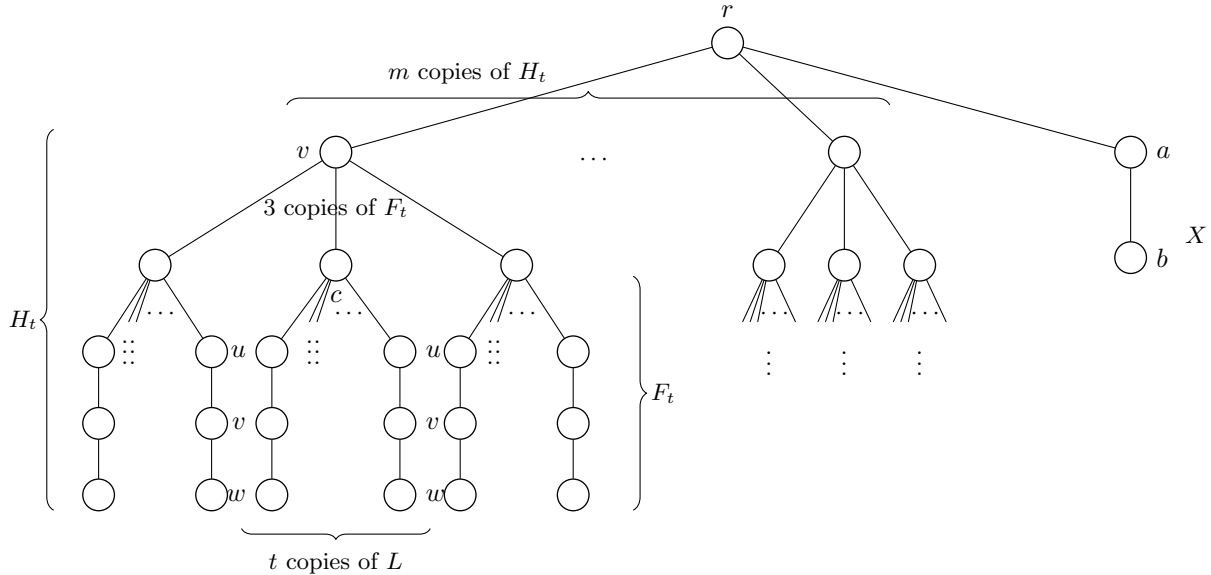


FIGURE 2. The tree  $W_{m,t}$ . The top root  $r$  is adjacent to  $m$  copies of  $H_t$  and to one additional copy of  $X$ . In the leftmost branch, one copy of  $H_t$  is expanded; inside it, the middle copy of  $F_t$  is expanded; and inside that copy of  $F_t$ , the rooted path  $L$  is repeated  $t$  times.

- $\mathcal{B}_S(x)$ : generating polynomial for dominating sets of  $S$  in which  $\rho$  is not chosen and is not dominated by its parent (so it is dominated by one of its children).
- $\mathcal{C}_S(x)$ : generating polynomial for dominating sets of  $S$  in which  $\rho$  is not chosen and is not dominated by any child (so it is dominated by its parent).

In every case, all other vertices of  $S$  must already be dominated inside  $S$ .

If the root of  $S$  has child-subtrees  $S_1, \dots, S_d$ , then the standard recurrences are

$$\mathcal{A}_S(x) = x \prod_{i=1}^d (\mathcal{A}_{S_i}(x) + \mathcal{B}_{S_i}(x) + \mathcal{C}_{S_i}(x)), \quad (4.1)$$

$$\mathcal{B}_S(x) = \prod_{i=1}^d (\mathcal{A}_{S_i}(x) + \mathcal{B}_{S_i}(x)) - \prod_{i=1}^d \mathcal{B}_{S_i}(x), \quad (4.2)$$

$$\mathcal{C}_S(x) = \prod_{i=1}^d \mathcal{B}_{S_i}(x). \quad (4.3)$$

Indeed, if the root is chosen then it dominates every child-root, so each child-root may be in any of the three states; if the root is in state  $\mathcal{C}$ , then every child-root must be in state  $\mathcal{B}$ ; and if the root is in state  $\mathcal{B}$ , then every child-root must lie in state  $\mathcal{A}$  or  $\mathcal{B}$ , with at least one child in state  $\mathcal{A}$ .

**Remark 4.1.** *The recursions (4.1), (4.2) and (4.3) imply a dynamic programming algorithm to efficiently compute the dominating set sequence of a tree. Given a tree, choose a root, then recursively compute  $\mathcal{A}, \mathcal{B}$  and  $\mathcal{C}$  for each rooted subtree, using depth-first search.*

#### 4.2. The basic branch and the t-fan gadget.

**Lemma 4.2.** *Let  $L$  be a path graph with vertices  $(u, v, w)$  rooted at  $u$ , considered as a subgraph of a larger tree, i.e  $u$  is connected to a vertex not equal to  $v$  in the larger tree. Then*

$$\mathcal{A}_L(x) = x^2(2+x), \quad \mathcal{B}_L(x) = x(1+x), \quad \mathcal{C}_L(x) = x.$$

*Proof.* Direct enumeration gives the answer: In state  $\mathcal{A}$ , the root  $u$  is chosen. To dominate  $w$ , we must also choose either  $v$  or  $w$ . Hence  $\mathcal{A}_L = 2x^2 + x^3 = x^2(2+x)$ . In state  $\mathcal{B}$ , the root  $u$  is not chosen but must be dominated from below, so  $v$  must be chosen; then  $w$  may or may not be chosen. Hence  $\mathcal{B}_L = x + x^2 = x(1+x)$ . In state  $\mathcal{C}$ , the root  $u$  is left to be dominated by the parent, so  $v$  cannot be chosen. Then  $w$  must be chosen in order to dominate  $v$ . Hence  $\mathcal{C}_L = x$ .  $\square$

Now let  $F_t$  be the rooted tree whose root  $c$  has  $t$  children, each a copy of  $L$ . Define

$$P_t := (1 + 3x + x^2)^t, \quad Q_t := (2 + 3x + x^2)^t, \quad R_t := \frac{P_t - (1+x)^t}{x}. \quad (4.4)$$

**Lemma 4.3.** *For the rooted tree  $F_t$ ,*

$$\mathcal{A}_{F_t}(x) = x^{t+1}Q_t, \quad \mathcal{B}_{F_t}(x) = x^{t+1}R_t, \quad \mathcal{C}_{F_t}(x) = x^t(1+x)^t.$$

