Spring 2021

AN ENUMERATION OF NESTED NETWORKS

Nathan Cornelius
nmc71@zips.uakron.edu

AN ENUMERATION OF NESTED NETWORKS

A Thesis

Presented to

The Williams Honors College of The University of Akron

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science

Nathan Michael Cornelius

April, 2021

ABSTRACT

Nested networks have several applications in phylogenetics and electrical circuit theory. In many cases, there may exist more than one distinct network which correctly models a given data set. This proposes a combinatorial problem to determine all possible network solutions. In this paper, we partially solve this problem by developing exponential generating functions which enumerate all 1-nested and 2-nested unicyclic networks. We also describe our procedure to directly count all 1-nested and 2-nested networks and provide all 1-nested networks with 7, 8, and 9 terminal nodes.

Introduction

One important application of mathematical graph theory is the presentation of information in a compact and organized way. This includes the development of appropriate data structures which can effectively display various forms of information. This application is often used in phylogenetic biology, where evolutionary events such as recombination, hybridization, or lateral gene transfer are modeled using phylogenetic networks (Arenas et al., 2008; Huson et al., 2010). Likewise, various problems concerning the properties of electrical circuits can be modeled and solved using similar graphs (Forcey & Scalzo, 2020; Knox & Moradifam, 2017). In many problems, multiple networks can be considered as possible solutions. In those cases, one can utilize problem-specific algorithms to find which network is the best fit (Forcey et al., 2018; Saitou & Nei, 1987; Zhang & Sun, 2008.). This suggests a combinatorial problem to count all possible networks (up to isomorphism) for a given application.

Definitions and Motivation

Definitions for these types of networks vary among the literature. In the most general sense, phylogenetic and electrical networks are simple graphs with n boundary nodes of degree 1, called *leaves*. These nodes can be bijectively labeled with some section of the integers, $[n] = \{1, \ldots, n\}$, where each integer represents some taxon or boundary voltage. Networks in both applications are taken to be circular and planar. This narrows our discussion to graphs which can be embedded in some disc in the Euclidean plane. All boundary nodes are then restricted to lie on some circle which bounds the disc. Nodes interior to the disc will be unlabeled for our purposes.

We further restrict the graphs to be binary, where all non-leaf nodes are of degree 3. This simplification is necessary for the development of our counting procedure – e.g., the use of certain generating functions. Furthermore, graphs will be triangle-free, disallowing length 3 cycles. This constraint is justified in electrical circuit theory; researchers often make use of the Y-Δ transformation to create simpler graphs with equivalent resistance distances

(Akers, 1960; Curtis et al., 1998). This property is also mirrored in phylogenetics, as the linear functionals used in balanced minimal evolution (BME) algorithms are invariant under the transformation (Forcey & Scalzo, 2019).

These networks can vary in complexity, from simple binary trees to more complicated cyclic structures. There are two classifications in the literature that attempt to quantify this complexity: nestedness and graph level. We generalize the definition given by Gambette et al. (2017) and define an undirected network where every edge is contained in at most k cycles to be *k-nested*. Similarly, in accordance with Forcey & Durell (2019), we define a *level-k* network as a directed network where each biconnected component has a maximum of k reticulation nodes. In many cases, a level-k network has an underlying j-nested network as a subgraph where j $\leq$ k. In fact, it follows that a level-1 binary network must have an underlying undirected subgraph with at most 1 cycle. Thus, level-1 binary networks and 1-nested networks are synonymous (Forcey & Durell, 2019).

The consideration of these networks as new data structures for phylogenetic information has led to the development of new BME algorithms. Typically, solutions to a BME problem are convex polytopes of phylogenetic trees. Forcey & Durell (2019) generalized the method and allowed for solutions consisting of polytopes of phylogenetic networks. Through this generalization, they proved that the vertices of these polytopes are in bijection with the set of binary level-1 (1-nested) networks. This relationship led to the development of a formula for the number of 1-nested networks with n leaves and k non-trivial bridges[1]:

$$N(n,k) = \binom{n-3}{k} \frac{(n+k-1)!}{(2k+2)!!}.$$

The results of this formula for the first few collections of 1-nested networks ($3 \leq n \leq 9$) are shown in Table 1.

[1] *Bridges* are graph edges which connect two otherwise disconnected subgraphs. The restriction of *non-trivial* implies neither of the connected subgraphs is a leaf.

$n =$	$k =$	0	1	2	3	4	5
4		3					
5		12	30				
6		60	270	315			
7		360	2520	5040	3780		
8		2520	25200	75600	94500	51975	
9		20160	272160	1134000	2079000	1871100	810810

Table 1: The number of 1-nested (level-1) networks with n leaves and k non-trivial bridges.

This enumeration of 1-nested networks motivates two more ancillary questions: *How are 1-nested networks distributed?* and *Does a formula exist for counting 2-nested networks?*. With regard to the former, we would like to be able to count the number of 1-nested networks in particular subsets. This breaks down into two obvious cases: *unicyclic* networks, where there exists only one embedded cyclic subgraph, and networks where there are multiple cyclic subgraphs. The latter question implies a straightforward analysis of 2-nested networks, not unlike what Forcey & Durell (2019) accomplished with respect to 1-nested networks and their BME polytopes.

Fortunately, any 2-nested network can be created by the addition of another graph edge, or chord, to at least one underlying cyclic subgraph of some 1-nested network. Therefore, we need only develop a method of counting particular 1-nested networks (up to isomorphism) and then relate this to the number of ways to add a chord to their cyclic subgraphs. We will partially solve this problem by constructing an exponential generating function (EGF) for both 1-nested and 2-nested unicyclic networks.

Generating Functions for Unicyclic Networks

We will construct EGFs for unicyclic networks by first considering their fundamental components: cyclic subgraphs, binary trees, and inner cycle chords. To this end, we will formulate EGFs for these structures. They will then be used to derive the desired EGFs for 1-nested and 2-nested networks.

We begin this task with a basic derivation of a formula for the number of non-isomorphic cycles with $n \geq 4$ nodes.

Lemma 1: *The number of distinct (up to isomorphism) cycles of size* $n \geq 4$ *is* $(n-1)!/2$

Proof: A distinct cycle may be represented as a labeling of vertices on a regular polygon. So to count the number of distinct cycles with n nodes, we only have to count the number of ways to label a regular n-gon up to graphical isomorphism.

Consider a section of integers, $[n] = \{1, 2, ..., n\}$, with $n \geq 4$. Using these integers as labels, it follows that there are $n!$ ways to label the vertices of a regular n-gon. Since $n \geq 4 \geq 3$, it follows that the number of distinct isomorphisms of the regular n-gon is $|D_n| = 2n$, where D_n is the dihedral group of the n-gon. Therefore, given any labeling, there exists $2n$ equivalent labelings that can be obtained through an isomorphism. This constitutes an overcount by a factor of $2n$ in our intital count of $n!$ labelings. Therefore, the number of distinct cycles (up to isomorphism) is

$$\frac{n!}{2n} = \frac{(n-1)!}{2},$$

as claimed.

Using this formula, we now define the EGF for distinct cycles, $C(x)$, as

$$C(x) = \sum_{n=4}^{\infty} a_n \frac{x^n}{n!},$$

where

$$a_n = \frac{(n-1)!}{2}.$$

The next lemma will show that $C(x)$ can be written in terms of elementary functions.

Lemma 2: *The EGF for the number of distinct cycles of size* $n \geq 4$ *is*

$$C(x) = -\frac{\ln(1-x)}{2} - \frac{x}{2} - \frac{x^2}{4} - \frac{x^3}{6}.$$

Proof: We will use the result of Lemma 1 to construct the exponential generating function. Consider the basic power series,

$$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}.$$

By integration,

$$\int \sum_{n=0}^{\infty} x^n dx = \int \frac{1}{1-x} dx$$

$$\sum_{n=0}^{\infty} \frac{x^{n+1}}{n+1} = -\ln(1-x) + C.$$

Plugging in $x = 0$ will yield $C = 0$. Acknowledging this and shifting the index gives us,

$$\sum_{n=1}^{\infty} \frac{x^n}{n} = -\ln(1-x).$$

We now divide both sides by 2 and rearrange some terms to get

$$\sum_{n=1}^{\infty} \frac{(n-1)!}{2} \frac{x^n}{n!} = -\frac{\ln(1-x)}{2}.$$

Lastly, we subtract the first 3 terms of the sum from both sides:

$$\sum_{n=4}^{\infty} \frac{(n-1)!}{2} \frac{x^n}{n!} = -\frac{\ln(1-x)}{2} - \frac{x}{2} - \frac{x^2}{4} - \frac{x^3}{6}.$$

This ensures the formula starts at $n = 4$, as desired.

We will now emulate this procedure for binary trees with $n \geq 3$ leaves. But first we must prove a separate condition on the number of edges which compose them.

Lemma 3: *A binary tree with $n \geq 3$ (unlabeled) leaves has $2n-3$ edges.*

Proof: We will prove this by induction on n. Let $E(n)$ be the number of edges for a binary tree with n leaves. For the base case, $n = 3$, there is

only 1 such binary tree; it can be visualized as the shape of a "Y". Clearly, it has 3 edges. It then holds that $E(3) = 3 = 2(3) - 3$. Now assume that the formula holds for all binary trees with k leaves. Then $E(k) = 2k - 3$. Consider now that any binary tree with $k + 1$ leaves can be reduced to a binary tree with k leaves by the elimination a leaf edge. Such an elimination produces a graph with 2 less edges than the original graph. Therefore, the number of edges in all binary trees with $k + 1$ leaves is constant and equal to $E(k) + 2$. Consequently,

$$E(k+1) = E(k) + 2.$$

It follows that

$$E(k+1) = (2k-3) + 2 = 2(k+1) - 3,$$

by the induction hypothesis. Thus, $E(n) = 2n - 3$ for all $n \geq 3$.

We now proceed to a derivation of a formula for distinct binary trees with $n \geq 3$ leaves.

Lemma 4: *The number of distinct (up to isomorphism) binary trees with $n \geq 3$ labeled leaves is $(2n-5)!!$.*

Proof: We will prove this by induction on n. Let $B(n)$ be the number of distinct binary trees with n labeled leaves. As in Lemma 3, we note that there exists only 1 binary tree with 3 edges: the "Y" graph. There are 3! ways of labeling its leaves. But the graph also has 6 distinct symmetries (3 rotations and 3 reflections). Therefore, for any labeling of leaves, there are 6 equivalent labelings that can be obtained by an isomorphism. This implies that there is $\frac{3!}{6} = 1$ way of distinctly labeling the graph. Thus, $B(3) = 1 = (2(3) - 5)!!$.

Now assume the formula holds for some integer $k \geq 3$. Then

$$B(k) = (2k-5)!!.$$

Define Ω to be the set of binary trees with $k+1$ leaves. Likewise, define Λ to be the set of binary trees with k leaves. Now consider an arbitrary element, $\omega \in \Omega$. The removal of any leaf edge from ω will result in a unique binary tree subgraph, $\lambda \in \Lambda$. Since we may observe this process in reverse, it follows

that ω can be constructed by the addition of a $(k+1)$th leaf to a specific edge in λ.

Conversely, it is also true that any addition of a new leaf edge to any $\lambda \in \Lambda$ results in a unique $\omega \in \Omega$. Therefore, $B(k+1) = |\Omega|$ is equal to every possible construction on every element of Λ. And since every $\lambda \in \Lambda$ is binary, a new edge can only be attached to another edge. Thus, the number of ways to add an edge to any $\lambda \in \Lambda$ is $E(k)$, the number of unlabeled edges in a binary tree with k leaves (as defined in lemma 3). Consequently,

$$B(k+1) = |\Omega| = |\Lambda| \cdot E(k) = B(k)E(k).$$

And thus,

$$B(k+1) = B(k)E(k).$$

By the induction hypothesis, we know that $B(k) = (2k-5)!!$. And by Lemma 3, we know that $E(k) = (2k-3)$. Thus,

$$B(k+1) = (2k-5)!!(2k-3)$$

which simplifies to

$$B(k+1) = (2(k+1)-5)!!.$$

Therefore, by induction, the formula holds hold for all $n \geq 3$.