*Proof.* By Lemma 4.2,

$$\mathcal{A}_L + \mathcal{B}_L + \mathcal{C}_L = x^2(2+x) + x(1+x) + x = x(2+3x+x^2),$$

so by (4.1),

$$\mathcal{A}_{F_t} = x(x(2+3x+x^2))^t \stackrel{(4.4)}{=} x^{t+1}Q_t.$$

Also,

$$\mathcal{A}_L + \mathcal{B}_L = x^2(2+x) + x(1+x) = x(1+3x+x^2),$$

so by (4.2) and (4.3),

$$\mathcal{B}_{F_t} = (x(1+3x+x^2))^t - (x(1+x))^t = x^t((1+3x+x^2)^t - (1+x)^t) \stackrel{(4.4)}{=} x^{t+1}R_t,$$

while by (4.3)

$$\mathcal{C}_{F_t} = (x(1+x))^t = x^t(1+x)^t. \quad \square$$

**4.3. The three-branch gadget.** Let  $H_t$  be the rooted tree whose root has three children, each a copy of  $F_t$ .

**Lemma 4.4.** *For  $H_t$ ,*

$$\mathcal{A}_{H_t}(x) = x^{3t+1}U_t, \quad \mathcal{B}_{H_t}(x) = x^{3t+3}V_t, \quad \mathcal{C}_{H_t}(x) = x^{3t+3}R_t^3,$$

where

$$U_t(x) := (P_t + xQ_t)^3, \quad V_t(x) := (Q_t + R_t)^3 - R_t^3. \quad (4.5)$$

*Proof.* From Lemma 4.3,

$$\mathcal{A}_{F_t} + \mathcal{B}_{F_t} + \mathcal{C}_{F_t} = x^{t+1}Q_t + x^{t+1}R_t + x^t(1+x)^t = x^t(xQ_t + xR_t + (1+x)^t). \quad (4.6)$$

By the definition of  $R_t$  in (4.4), we have

$$xR_t + (1+x)^t = (1+3x+x^2)^t \stackrel{(4.4)}{=} P_t,$$

so

$$\mathcal{A}_{F_t} + \mathcal{B}_{F_t} + \mathcal{C}_{F_t} \stackrel{(4.6)}{=} x^t(P_t + xQ_t).$$

Hence (4.1) yields

$$\mathcal{A}_{H_t} = x(x^t(P_t + xQ_t))^3 = x^{3t+1}(P_t + xQ_t)^3 \stackrel{(4.5)}{=} x^{3t+1}U_t.$$

Likewise,

$$\mathcal{A}_{F_t} + \mathcal{B}_{F_t} = x^{t+1}(Q_t + R_t),$$

so (4.2) gives

$$\mathcal{B}_{H_t} = (x^{t+1}(Q_t + R_t))^3 - (x^{t+1}R_t)^3 = x^{3t+3}((Q_t + R_t)^3 - R_t^3) \stackrel{(4.5)}{=} x^{3t+3}V_t.$$

Finally, by (4.3)

$$\mathcal{C}_{H_t} = (x^{t+1}R_t)^3 = x^{3t+3}R_t^3.$$

□

**4.4. Coefficient growth estimates.** We now isolate the exponential growth in  $t$  of the low-degree coefficients of  $U_t$  and  $V_t$ .

**Lemma 4.5.** *Fix  $r \geq 0$ .*

- (1)  $[x^r]P_t$  and  $[x^r]R_t$  are polynomials in  $t$ .
- (2) There is a polynomial  $q_r(t)$  with positive leading coefficient such that

$$[x^r]Q_t = 2^t q_r(t).$$

In particular,  $[x^r]Q_t = 2^t t^{O(1)}$ .

As a consequence,

$$[x^r]V_t = 2^{3t} t^{O(1)}, \quad [x^r]U_t = 2^{\min(r,3)t} t^{O(1)}.$$

*Proof.* For  $P_t = (1 + 3x + x^2)^t$ , the coefficient of  $x^r$  is obtained by choosing in each factor one of  $1$ ,  $3x$ , or  $x^2$ , with total degree  $r$ . For fixed  $r$ , the number of contributing choices is finite and independent of  $t$ , and each contribution is a product of binomial coefficients in  $t$ ; hence  $[x^r]P_t$  is a polynomial in  $t$ .

Since

$$R_t \stackrel{(4.4)}{=} \frac{P_t - (1+x)^t}{x},$$

we have

$$[x^r]R_t = [x^{r+1}]P_t - [x^{r+1}](1+x)^t,$$

and each term on the right is polynomial in  $t$ , proving part (1).

For part (2), write

$$Q_t = (2 + 3x + x^2)^t \stackrel{(4.4)}{=} 2^t \left(1 + \frac{3}{2}x + \frac{1}{2}x^2\right)^t.$$

The coefficient of  $x^r$  in the second factor is again a polynomial in  $t$  for fixed  $r$ , and its leading term is positive because the term  $(\frac{3}{2}x)^r$  contributes positively. Thus

$$[x^r]Q_t = 2^t q_r(t)$$

for a polynomial  $q_r(t)$  with positive leading coefficient.

Now,

$$V_t \stackrel{(4.5)}{=} (Q_t + R_t)^3 - R_t^3 = Q_t^3 + 3R_t Q_t^2 + 3R_t^2 Q_t.$$

By part (2), every fixed low-degree coefficient of  $Q_t$  is  $2^t t^{O(1)}$ , while by part (1) every fixed low-degree coefficient of  $R_t$  is  $t^{O(1)}$ . Hence the coefficient of  $x^r$  in  $Q_t^3$  has size  $2^{3t} t^{O(1)}$ , and the coefficients in the remaining two summands are at most  $2^{2t} t^{O(1)}$  and  $2^t t^{O(1)}$ . Therefore

$$[x^r]V_t = 2^{3t} t^{O(1)}.$$

For  $U_t \stackrel{(4.5)}{=} (P_t + xQ_t)^3$ , expand

$$U_t = P_t^3 + 3P_t^2(xQ_t) + 3P_t(xQ_t)^2 + (xQ_t)^3.$$

Each factor  $xQ_t$  contributes one unit of degree and one factor  $2^t$  to the exponential order. Therefore, in degree  $r$ , at most  $\min(r, 3)$  such factors can appear, and the largest exponential order is  $2^{\min(r,3)t}$ . This proves

$$[x^r]U_t = 2^{\min(r,3)t}t^{O(1)}.$$

□

The next lemma controls low-degree coefficients of powers of  $U_t$ .

**Lemma 4.6.** *Fix integers  $q \geq 0$  and  $0 \leq u \leq 2q$ . Then*

$$[x^u]U_t^q = 2^{ut}t^{O(1)}.$$

*Proof.* For the upper bound, use convolution together with Lemma 4.5. Writing

$$[x^u]U_t^q = \sum_{\substack{u_1, \dots, u_q \geq 0: \\ u_1 + \dots + u_q = u}} \prod_{i=1}^q [x^{u_i}]U_t, \quad (4.7)$$

each factor satisfies

$$[x^{u_i}]U_t = 2^{\min(u_i, 3)t}t^{O(1)}.$$

Since  $\sum_{i=1}^q u_i = u \leq 2q$ , every  $u_i$  appearing in a nonnegative composition of  $u$  satisfies  $\sum_{i=1}^q \min(u_i, 3) \leq u$ , and therefore each summand in (4.7) is at most  $2^{ut}t^{O(1)}$ . Summing over finitely many compositions gives

$$[x^u]U_t^q \leq 2^{ut}t^{O(1)}.$$

For the lower bound, write

$$u = \varepsilon_1 + \dots + \varepsilon_q$$

with each  $\varepsilon_i \in \{0, 1, 2\}$ ; this is possible because  $u \leq 2q$ . The corresponding product term in the convolution contributes

$$\prod_{i=1}^q [x^{\varepsilon_i}]U_t.$$

By Lemma 4.5, for  $\varepsilon_i \in \{0, 1, 2\}$  we have

$$[x^{\varepsilon_i}]U_t = 2^{\varepsilon_i t}t^{O(1)},$$

with positive leading constant. Hence this product is

$$2^{(\varepsilon_1 + \dots + \varepsilon_q)t}t^{O(1)} = 2^{ut}t^{O(1)}.$$

Since all coefficients are nonnegative, this yields the matching lower bound. Therefore

$$[x^u]U_t^q = 2^{ut}t^{O(1)}.$$

□

**4.5. The whole tree.** We now incorporate the extra whiskered leaf at the top.

**Lemma 4.7.** *Let  $X$  be a path graph with vertices  $(a, b)$  rooted at  $a$ . Then*

$$\mathcal{A}_X(x) = x(1+x), \quad \mathcal{B}_X(x) = x, \quad \mathcal{C}_X(x) = 0.$$

*Proof.* In state  $\mathcal{A}$ , either  $\{a\}$  or  $\{a, b\}$  is chosen, so  $\mathcal{A}_X = x + x^2 = x(1+x)$ . In state  $\mathcal{B}$ , the root  $a$  is not chosen and must be dominated by its child, so  $b$  must be chosen; thus  $\mathcal{B}_X = x$ . State  $\mathcal{C}$  is impossible, because if  $a$  is left to be dominated by its parent then there is no way to dominate  $b$  without choosing  $a$  or  $b$ , and choosing  $b$  would force  $a$  into state  $\mathcal{B}$  instead. Hence  $\mathcal{C}_X = 0$ . □

**Lemma 4.8.** *For the tree  $W_{m,t}$ , its smallest dominating set has cardinality  $\gamma(W_{m,t})$  where*

$$\gamma(W_{m,t}) = m(3t+1) + 1.$$

*Proof.* Each branch  $H_t$  contributes at least  $3t+1$  vertices in any dominating set meeting the branch, because  $\mathcal{A}_{H_t}$  starts in degree  $3t+1$  and both  $\mathcal{B}_{H_t}$  and  $\mathcal{C}_{H_t}$  start in degree  $3t+3$  by Lemma 4.4. The extra branch  $X$  contributes at least one vertex by Lemma 4.7. Hence every dominating set of  $W_{m,t}$  has size at least  $m(3t+1)+1$ .

This lower bound is attained by leaving the top root  $r$  unchosen, putting the  $X$ -branch in state  $\mathcal{B}$ , and putting each  $H_t$ -branch in state  $\mathcal{A}$ . Therefore

$$\gamma(W_{m,t}) = m(3t+1) + 1.$$

□

Write

$$D(W_{m,t}; x) = x^{\gamma(W_{m,t})} \sum_{r \geq 0} e_r(m, t) x^r.$$

Thus  $e_r(m, t) = d_{\gamma(W_{m,t})+r}(W_{m,t})$ .

**Lemma 4.9.** *Fix  $m \geq 1$ . For every integer  $r$  with  $0 \leq r \leq 2m$ ,*

$$e_r(m, t) = 2^{v_r t} t^{O(1)}, \quad \text{where} \quad v_r = r + \left\lfloor \frac{r}{2} \right\rfloor,$$

where the implicit constants can depend on  $m, r$  but not  $t$ .

*Proof.* We analyze the contributions from the top root in states  $\mathcal{B}$  and  $\mathcal{A}$  separately.

**Root in state  $\mathcal{B}$ .** Using (4.2) with one child-subtree  $X$  and  $m$  child-subtrees  $H_t$ , and recalling  $\mathcal{C}_X = 0$ , we obtain

$$\mathcal{B}_{W_{m,t}} = (\mathcal{A}_X + \mathcal{B}_X)(\mathcal{A}_{H_t} + \mathcal{B}_{H_t})^m - \mathcal{B}_X \mathcal{B}_{H_t}^m.$$

By Lemmas 4.4 and 4.7,

$$\begin{aligned} \mathcal{A}_X + \mathcal{B}_X &= 2x + x^2 = x(2+x), \\ \mathcal{A}_{H_t} + \mathcal{B}_{H_t} &= x^{3t+1}U_t + x^{3t+3}V_t = x^{3t+1}(U_t + x^2V_t), \\ \mathcal{B}_{H_t} &= x^{3t+3}V_t. \end{aligned}$$

Since  $\gamma(W_{m,t}) = m(3t+1) + 1$  by Lemma 4.8, this gives

$$\mathcal{B}_{W_{m,t}} = x^{\gamma(W_{m,t})} \left( (2+x)(U_t + x^2V_t)^m - x^{2m}V_t^m \right). \quad (4.8)$$

Expand

$$(U_t + x^2V_t)^m = \sum_{j=0}^m \binom{m}{j} x^{2j} U_t^{m-j} V_t^j. \quad (4.9)$$

Fix  $r$  with  $0 \leq r \leq 2m$ . In the term indexed by  $j$ , the coefficient of  $x^r$  is governed by

$$[x^{r-2j}] U_t^{m-j} V_t^j.$$

By Lemmas 4.5 and 4.6, this coefficient has size

$$2^{(r-2j)t} t^{O(1)} \cdot 2^{3jt} t^{O(1)} = 2^{(r+j)t} t^{O(1)},$$

provided  $r-2j \geq 0$ ; otherwise the contribution is zero. Thus among all  $j$  with  $2j \leq r$ , the exponential order is maximized when  $j = \lfloor r/2 \rfloor$ , and the maximal exponent is

$$r + \left\lfloor \frac{r}{2} \right\rfloor = v_r.$$

Therefore the coefficient of  $x^r$  in the first term of (4.8) has size  $2^{v_r t} t^{O(1)}$ .

The subtraction term  $x^{2m}V_t^m$  in (4.8) only affects degree  $r = 2m$  among the range  $0 \leq r \leq 2m$ . In that degree, the factor  $(2+x)$  in the first term contributes two copies of the dominant  $x^{2m}V_t^m$

contribution, so subtracting one copy leaves a positive contribution of the same exponential order. Hence the coefficient of  $x^r$  in the normalized root- $\mathcal{B}$  contribution has size

$$2^{v_r t} t^{O(1)} \quad (0 \leq r \leq 2m).$$

**Root in state  $\mathcal{A}$ .** From (4.1),

$$\mathcal{A}_{W_{m,t}} = x(\mathcal{A}_X + \mathcal{B}_X + \mathcal{C}_X)(\mathcal{A}_{H_t} + \mathcal{B}_{H_t} + \mathcal{C}_{H_t})^m.$$

By Lemmas 4.4 and 4.7,

$$\mathcal{A}_X + \mathcal{B}_X + \mathcal{C}_X = x(2 + x),$$

and

$$\mathcal{A}_{H_t} + \mathcal{B}_{H_t} + \mathcal{C}_{H_t} = x^{3t+1}U_t + x^{3t+3}(V_t + R_t^3) = x^{3t+1}(U_t + x^2(V_t + R_t^3)).$$

Thus

$$\mathcal{A}_{W_{m,t}} = x^{\gamma(W_{m,t})+1}(2+x)(U_t + x^2(V_t + R_t^3))^m. \quad (4.10)$$

Compared with (4.8), there is an extra factor of  $x$ , so the contribution to  $e_r(m, t)$  coming from (4.10) has exponential order at most  $2^{v_{r-1}t} t^{O(1)}$  for  $r \geq 1$ , which is strictly smaller than  $2^{v_r t} t^{O(1)}$  because  $v_r > v_{r-1}$ . For  $r = 0$ , the root- $\mathcal{A}$  contribution vanishes anyway because of the extra factor of  $x$ .