We now have all the necessary components to begin the development of an EGF for unicyclic networks. The main idea behind this construction involves the enumeration of all the ways to uniquely "glue" labelled binary trees to the vertices of cycles, where the cycles are represented by a regular polygon. In general, one may count the number of ways to attach a set of discrete structures to another set of discrete structures by composing particular EGFs for those structures (Stanley, 1999). Bergeron et al.(2013) provides a rigorous development of this idea, with a general proof that a set of structures constructed in this manner is enumerated by the composition of the EGFs for the component structures. This proof is beyond the scope of this paper. And as such, we will only summarize the necessary requirements to construct the EGFs that we desire.

As explained by Bergeron et al.(2013), we may construct the desired EGF for unicyclic networks by first attaching an additional element, which

we will denote as $'*'$, to every labelled binary tree. This provides us with a marker by which to attach the tree to any vertex of a cycle. In effect, this transforms the set of binary trees with n labeled nodes into the set with $n+1$ nodes. This collection of structures is enumerated by

$$B(n+1) = (2(n+1)-5)!! = (2n-3)!!$$

for $n \geq 3$, where B(n) is the function as described in Lemma 4.

We could now define the EGF for binary trees and proceed to compose the function with the cycle EGF. But this would only provide us with networks where each vertex is attached to a tree with 3 or more leaves. We would also like to include all such cases where a vertex is attached to a single terminal node or possibly 2 terminal nodes. To remedy this, we simply define B(1) = 1 and B(2) = 1, since there is only 1 way (up to isomorphism) to attach a star element to a single labeled point or to a pair of labeled points. The EGF for this sequence can now be defined as

$$T(x) = \sum_{n=0}^{\infty} b_n \frac{x^n}{n!}.$$

where

$$b_n = \begin{cases} 1 & n < 2 \\ (2n-3)!! & n \geq 2 \end{cases}.$$

The nth coefficient of this EGF gives us the number of distinct labeled gluings of n terminal points to a single vertex. The next step would involve a formulation of this EGF in terms of elementary functions. Fortunately, Stanley (1999) considers this exact sequence of numbers when counting binary partitions. In his book, he provides a proof in which he shows that

$$T(x) = 1 - \sqrt{1-2x},$$

where T(x) is as we have defined. Finally, we may construct the EGF for 1-nested (unicyclic) networks.

Theorem 1: *The EGF for the number of nonisomorphic 1-nested networks with n terminal nodes is*

$$U(x) = -\frac{1}{4}\ln(1-2x) + \frac{5}{3}\sqrt{1-2x} - \frac{1}{3}x\sqrt{1-2x} + \frac{3}{2}x - \frac{5}{3}$$

Proof: We have already established both exponential generating functions, C(x) and T(x), for cycles with n terminal nodes and binary trees with n+1 leaves, respectively. By Bergeron's (2015) theorem, it follows that the EGF for the constructions of all possible unicyclic cycles with n terminal nodes is

$$U(x) = C(T(x))$$

where

$$C(T(x)) = \ln\left(1 - (1 - \sqrt{1-2x})\right) - \frac{1-\sqrt{1-2x}}{2} - \frac{(1-\sqrt{1-2x})^2}{4} - \frac{(1-\sqrt{1-2x})^3}{6}$$

which simplifies to

$$U(x) = -\frac{1}{4}\ln\left(1-2x\right) + \frac{5}{3}\sqrt{1-2x} - \frac{1}{3}x\sqrt{1-2x} + \frac{3}{2}x - \frac{5}{3}.$$

The number of unicyclic 1-nested networks with n terminal nodes can be determined by evaluating the nth derivative of U(x) at zero. This procedure yields the following sequence,

$$U_n = 0, 0, 0, 0, 3, 42, 555, 7920, 125055, 2187990, ...$$

The first 4 terms in the sequence are 0. Recall that the definition of a cycle was restricted to sizes ≥ 4. So it is expected that $U_n = 0$ for $0 < n < 4$, as this represents the counting procedure done on the empty set. The next 3 terms of the sequence, 3, 42, and 55, coincide with previous calculations done by Drew Scalzo (2020). The rest of the terms, $7920, 125055$, and 2187990, corresponding to $n = 7, 8$, and 9, have also been verified by the direct enumeration of all network types for each respective n. A summary of this procedure is discussed later in this paper.

We may similarly extend this construction to form an EGF for 2-nested networks. The main relationship between 1-nested and 2-nested networks is that any 2-nested network can be formed by adding chords interior to a cyclic subgraph of a 1-nested network. We will exploit this fact.

Theorem 2: *The EGF for the number of nonisomorphic 2-nested (unicyclic) networks is*

$$S(x) = \frac{(1-\sqrt{1-2x})^4}{4-8x}.$$

Proof: First, observe that the number of ways to add an edge (chord) to any 1-nested network with a cyclic subgraph with $n \geq 4$ edges is $\frac{n(n-3)}{2}$. This can be seen by representing the 1-nested network in such a way that its cyclic subgraph is a regular polygon. There are then n ways to pick an edge. Recall that we disallowed cycles of length 3 in our networks. Thus, there are $n-3$ ways to add a chord connecting the chosen side to some other edge while maintaining this restriction. But doing this to every edge of the cycle will produce an overcount by a factor of 2, since any additional edge connects to 2 edges of the original cycle.

With this established, a formula for the number of 2-nested (unicyclic) cycles with n$\geq$ 4 leaves can obtained by multiplying the formula above to the formula for cycles in Lemma 1. By doing this we obtain

$$Q(n) = \frac{(n-1)!}{2}\frac{n(n-3)}{2}$$

To find the EGF , Ch(x), for this sequence, we start as we have before with the standard power series formula:

$$\sum_{n=0}^{\infty} x^n = \frac{1}{1-x}.$$

Differentiate to obtain

$$\sum_{n=1}^{\infty} nx^{n-1} = \frac{1}{(1-x)^2}.$$

Then re-index back to $n = 0$ and multiply by $\frac{x^4}{4}$ on both sides. This gives

$$\sum_{n=0}^{\infty} \frac{1}{4}(n+1)x^{n+4} = \frac{x^4}{4(1-x)^2}$$

. Now re-index to $n = 4$ and multiply by $\frac{n!}{n!}$ in sum to get

$$\sum_{n=4}^{\infty} \frac{(n-3)}{4}\frac{n!}{n!}x^n = \frac{x^4}{4(1-x)^2}.$$

This simplifies to

$$\sum_{n=4}^{\infty} \frac{(n-1)!}{2} \frac{n(n-3)}{2} \frac{x^n}{n!} = \sum_{n=4}^{\infty} Q(n) \frac{x^n}{n!} = \frac{x^4}{4(1-x)^2}$$

Thus,

$$Ch(x) = \frac{x^4}{4(1-x)^2}$$

This is the EGF for 2-nested (unicyclic) cycles. To construct 2-nested networks, we attach binary trees to these cycles just as we did in Theorem 1. The EGF that counts all such structures for $n \geq 4$ is the composition

$$S(x) = Ch(T(x)) = \frac{(1-\sqrt{1-2x})^4}{4(1-(1-\sqrt{1-2x}))^2} = \frac{(1-\sqrt{1-2x})^4}{4-8x},$$

as claimed.

By calculating $S^n(0)$ for $n \geq 0$, we obtain the sequence

$$S_n = 0, 0, 0, 0, 6, 120, 2070, 36540, 688590, 14016240, ...$$

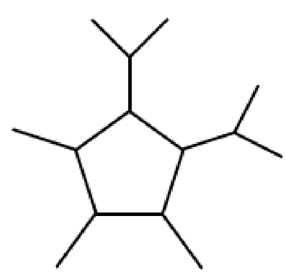
Yet again, we expect $S_n = 0$ for $0 \leq n \leq 3$. For $n = 4, 5$, there is at most one cycle in any binary network, so S_4 and S_5 give the full number of binary 2-nested networks. Furthermore, full counts of binary 2-nested networks can be obtained directly by revisiting each type of 1-nested network with n leaves. For each type of network, you simply multiply the number of ways to label it (up to isomorphism) by all the number of ways to add chords to make it 2-nested. This procedure is done for $n = 4, 5$, and 6 by Drew Scalzo (2020). The rest of the terms corresponding to $n = 7, 8$, and 9 can be directly calculated from the network drawings at the end of this paper.

Direct Network Counting

In the previous section, we have developed means to count a specific class of 1 or 2-nested networks. We would also like to determine the cardinality of each isomorphism class for a given number of leaves. So far, there is no convenient way of accomplishing this other than direct calculation. Thus, we can proceed only by drawing graphical representations for each class and counting all the ways to label them (up to isomorphism). Fortunately, we now have three formulas to help check calculations: the EGFs for 1 and 2-nested networks as well as Forcey and Durell's (2018) formula which was mentioned at the onset of this paper.

Two networks are considered isomorphic if there exists a bijection between them that preserves all boundary nodes and edge connections. If we imagine all networks as thin pieces of wire in 3-space, then an isomorphism between two networks is equivalent to manipulating the wire of one network until we obtain the other. These manipulations can be done in simple steps: rotations, flips (reflections), and bridge twisting. We will call each manipulation of one of these types a *simple symmetry.* Every isomorphism on a network can be obtained by a finite number of compositions of these symmetries. Thus, one may obtain an entire equivalence class of networks by choosing one and conducting every combination and permutation of simple symmetries on that network[2]. Using this observation, we can easily count each equivalence class of networks with n leaves.

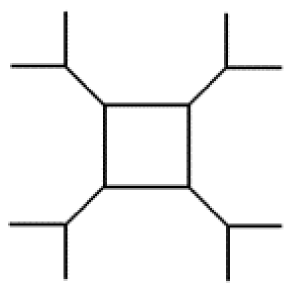
Let's consider the following network as an example:



[2]In practice, considering all permutations of simple symmetries is quite unnecessary. This is because most permutations will be equivalent to each other.

There are 7! ways of labeling its leaves. But there are also 3 simple symmetries: a twist about each binary tree in the top right and twisting the whole graph about the bridge in the bottom left. Thus, given any particular labeling, there are 2^3 equivalent labelings which correspond to different combinations of these symmetries. Thus, there are $\frac{7!}{2^3} = 630$ distinct networks of this form.

In many cases, a simple symmetry is not equivalent to a combination of other simple symmetries. This is usually the case with networks that are inherently non-symmetrical, such as the previous example. But in cases where a network has many symmetries, care must be taken. Consider the following network:



This network has 4 rotational symmetries, 4 flip symmetries, and 8 twist symmetries. Several of these are equivalent to combinations of other simple symmetries. For instance, a 90 degree rotation is equivalent to reflecting about the vertical and then reflecting about the horizontal. In cases like this, it is best to count the labelings in a way that eliminates the need to consider rotational symmetry.

Considering this most recent network, choose 2 arbitrary labels to be assigned to the upper left corner and permute them. There are $2\binom{8}{2}$ ways of doing this and then 6! ways of labeling the rest of the leaves. Since we are restricting ourselves to cases where the arbitrary terms are in the upper left corner, it follows that we have no reflectional or rotation symmetry, for any of these would remove those labels from that corner. Now we only need to consider bridge twists, of which there are 5. Therefore, there are

$$2\binom{8}{2}\frac{6!}{2^5} = 1260$$

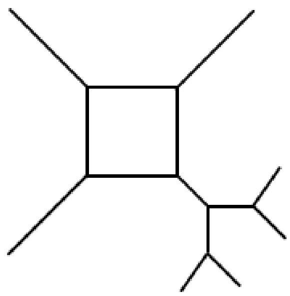
labelings. But we could have chosen the 2 arbitrary terms to be in any of the 4 corners, and we would have proceeded in an identical manner. To account for this, we must divide our answer by 4 to get $1260/4 = 315$. This is the number of distinct networks of that form.