Combining the two cases shows that the total normalized coefficient satisfies

$$e_r(m, t) = 2^{v_r t} t^{O(1)} \quad (0 \leq r \leq 2m),$$

as claimed. □

**4.6. Main Theorem: Arbitrarily Many Log-Concavity Violations.** We now restate Theorem 1.2 as Theorem 4.10 below, and prove it.

**Theorem 4.10.** *Fix  $m \geq 1$ . Then for all sufficiently large  $t$ , the domination polynomial of  $W_{m,t}$  has at least  $m$  log-concavity failures. More precisely, if*

$$D(W_{m,t}; x) = \sum_{s \geq \gamma(W_{m,t})} d_s x^s,$$

then for every  $j = 0, 1, \dots, m-1$  one has

$$d_{\gamma(W_{m,t})+2j+1}^2 < d_{\gamma(W_{m,t})+2j} d_{\gamma(W_{m,t})+2j+2}$$

for all sufficiently large  $t$ .

*Proof.* By Lemma 4.9, for each  $0 \leq j \leq m-1$ ,

$$\begin{aligned} e_{2j}(m, t) &= 2^{(2j+j)t} t^{O(1)} = 2^{3jt} t^{O(1)}, \\ e_{2j+1}(m, t) &= 2^{(2j+1+j)t} t^{O(1)} = 2^{(3j+1)t} t^{O(1)}, \\ e_{2j+2}(m, t) &= 2^{(2j+2+j+1)t} t^{O(1)} = 2^{(3j+3)t} t^{O(1)}. \end{aligned}$$

Therefore

$$e_{2j}(m, t) e_{2j+2}(m, t) = 2^{(6j+3)t} t^{O(1)},$$

while

$$e_{2j+1}(m, t)^2 = 2^{(6j+2)t} t^{O(1)}.$$

Consequently, as  $t \rightarrow \infty$ ,

$$\frac{e_{2j}(m, t) e_{2j+2}(m, t)}{e_{2j+1}(m, t)^2} = 2^t t^{O(1)} \rightarrow \infty.$$

Hence for all sufficiently large  $t$ ,

$$e_{2j+1}(m, t)^2 < e_{2j}(m, t) e_{2j+2}(m, t).$$

Since  $e_r(m, t) = d_{\gamma(W_{m,t})+r}(W_{m,t})$ , this is exactly

$$d_{\gamma(W_{m,t})+2j+1}^2 < d_{\gamma(W_{m,t})+2j} d_{\gamma(W_{m,t})+2j+2},$$

for each  $j = 0, 1, \dots, m-1$ . Thus the domination polynomial has at least  $m$  log-concavity failures, occurring at the indices

$$\gamma(W_{m,t}) + 1, \gamma(W_{m,t}) + 3, \dots, \gamma(W_{m,t}) + 2m - 1.$$

□

**Remark 4.11.** *The theorem proves the existence of at least  $m$  failure indices. It does not assert that these are the only failures.*

## 5. EXAMPLES OF GRAPHS

Run: 53590741 File: search\_output\_1.txt Line: 1  
 Score = +0.173953 | n = 12 | edges = 27

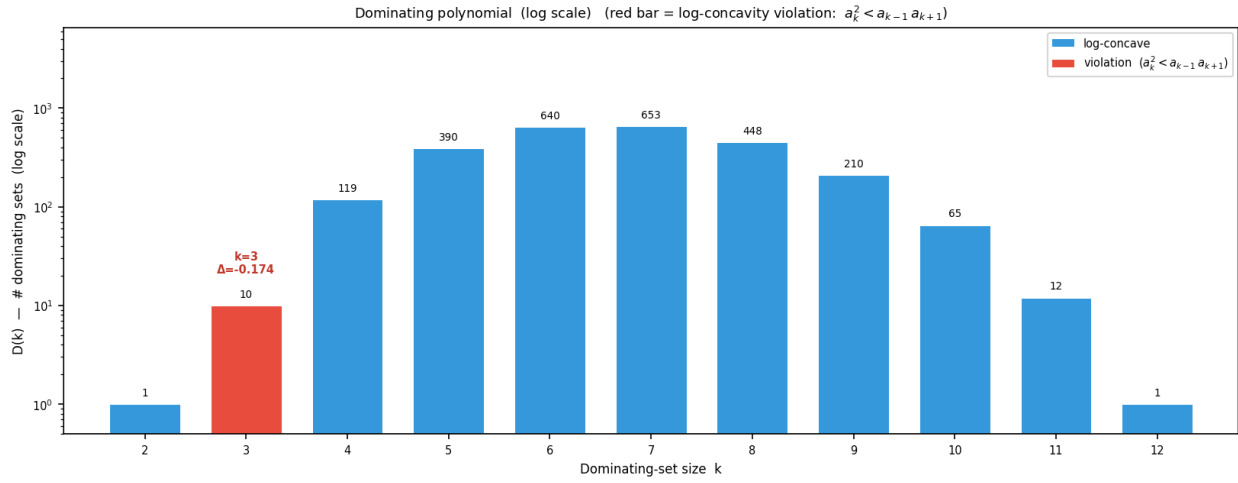
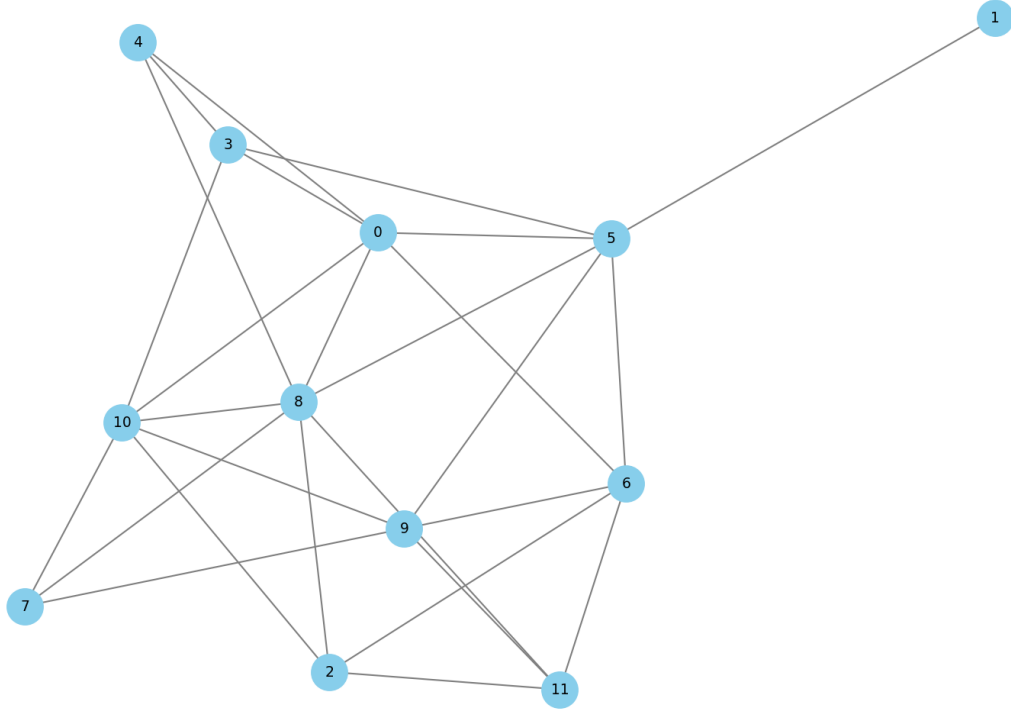