This procedure has been done for all equivalence classes of networks with $n = 7, 8$ and 9 leaves. Drawings and final counts for the number of distinct networks of a particular type are provided at the end of this paper. It is also a quick check to see that the sum of these 1-nested networks for a given n is equivalent to Forcey and Durell's (2020) formula given at the onset of this paper. The same is true of the sums of counts for distinct 1-nested unicyclic networks; they are in accordance with the EGF we have derived.

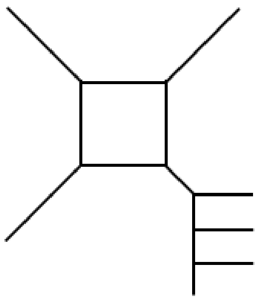
This procedure can be extended to 2-nested networks as well. First you must count all of the distinct 1-nested networks. Then for each type, multiply the number calculated to the number of ways of adding a chord (or chords) to interior polygons. Doing this for the unicyclic networks we have provided at the end of this paper, we obtain the sums $36540, 688590$, and 14016240, as predicted by our second EGF. And so it is apparent that our direct calculations for distinct 1-nested and 2-nested networks are consistent with both exponential generating functions derived in this paper.

1-Nested Networks with n=7 Leaves

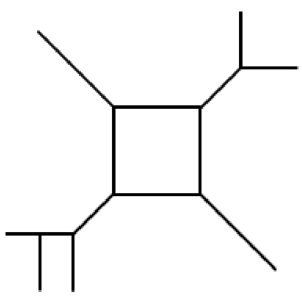
Let K be the number of networks in each indicated class.



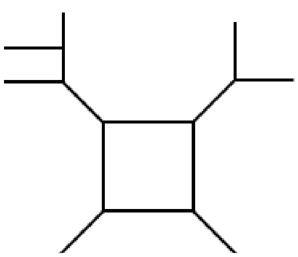
K = 315



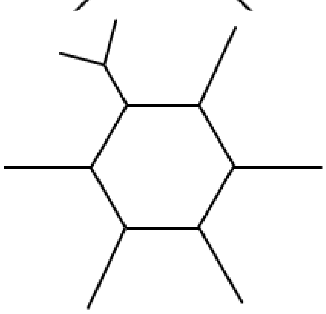
K = 1,260



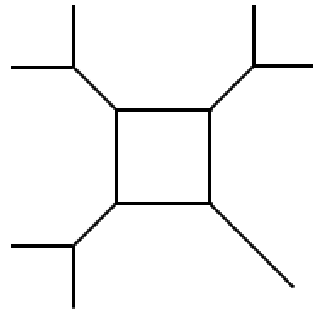
K = 630



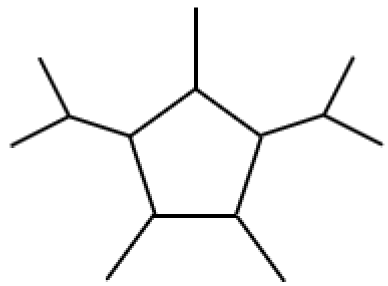
K = 1,260



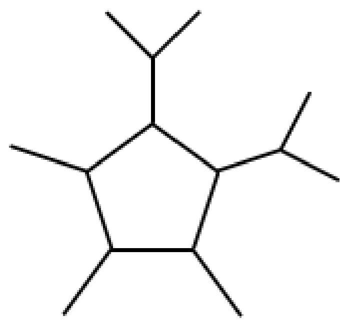
K = 1,260



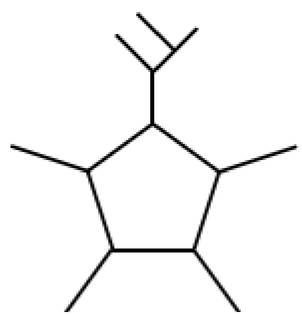
K = 315



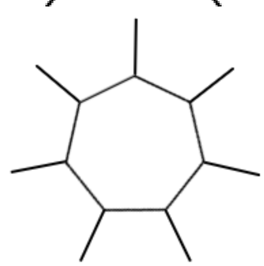
K = 630



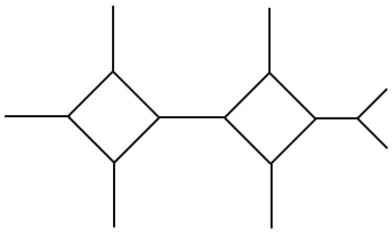
K = 630



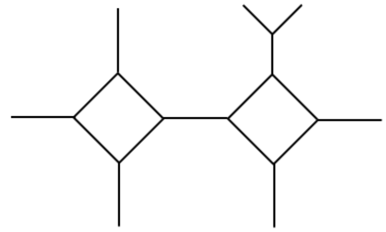
K = 1,260



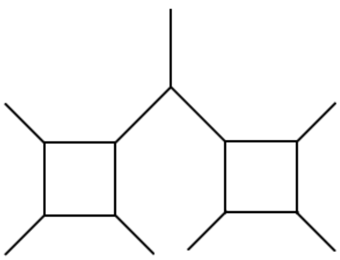
K = 360



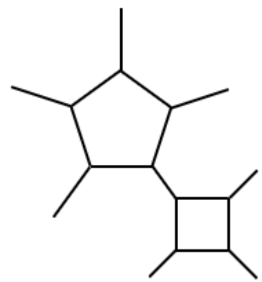
K = 630



K = 1,260



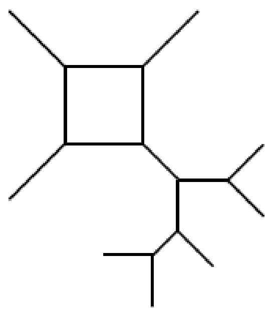
K = 630



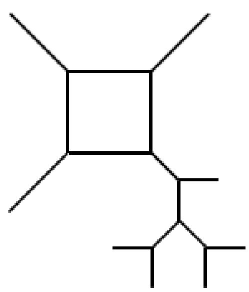
K = 1,260

1-Nested Networks with n=8 Leaves

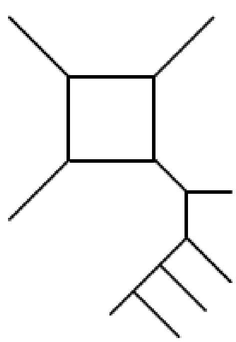
Let K be the number of networks in each indicated class.