FIGURE 3. A 12-vertex graph with a non log-concave dominating set sequence

Run: 53590741 File: search\_output\_2.txt Line: 35  
 Score = +0.207014 | n = 12 | edges = 24

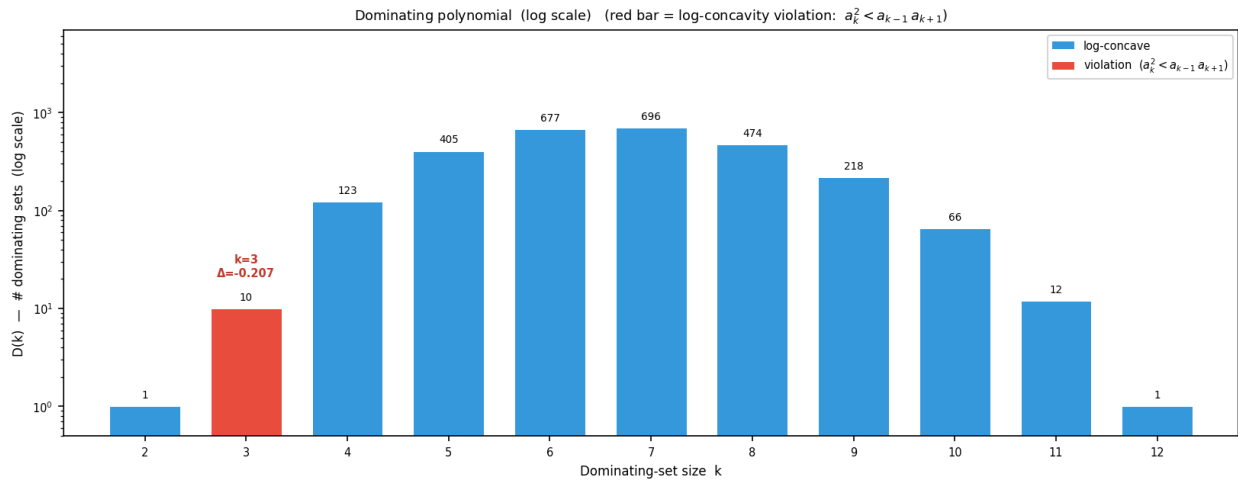
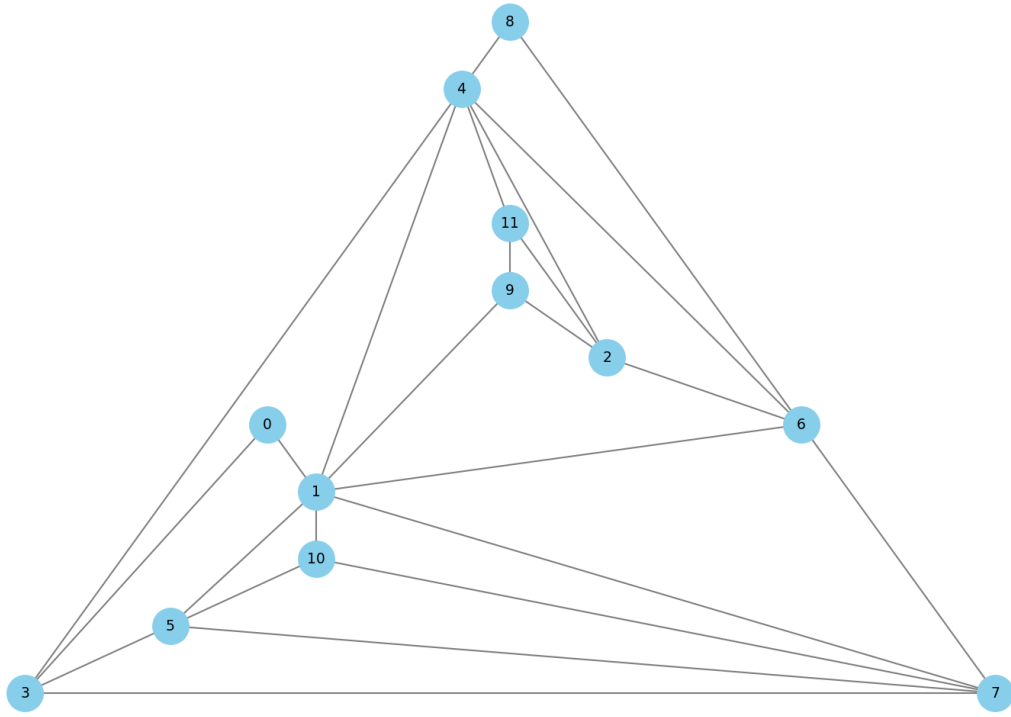


FIGURE 4. A 12-vertex graph with a non log-concave dominating set sequence

Run: 53590741 File: search\_output\_2.txt Line: 102  
 Score = +0.104360 | n = 12 | edges = 22