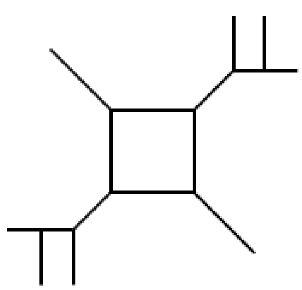
K = 5,040



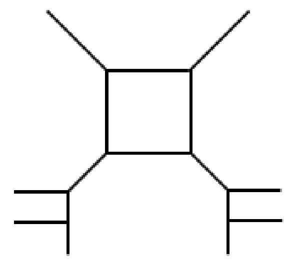
K = 2,520



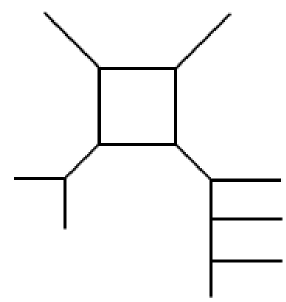
k = 10,080



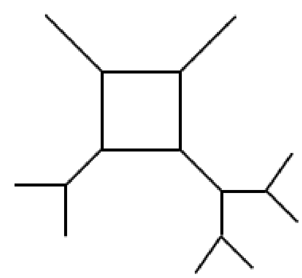
K = 2,520



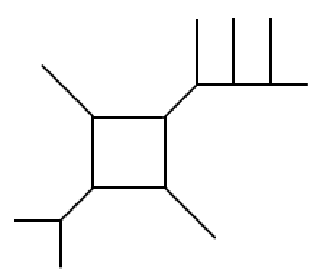
K = 5,040



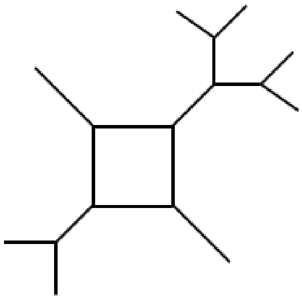
K = 10,080



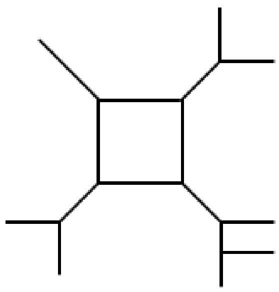
K = 2,520



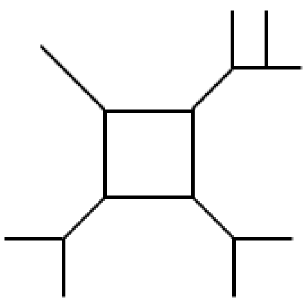
K = 5,040



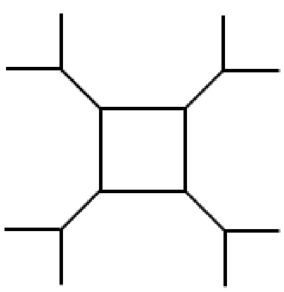
K = 1,260



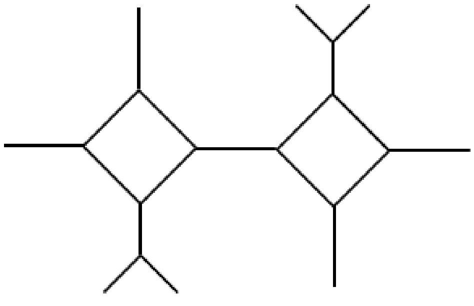
K = 2,520



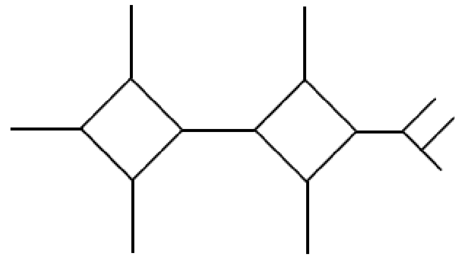
K = 5,040



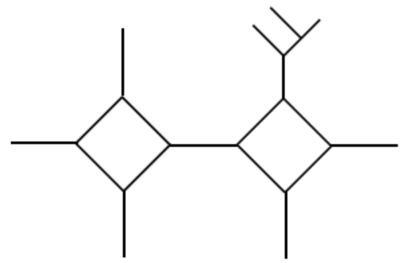
K = 315



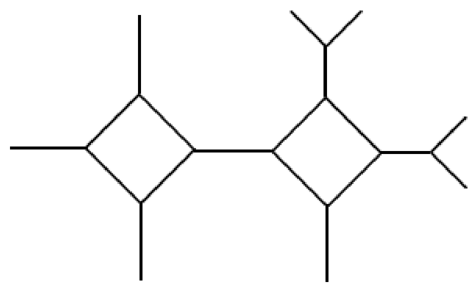
K = 5,040



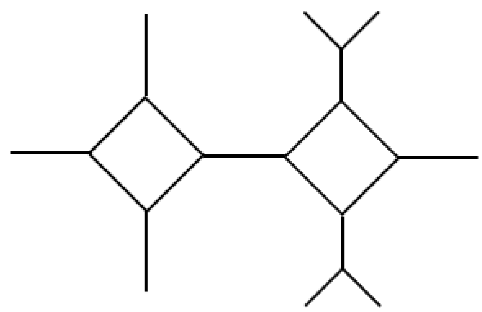
K = 5,040



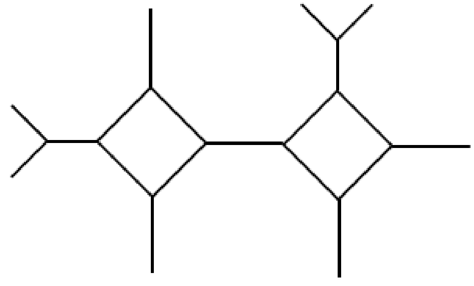
K = 10,080



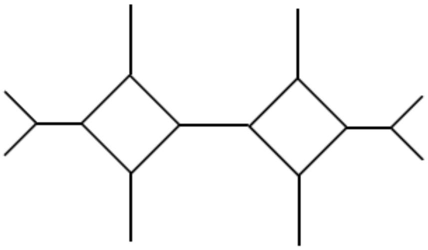
K = 5,040



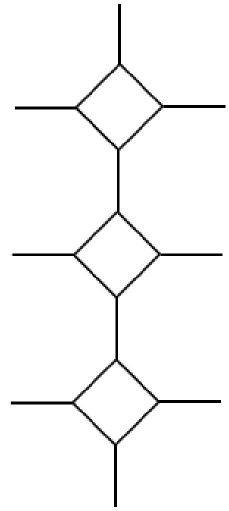
K = 1,260



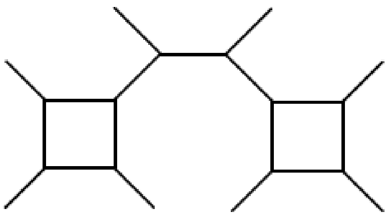
K = 5,040



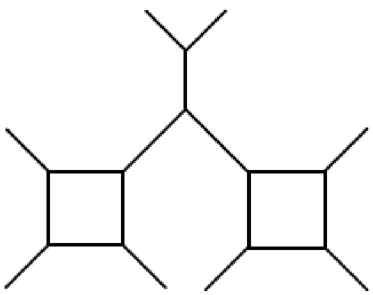
K = 1,260



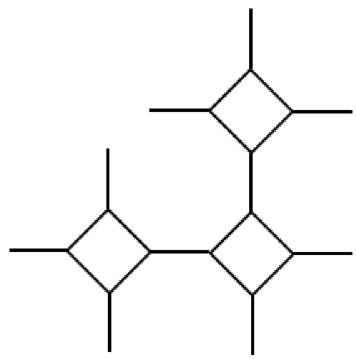
K = 2,520



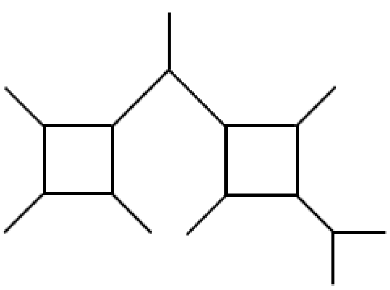
K = 5,040



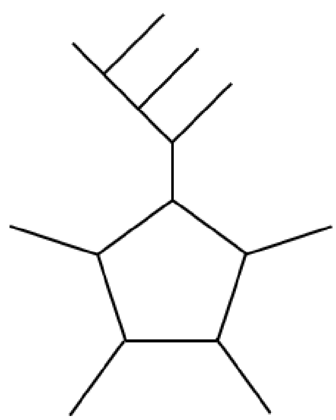
K = 2,520



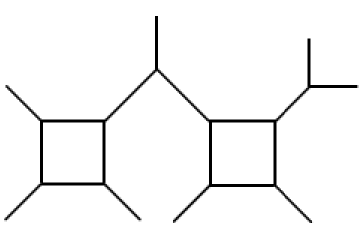
K = 5,040



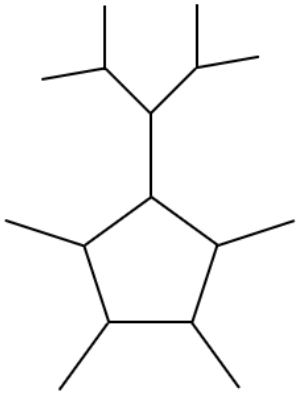
K = 5,040



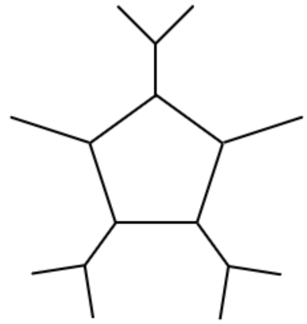
K = 10,080



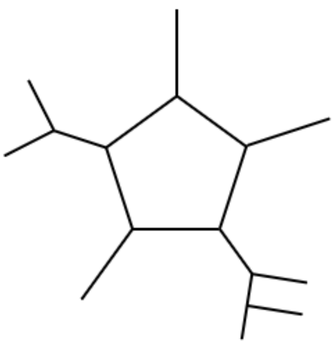
K = 10,080



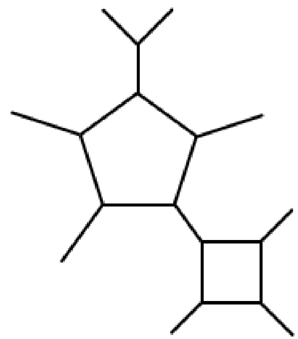
K = 2,520



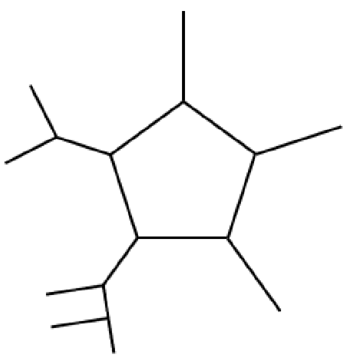
K = 2,520



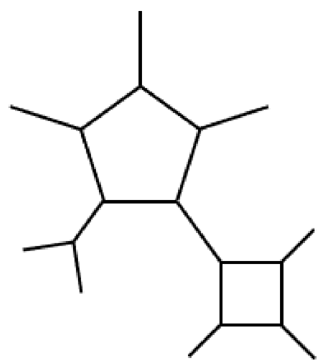
K = 10,080



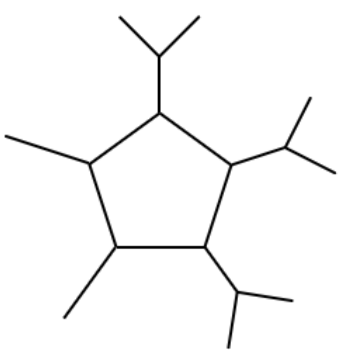
K = 10,080



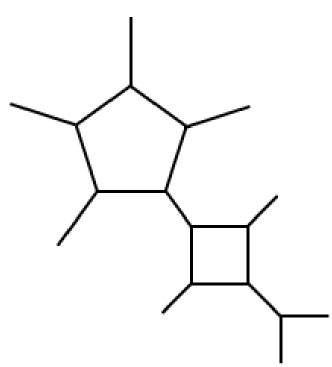
K = 10,080



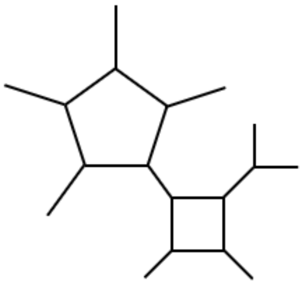
K = 10,080



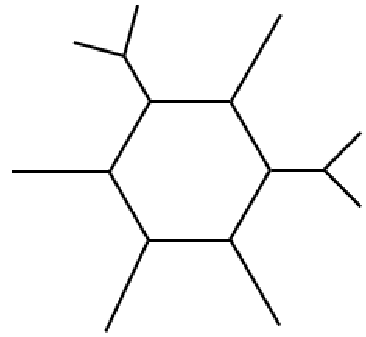
K = 2,520



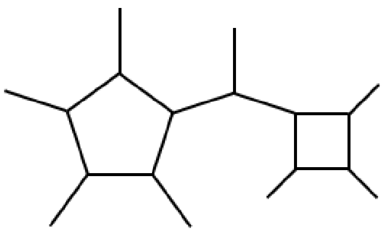
K = 5,040



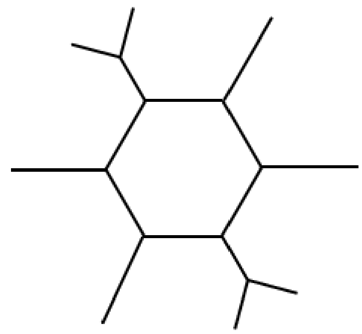
K = 10,080



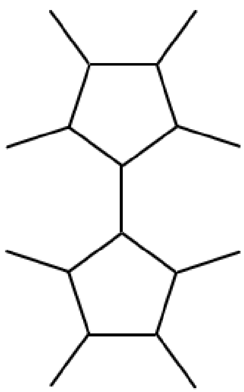
K = 5,040



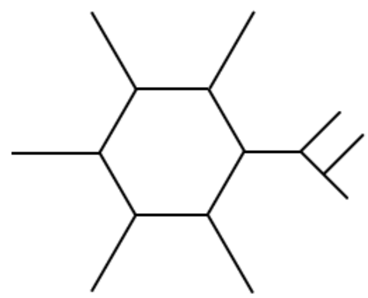
K = 10,080



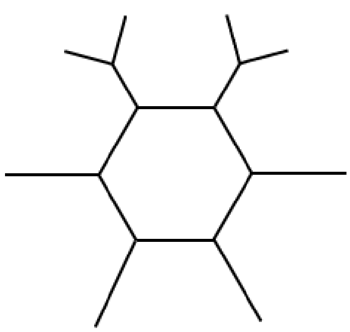
K = 2,520



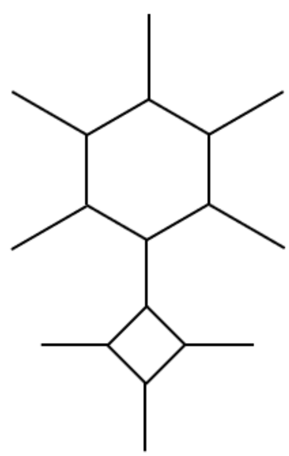
K = 5,040



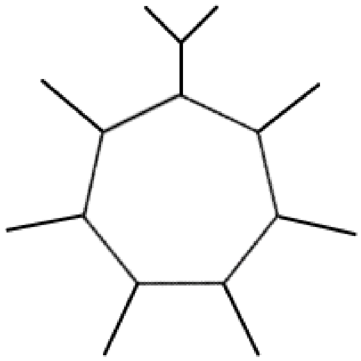
K = 10,080



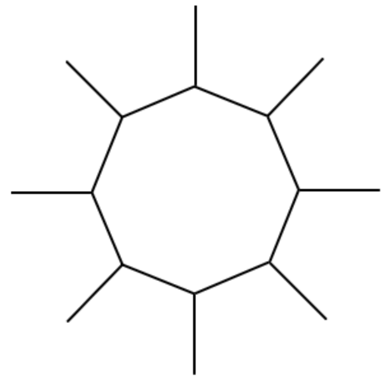
K = 5,040



K = 10,080



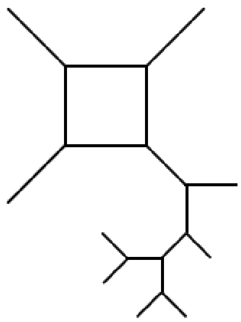
K = 10,080



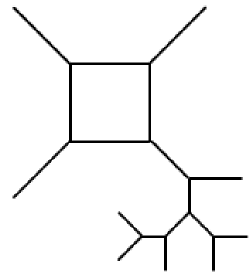
K = 2,520

1-Nested Networks with n=9 Leaves

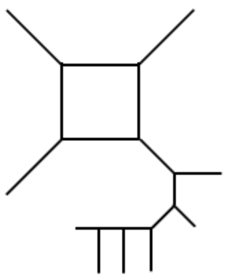
Let K be the number of networks in each indicated class.



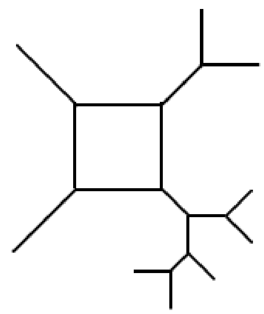
K = 22,680



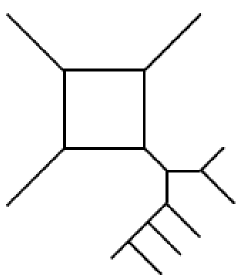
K = 45,360



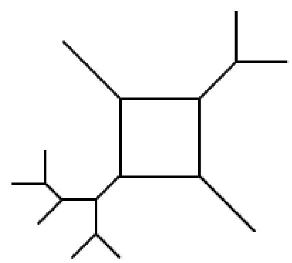
K = 90,720



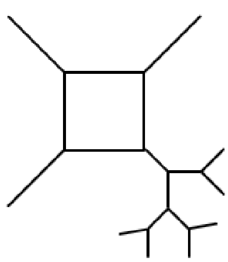
K = 45,360



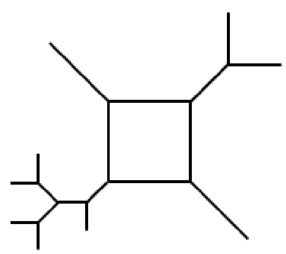
K = 45,360



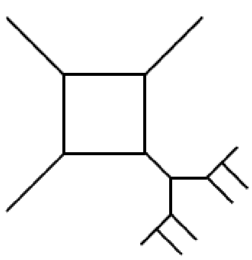
K = 22,680



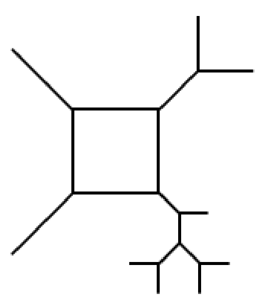
K = 11,340



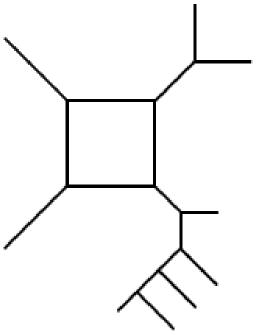
K = 11,340



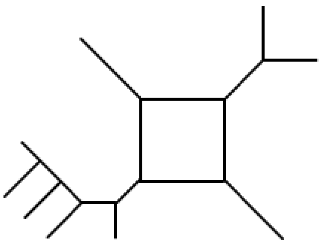
K = 22,680



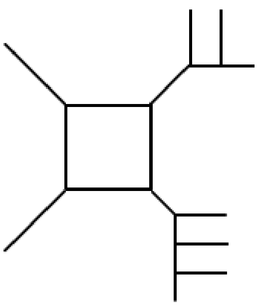
K = 22,680



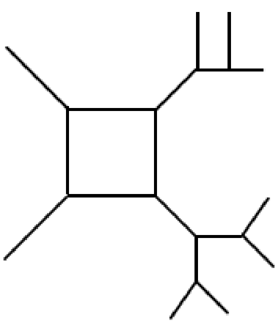
K = 90,720



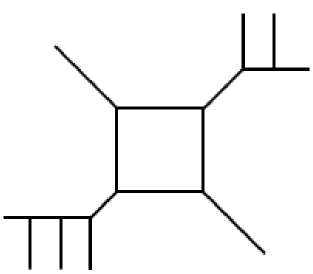
K = 45,360



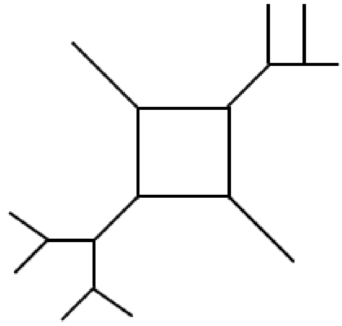
K = 90,720



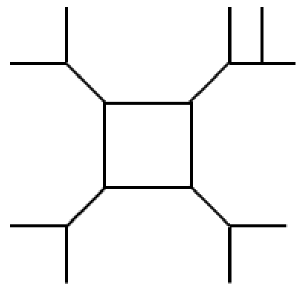
K = 22,680



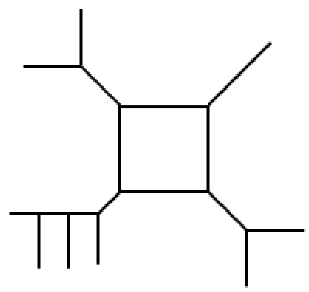
K = 45,360



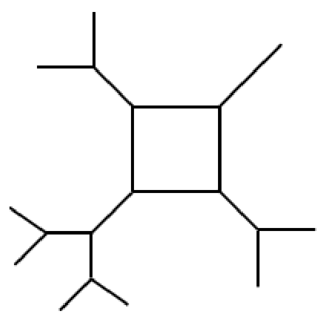
K = 11,340



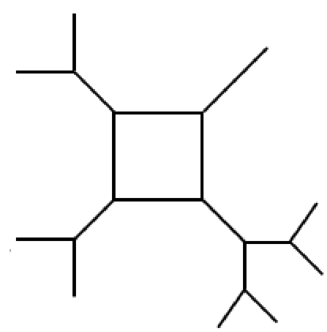
K = 11,340



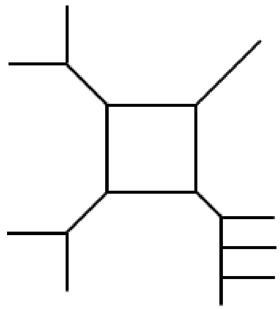
K = 22,680



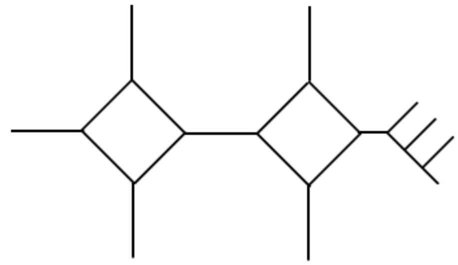
K = 5,670



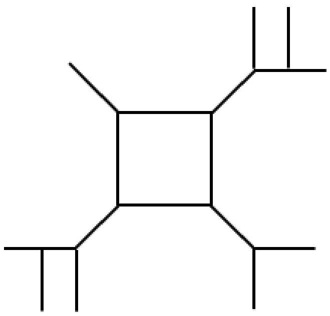
K = 11,340



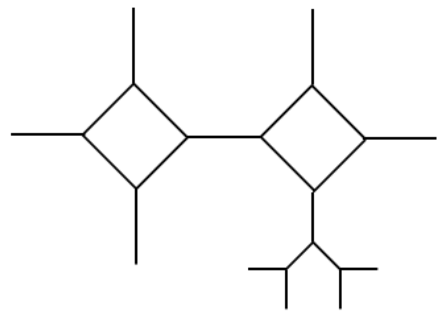
K = 45,360



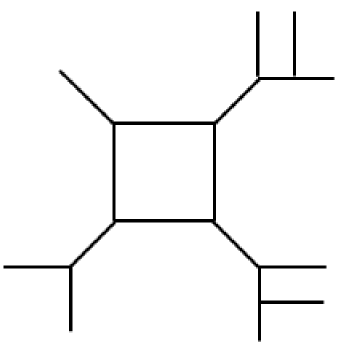
K = 45,360



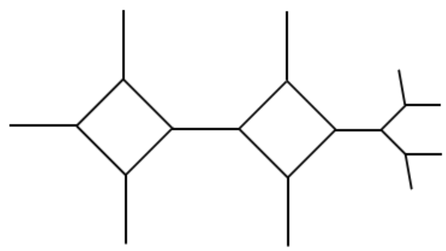
K = 22,680



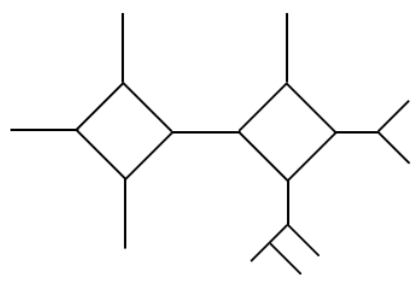
K = 22,680



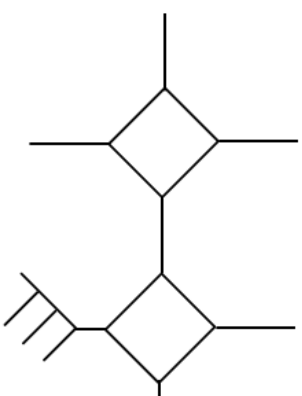
K = 45,360



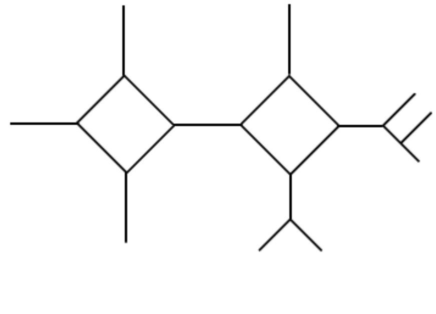
K = 11,340



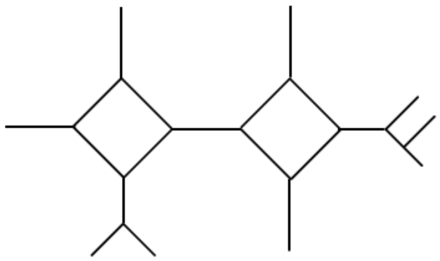
K = 45,360



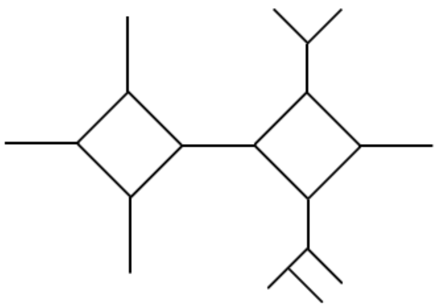
K = 90,720



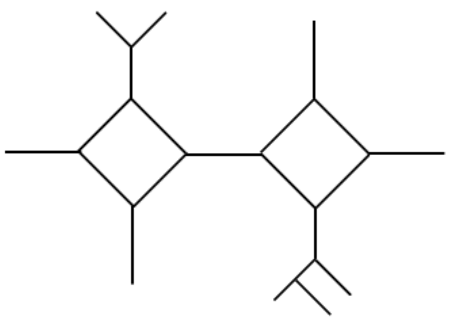
K = 45,360



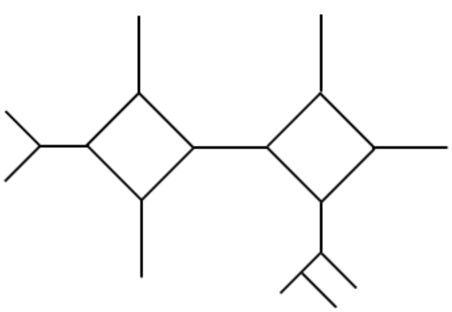
K = 45,360



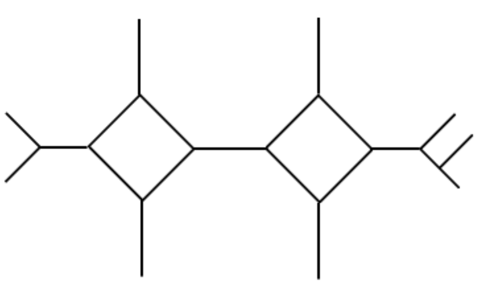
K = 45,360



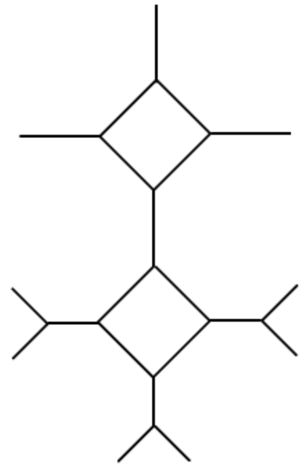
K = 90,720



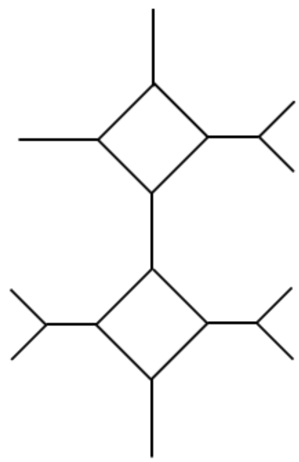
K = 45,360



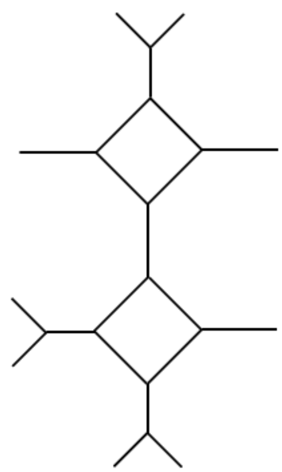
K = 22,680



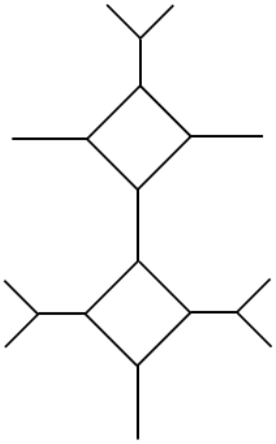
K = 11,340



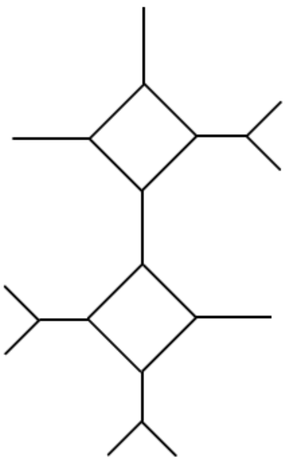
K = 22,680



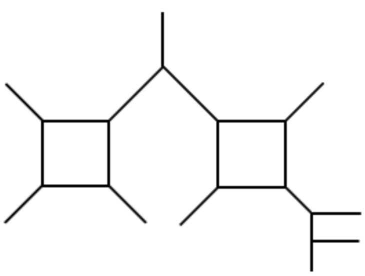
K = 22,680



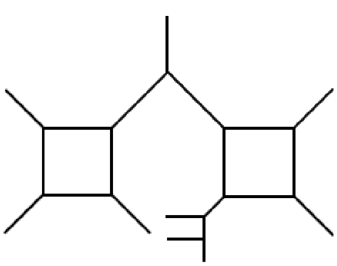
K = 11,340



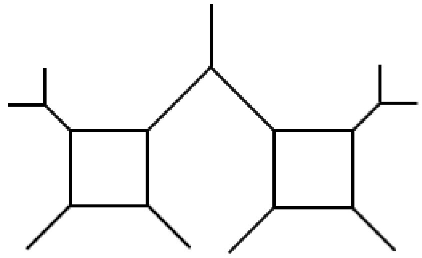
K = 45,360



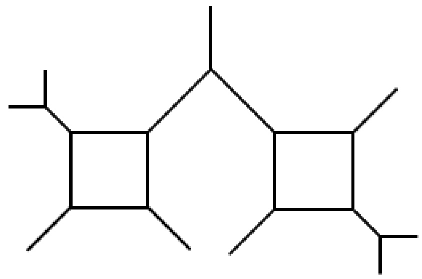
K = 45,360



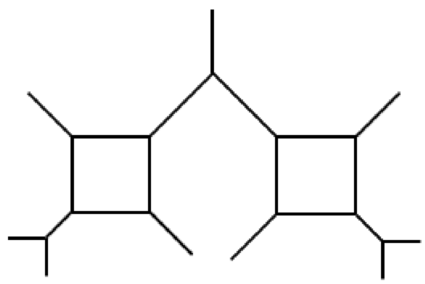