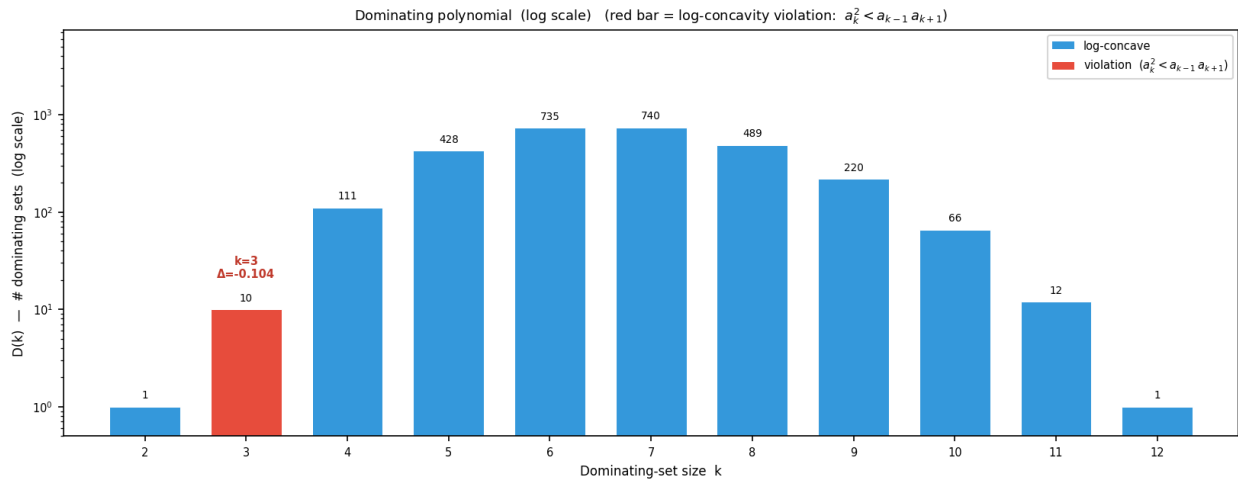
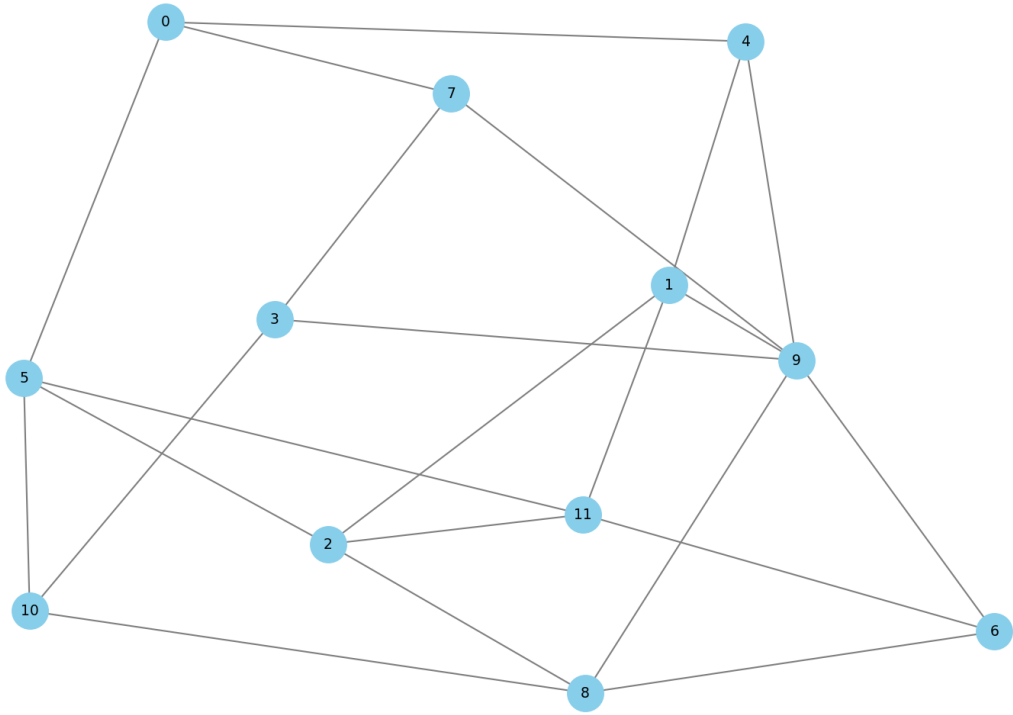
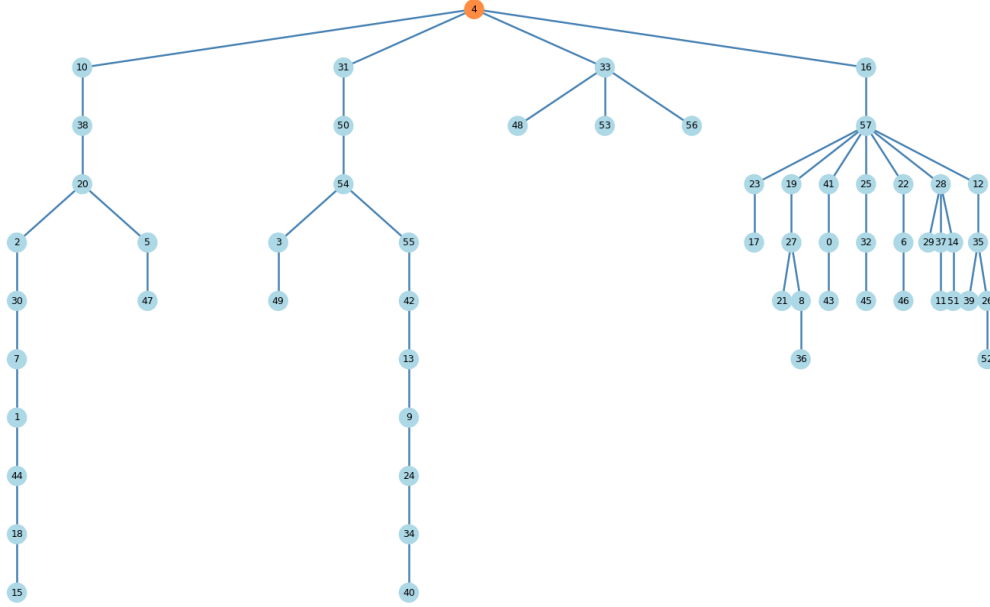


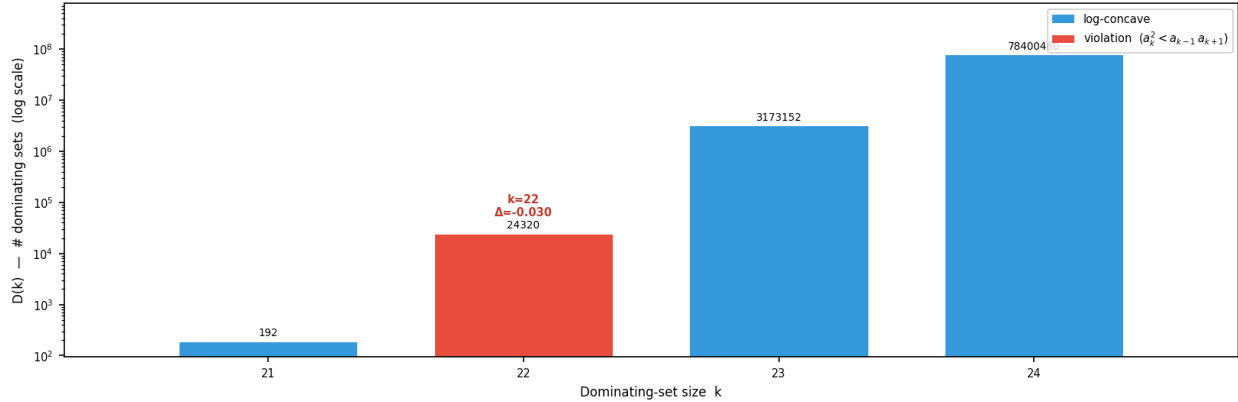
FIGURE 5. A 12-vertex graph with a non log-concave dominating set sequence