K = 90,720



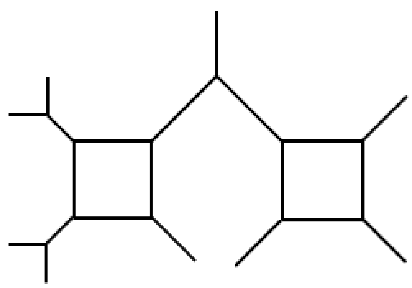
K = 45,360



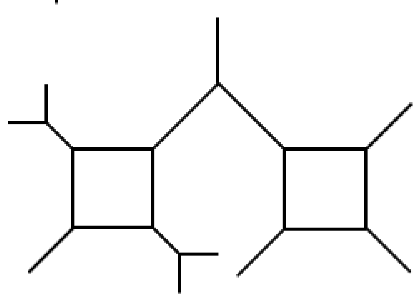
K = 45,360



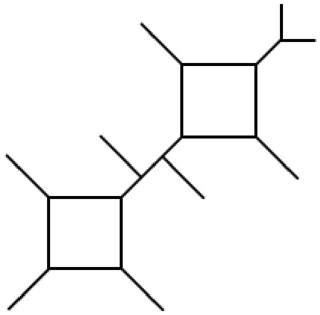
K = 11,340



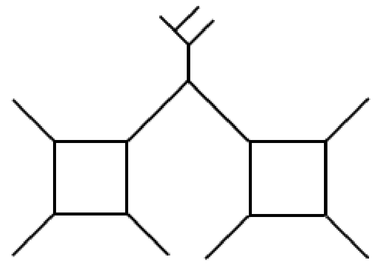
K = 45,360



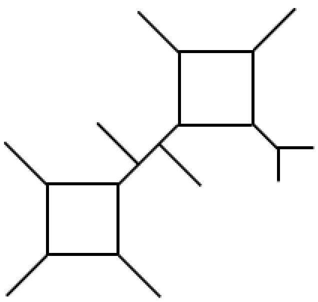
K = 22,680



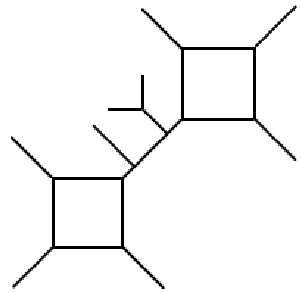
K = 45,360



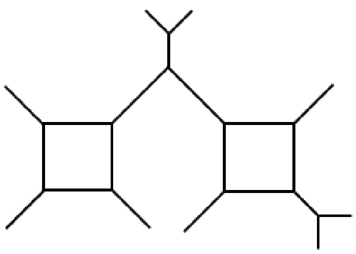
K = 22,680



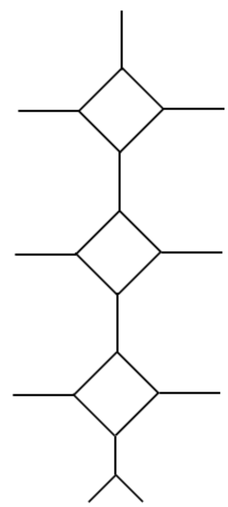
K = 90,720



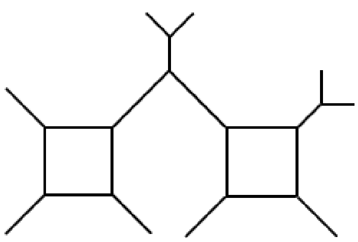
K = 45,360



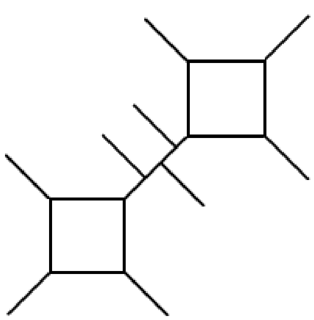
K = 22,680



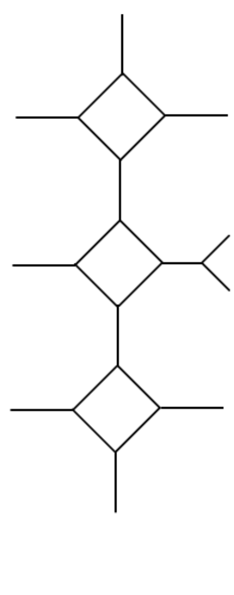
K = 22,680



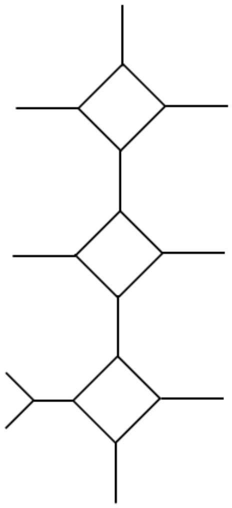
K = 45,360



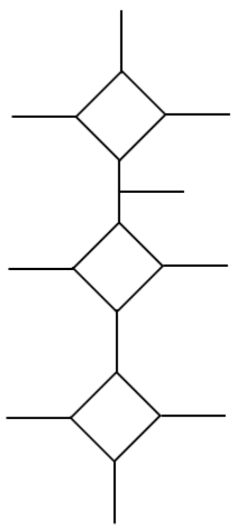
K = 45,360



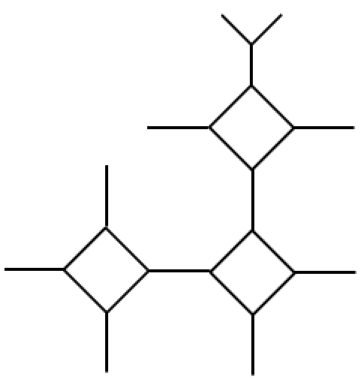
K = 22,680



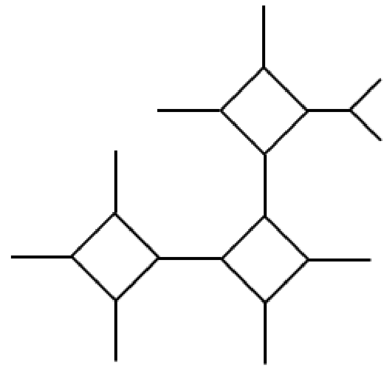
K = 45,360



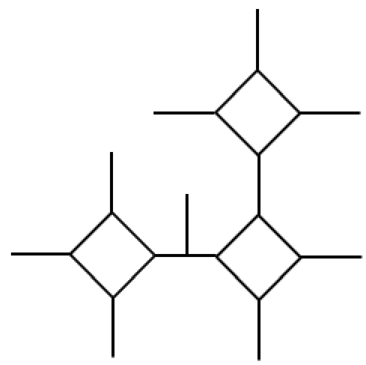
K = 45,360



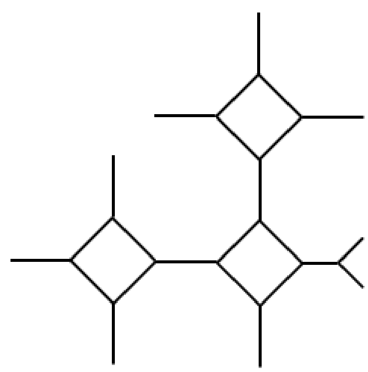
K = 45,360



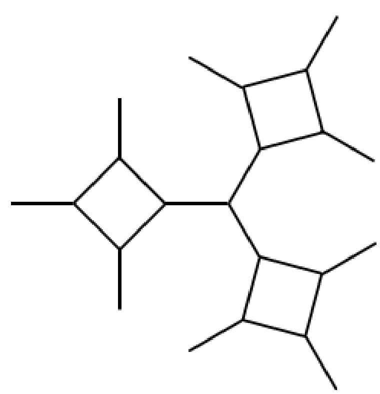
K = 90,720



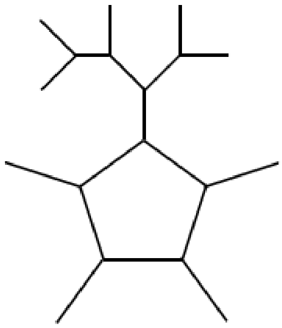
K = 90,720



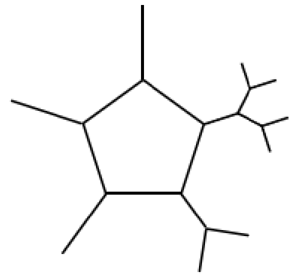
K = 45,360



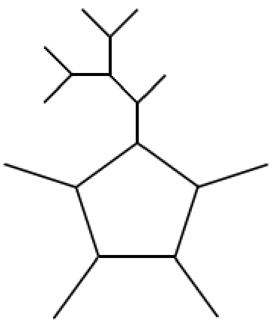
K = 7,560



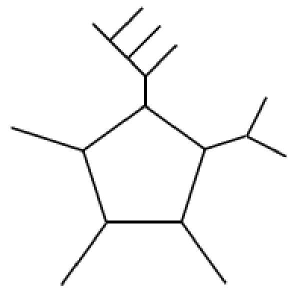
K = 45,360



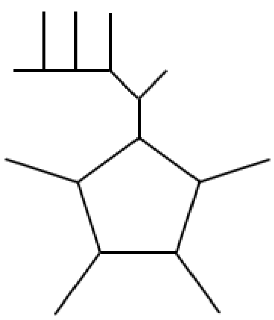
K = 22,680



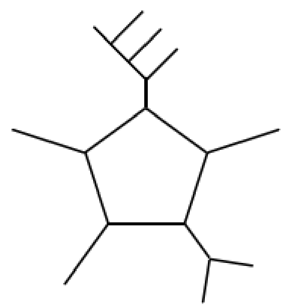
K = 22,680



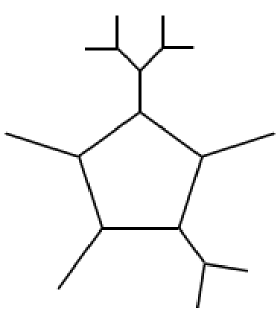
K = 90,720



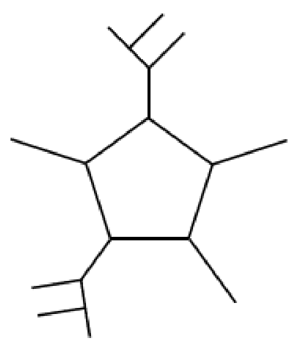
K = 90,720



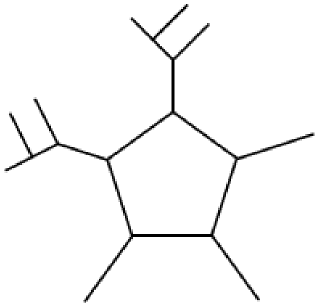
K = 90,720



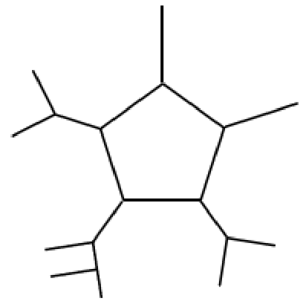
K = 22,680



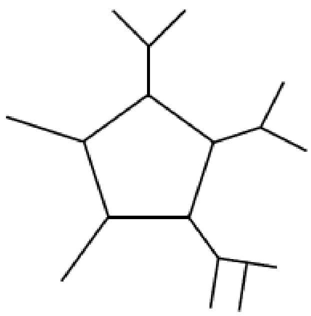
K = 45,360



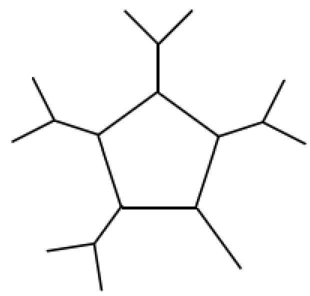
K = 45,360



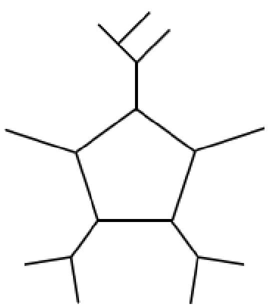
K = 22,680



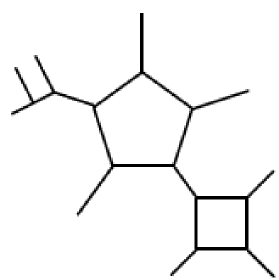
K = 45,360



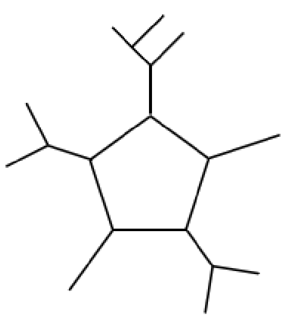
K = 11,340



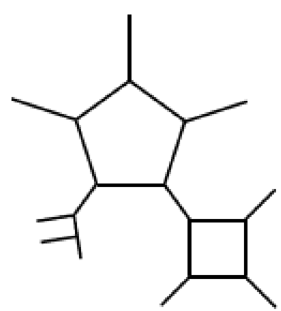
K = 22,680



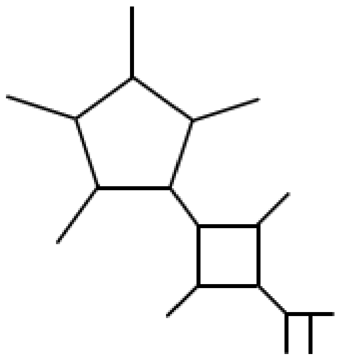
K = 90,720



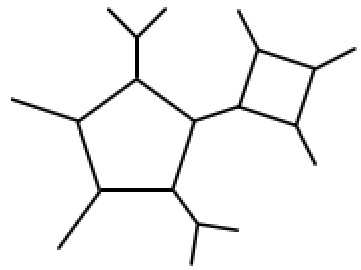
K = 45,360



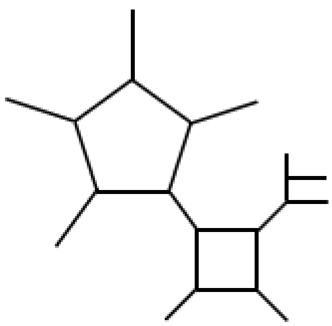
K = 90,720



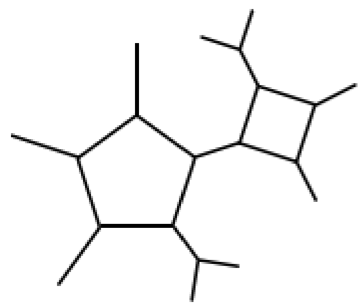
K = 45,360



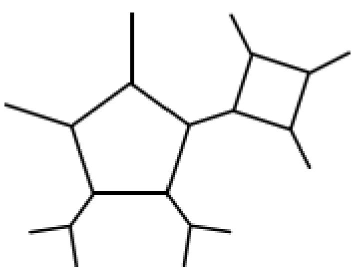
K = 22,680



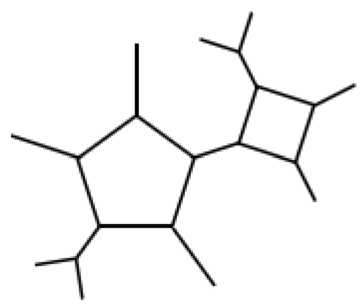
K = 90,720