## 6. EXAMPLES OF TREES

Run: 4405294 | search\_output\_1.txt line 1  
 Score: +0.029623 | Nodes: 58 | Prufer length: 56 | center: 4



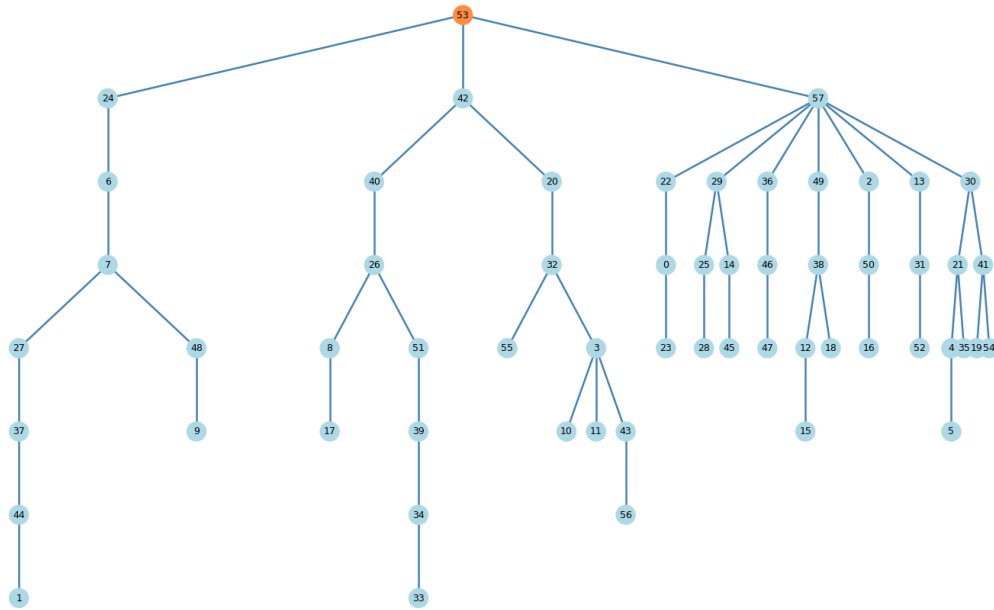
Dominating polynomial (log scale) (red bar = log-concavity violation:  $a_k^2 < a_{k-1} a_{k+1}$ )  
 (showing first 4 of 38 coefficients)



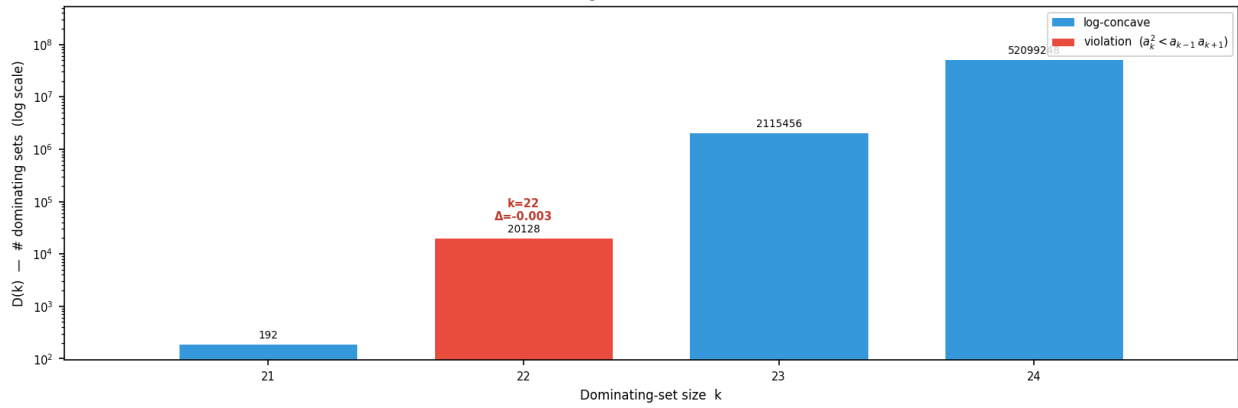
Prufer code (1-indexed):  
 [38, 19, 24, 45, 28, 58, 29, 9, 28, 29, 58, 29, 36, 35, 25, 10, 14, 43, 56, 1,  
 42, 58, 2, 8, 31, 3, 21, 33, 26, 58, 7, 23, 58, 6, 21, 39, 11, 5, 34, 4,  
 55, 15, 29, 58, 27, 36, 13, 58, 34, 55, 51, 32, 5, 34, 5, 17]

FIGURE 6. A 58-vertex tree with a non log-concave dominating set sequence

Run: 4405294 | search\_output\_1.txt line 3  
 Score: +0.002542 | Nodes: 58 | Prufer length: 56 | center: 53



Dominating polynomial (log scale) (red bar = log-concavity violation:  $a_k^2 < a_{k-1} a_{k+1}$ ) (showing first 4 of 38 coefficients)



Prufer code (1-indexed):  
 [45, 5, 22, 49, 4, 4, 13, 39, 51, 9, 27, 39, 42, 1, 23, 58, 26, 30, 35, 40,  
 22, 31, 50, 52, 38, 28, 8, 15, 30, 58, 47, 37, 58, 8, 7, 25, 54, 58, 3, 58,  
 27, 41, 43, 32, 14, 58, 42, 31, 58, 33, 44, 4, 33, 21, 43, 54]

FIGURE 7. A 58-vertex tree with a non log-concave dominating set sequence

Run: 44052941 | search\_output\_2.txt line 3  
 Score: +0.153198 | Nodes: 32 | Prufer length: 30 | center: 26

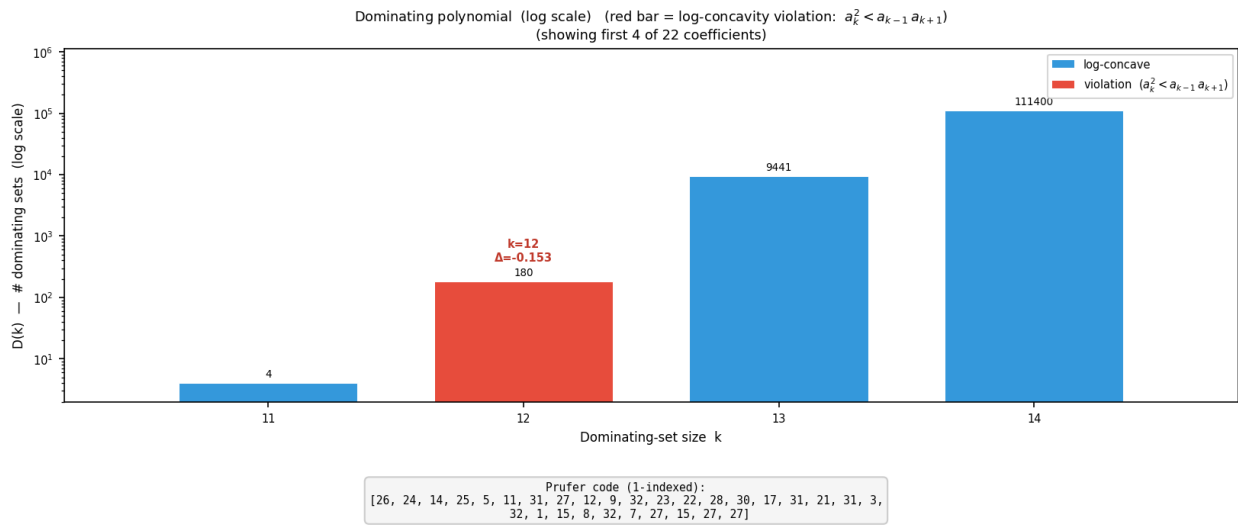
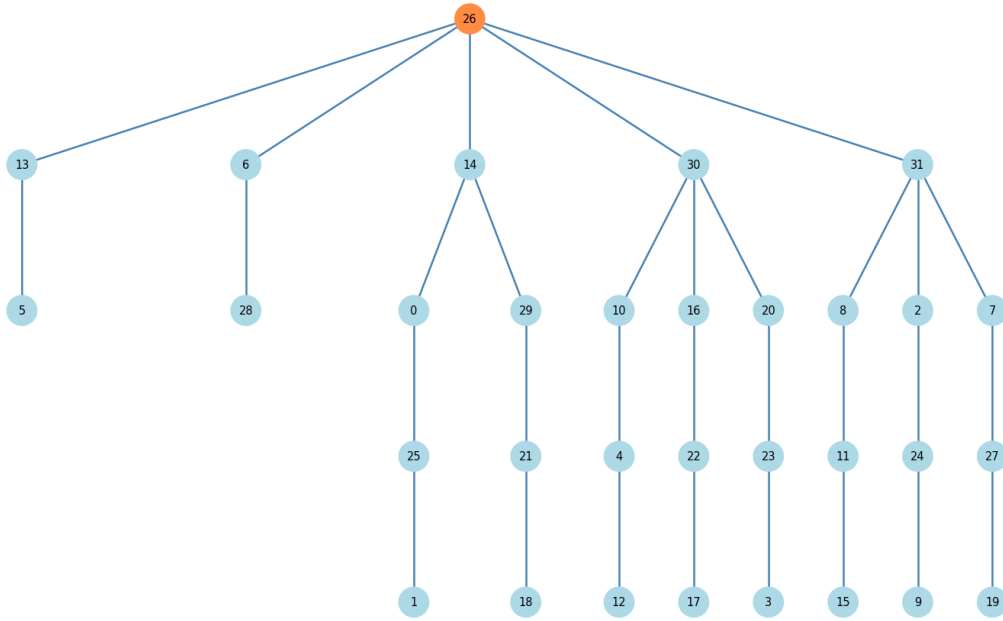


FIGURE 8. A 32-vertex tree with a non log-concave dominating set sequence

**Acknowledgement.** Thanks to Eric Ramos for detailed discussions on the PatternBoost codes. Thanks also to Ferenc Bencs, David Galvin and Elchanan Mossel for helpful discussions. Thanks also to Geordie Williamson for suggesting to use Patternboost on our problem.

#### REFERENCES

- [AHK18] Karim Adiprasito, June Huh and Eric Katz. *Hodge theory for combinatorial geometries*. *Annals of Mathematics*. **2** (2018), vol. 188, pp. 381–452.
- [AMSE87] Yousef Alavi, Paresch J. Malde, Allen J. Schwenk, and Paul Erdős. *The vertex independence sequence of a graph is not constrained*, Eighteenth Southeastern International Conference on Combinatorics, Graph Theory, and Computing (Boca Raton, Fla.), (1987) vol. 58, pp. 15–23.
- [AP14] S. Alikhani and Y. H. Peng. *Introduction to domination polynomial of a graph*. *Ars Combinatoria*, **5** (2014), vol. 114, pp. 257–266.
- [BES18] P. Bahls, B. Ethridge and L. Szabo. *Unimodality of the independence polynomials of non-regular caterpillars*, *Australas. J. Combin.* **1** (2018), vol. 71, pp. 104–112.
- [B25] César Bautista-Ramos. *Multiple breaks of log-concavity in the independence polynomials of trees*. Preprint (2025), [arXiv:2511.00334](https://arxiv.org/abs/2511.00334).
- [B21] I. Beaton. *On dominating sets and the domination polynomial*. Ph.D. Thesis. 2021.
- [BB22] Beaton, I., Brown, J.I. *On the Unimodality of Domination Polynomials*. *Graphs and Combinatorics* **3** (2022), vol. 38, pp. 1–11.
- [BG21] Abdul Basit and David Galvin. *On the Independent Set Sequence of a Tree*. *The Electronic Journal of Combinatorics*. **3** (2021), vol. 28, pp. 1–23.
- [BS24] Iain Beaton and Sam Schoonhoven. *On the unimodality of nearly well-dominated trees*. Preprint (2024), [arxiv:2411.02288](https://arxiv.org/abs/2411.02288).
- [CEWW24] François Charton, Jordan S. Ellenberg, Adam Zsolt Wagner, Geordie Williamson. *PatternBoost: Constructions in Mathematics with a Little Help from AI*. Preprint (2024), [arxiv:2411.00566](https://arxiv.org/abs/2411.00566).
- [GH18] David Galvin and Justin Hilyard. *The independent set sequence of some families of trees*, *Australas. J. Combin.*, **2** (2018), vol. 70, pp. 236–252.
- [HKM10] Prahladh Harsha, Adam Klivans and Raghu Meka. *An invariance principle for polytopes*. *Proceedings of the Forty-Second ACM Symposium on Theory of Computing* (2010). pp. 543–552.
- [Hei25] Steven Heilman. *Independent Sets of Random Trees and Sparse Random Graphs*. *Journal of Graph Theory*, **3** (2025), vol. 109, pp. 1–28.
- [KL23] Ohr Kadrawi and Vadim E. Levit. *The independence polynomial of trees is not always log-concave starting from order 26*. *Ars Mathematica Contemporanea*. **4** (2025), vol. 25, pp. 1–23.
- [KPT17] Tomer Kotek, James Preen and Peter Tittmann. *Domination Polynomials of Graph Products*. *Journal of Combinatorial Mathematics and Combinatorial Computing*. **1** (2017), vol. 101, pp. 245–258.
- [RS25] Eric Ramos, Sunny Sun. *An AI enhanced approach to the tree unimodality conjecture*. Preprint (2025), [arXiv:2510.18826](https://arxiv.org/abs/2510.18826).
- [WZ11] Yi Wang and Bao-Xuan Zhu. *On the unimodality of independence polynomials of some graphs*. *European J. Combin.* **1** (2011), vol. 32, pp. 10–20.
- [Zhu07] Zhi-Feng Zhu. *The unimodality of independence polynomials of some graphs*. *Australas. J. Combin.* **1** (2007), vol. 38, pp. 27–33.