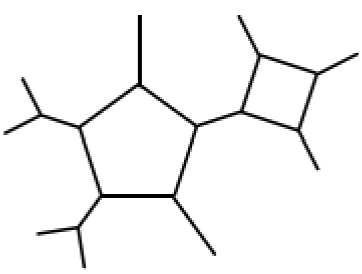
K = 90,720



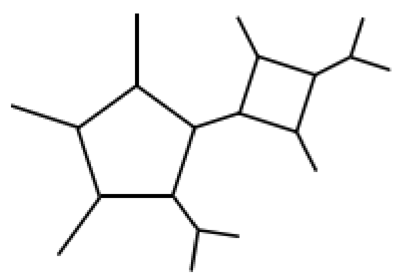
K = 45,360



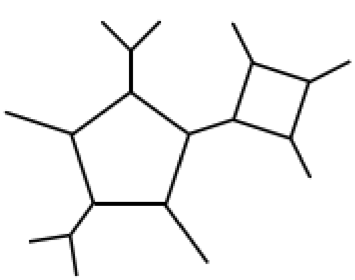
K = 90,720



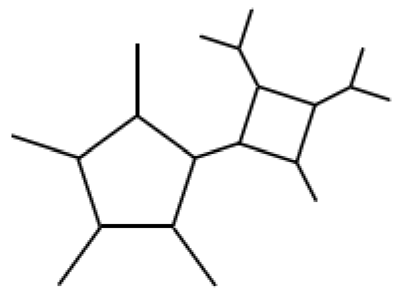
K = 22,680



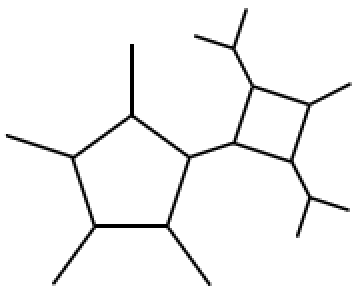
K = 45,360



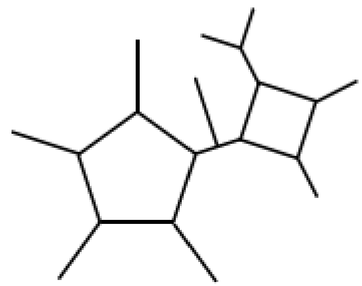
K = 45,360



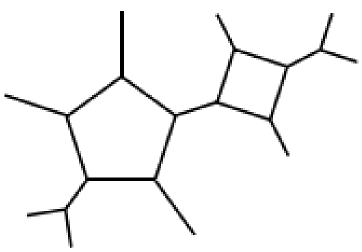
K = 45,360



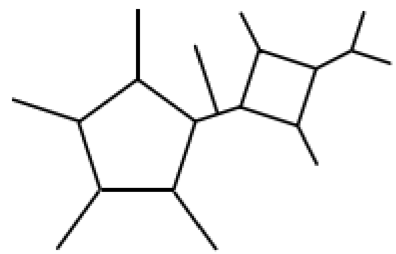
K = 22,680



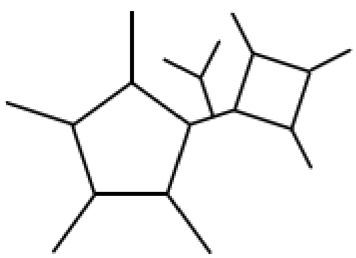
K = 90,720



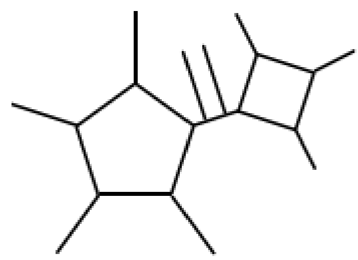
K = 45,360



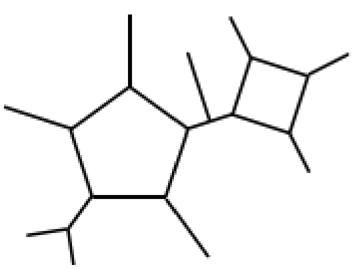
K = 45,360



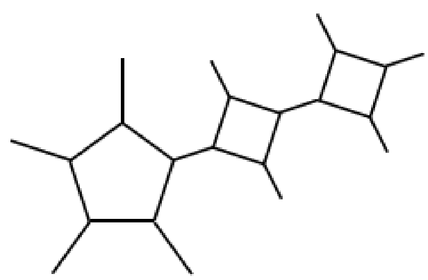
K = 45,360



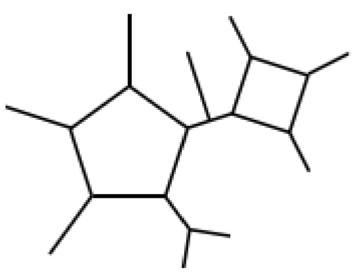
K = 90,720



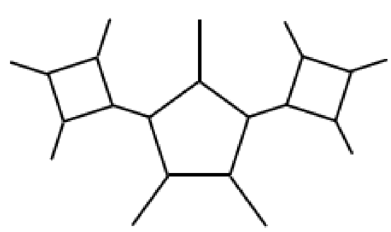
K = 90,720



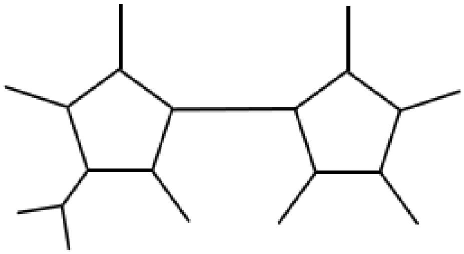
K = 45,360



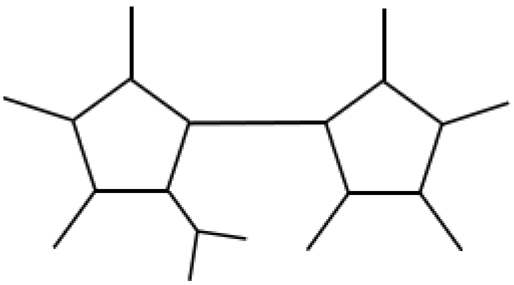
K = 90,720



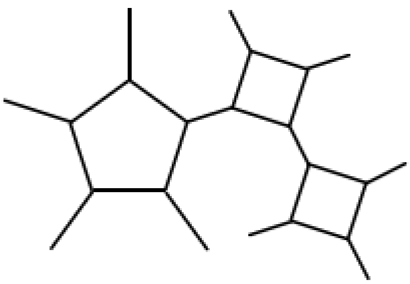
K = 45,360



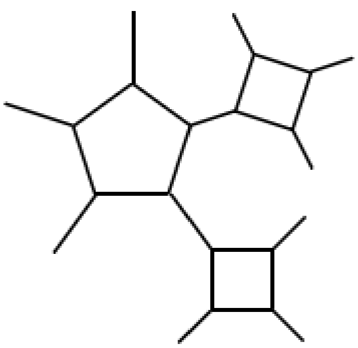
K = 90,720



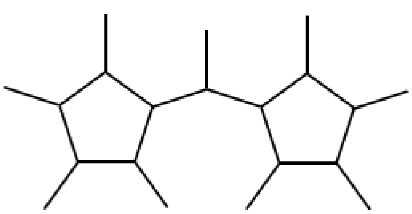
K = 90,720



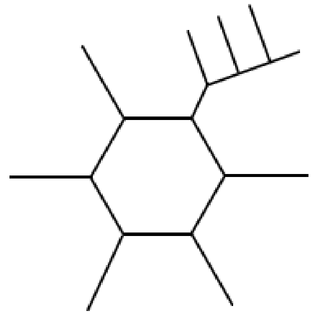
K = 90,720



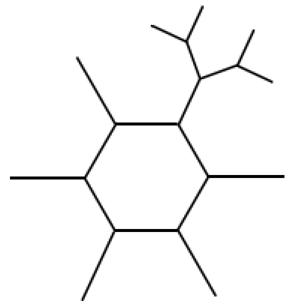
K = 45,360



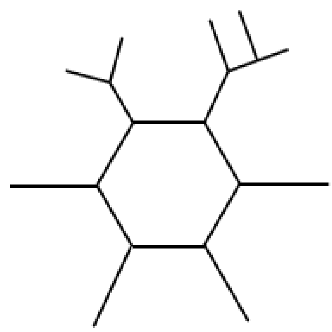
K = 45,360



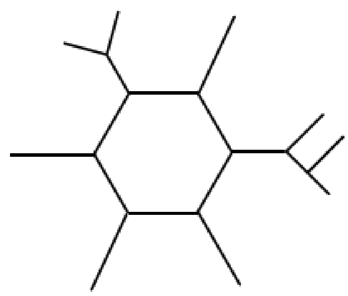
K = 90,720



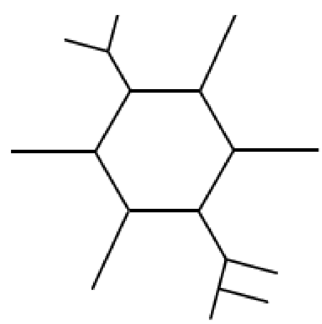
K = 22,680



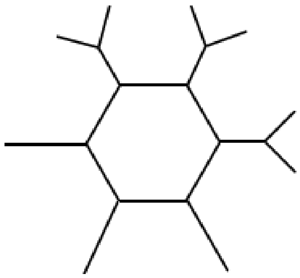
K = 90,720



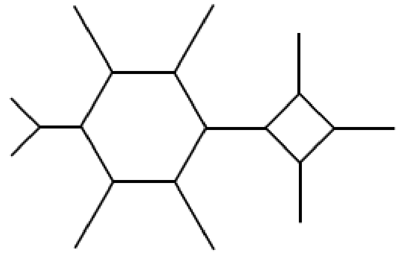
K = 90,720



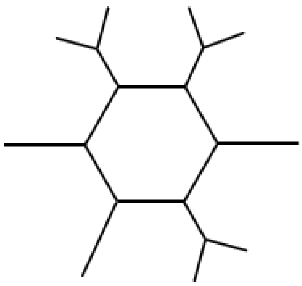
K = 45,360



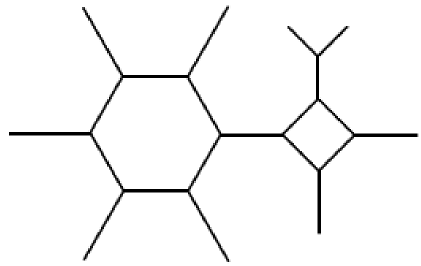
K = 22,680



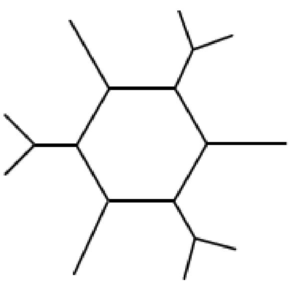
K = 45,360



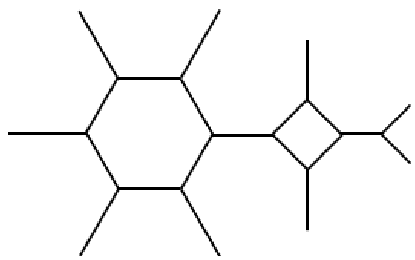
K = 45,360



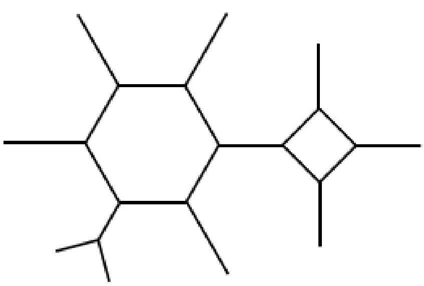
K = 90,720



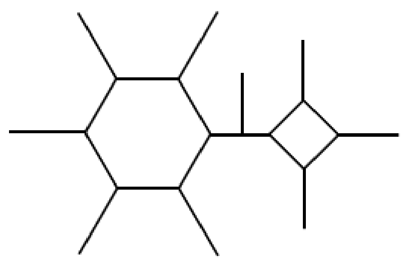
K = 7,560



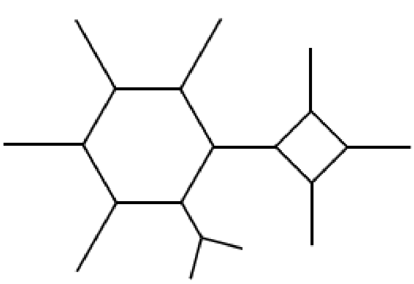
K = 45,360



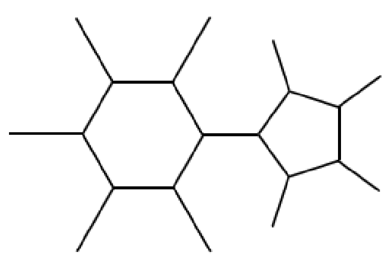
K = 90,720



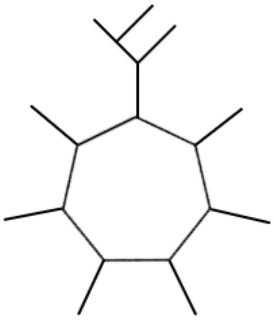
K = 90,720



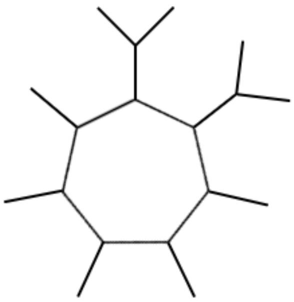
K = 90,720



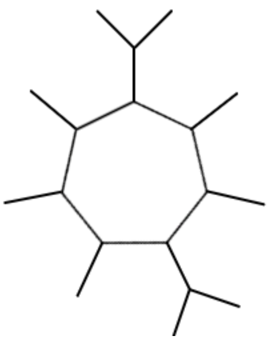
K = 90,720



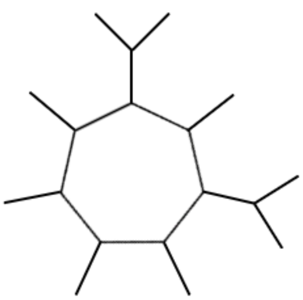
K = 90,720



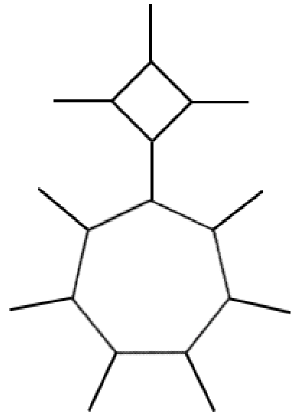
K = 45,360



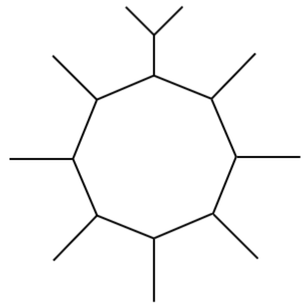
K = 45,360



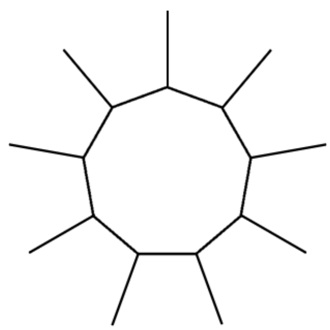
K = 45,360



K = 90,720



K = 90,720



K = 20,160

References

Akers, S. B. (1960). The Use of Wye-Delta Transformations in Network Simplification. Operations Research, 8(3), 311–323.

Arenas, M., Valiente, G., & Posada, D. (2008). Characterization of Reticulate Networks Based on the Coalescent with Recombination. *Molecular Biology and Evolution*, *25*(12), 2517–2520.

Bergeron, F., Labelle, G., & Leroux P. (2013). *Introduction to the Theory of species of Structures*. Université du Québec à Montréal

Curtis, E. B., Ingerman, D., & Morrow, J. A. (1998). Circular planar graphs and resistor networks. Linear Algebra and Its Applications, 283(1), 115–150.

Durell, C., & Forcey, S. (2019). Level-1 phylogenetic networks and their balanced minimum evolution polytopes. *Journal of Mathematical Biology*, *80*(5), 1235.

Forcey, S., Hamerlink, G., Keefe, L., & Sands, W. (2018). The minimum evolution problem in phylogenetics: polytopes, linear programming, and interpretation.

Forcey, S., & Scalzo, D. (2020). Phylogenetic networks as circuits with resistance distance.

Gambette, P., Huber, K. T., & Scholz, G. E. (2017). Uprooted Phylogenetic Networks. *Bulletin of Mathematical Biology: A Journal Devoted to Research at the Junction of Computational, Theoretical and Experimental Biology Official Journal of The Society for Mathematical Biology*, 79(9), 2022.

Huson, D., Rupp, R., & Scornavacca, C. (2010). *Phylogenetic Networks: Concept, Algorithms and Applications*. Cambridge University Press.

Knox, C., & Moradifam, A. (2017). Electrical Networks with Prescribed Current and Applications to Random Walks on Graphs.

Saitou, N., & Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees., *Molecular Biology and Evolution*, 4(4) Jul 1987, Pages 406–425, https://doi.org/10.1093/oxfordjournals.molbev.a040454

Scalzo, D., (2020). Phylogenetic Networks and Functions that Relate Them. *Williams Honors College, Honors Research Project*. 1034

Stanley, R., P. (1999). *Enumerative Combinatorics* (Vol. 2). Cambridge University Press.

Zhang, W., & Sun, Z. (2008). Random local neighbor joining: A new method for reconstructing phylogenetic trees. *Molecular Phylogenetics and Evolution*, 47(1), 117–128.