

20. Counting Trees

1. Introduction

We here consider two kinds of counting problems, and introduce a little algebra to help with them.

If we define some kind of structure, which has some kind of size parameter, N , we can always ask:

How many structures of size N are there?

Here are some questions of this kind:

How many subsets of an M element set are there of cardinality (size) N ?

How many graphs are there on V vertices with N edges?

How many trees are there with N vertices?

How many trees on N vertices have exactly k leaves?

How many bipartite graphs are there on N vertices?

Another kind of question arises when there is some sort of symmetry among the structures we want to consider.

For example, for graphs we can consider all possible permutations of their vertices as symmetry operations.

We can then ask, how many different patterns of structures with given parameter do we have, where two structures have the same pattern when one can be gotten from the other by a symmetry operation.

Thus, for example we can ask: how many different patterns of graphs with N edges on V vertices are there, with permutations of the vertices as symmetries?

Notice that the answer for $N=1$ or 2 is easy to find but quite different from what it is for the first kind of problem

If we ask how many graphs on V vertices are there with 1 edge, the answer is $V(V-1)/2$, because that is the number of different edges of the form (i,j) with $i < j$ that we can form.

On the other hand, they all have the same pattern, that of a single edge.

Similarly there are $(V(V-1)(V(V-1)-1))/2$ different pairs of edges, but only two patterns of them: namely two edges can be either disjoint or they can form a two length path. So the answer to the pattern question for $N=2$ is 2 .

We first look at some ordinary counting problems, then consider how we represent symmetry

operations, and then consider some pattern counting problems.

Of the counting questions listed at the beginning of this section, the first two are straightforward.

The number of subsets of an M element set of size N we have already seen to be the binomial coefficient $C(M,N)$, which is $M!/(N!(M-N)!)$.

By the way there a simpler notation for $M!/(M-N)!$ which is the product of N factors, $M*(M-1)* \dots *(M-N+1)$. It is $M_{(N)}$, which denotes a descending product of N terms starting with M .

Similarly we can write $M*(M+1)* \dots *(M+N-1)$, which is an ascending product of N terms starting with M as $M^{(N)}$.

The number of graphs on V vertices and N edges is the number of ways of picking N edges out of the possible set of $V(V-1)/2$ of them.

Thus, it is the binomial coefficient, $C(V(V-1)/2,N)$ or $(V(V-1)/2)_{(N)}/N!$.

We now ask: How Many trees on N vertices are there?

2. Counting Trees

Before counting trees, let us recall what they are.

A graph on a vertex set is a collection of unordered pairs of vertices, called edges

A tree is a connected graph without cycles. That is, there is a path from any vertex to any other, but no path from a vertex to itself that does not traverse each edge on it an even number of times.

Without edges, the empty graph has $|V|$ connected components. Each edge that can be added to a graph provides a path from one of its ends to the other.

If there already was a path with these ends, so that the ends were in the same connected component without the new edge, then the new edge completes a cycle, and we will not have a tree.

Otherwise, the new edge joins two previously connected components of the graph to which it is added, into one, and so, after $|V|-1$ edges are added to the empty graph, we will have a tree.

Thus every tree on n vertices has $n-1$ edges. We could have define trees as connected graphs with $n-1$ edges, or as graphs with $n-1$ edges without cycles. In other words, any two of the three properties, $n-1$ edges, connected and no cycles implies the third.

We now ask, how many trees are there on n vertices?

You can guess the answer by seeing what it is for small values of n . There is only one tree with two vertices.

With three vertices all trees are paths of length two; there are three of them, namely $12\ 23, 13\ 23$

and 12 13.

With four vertices there are two patterns of trees; a path of length three and a “claw” consisting of one vertex linked to each of the others, as in 12 13 14.

There are 4 claws with each vertex as center; on the other hand there are 12 paths; there are 6 or $C(4,2)$ pairs of endpoints for the paths, and two ways to arrange the intermediate vertices within it, for a total of 16 trees all together.

With 5 vertices there are 3 patterns: a claw, a Y (whose lower part is a path of length two) and a path of length 4.

There are 5 claws, and $C(5,2)*3!$ paths, ($C(5,2)$ possible endpoints and $3!$ ways to arrange the endpoints) and $C(5,2)*3!$ Y's ($C(5,2)$ ways to choose the top vertices of the Y and $3!$ ways to arrange the rest), for a total of 125 trees.

We therefore find, if we define the number of trees on n vertices to be $F(n)$:

$$F(2) = 2^0, F(3) = 3^1, F(4) = 4^2 \text{ and } F(5) = 5^3.$$

This suggests the hypothesis: $F(n) = n^{n-2}$.

3. Proving that the number of trees on n vertices is n^{n-2}

There are at least a dozen different ways to prove this fact.

We will do so by defining other structures whose number we know to be n^{n-2} , and then show that we can assign a unique tree to each of them, and vice versa.

From this argument we conclude that there are at least as many trees as there are our structures, and vice versa, which is equivalent to our claim.

What then does n^{n-2} count?

Suppose we have n objects, O_1, \dots, O_n , and we pick one. There are n ways to do it.

If we throw it back and pick again, independently, a total of $n-2$ times, there will be a total of n^{n-2} ways to do all this.

There is a neat way to describe this. We can give a name to each choice, say x_j to choosing the j -th object. If we define addition for the x 's, we can describe **each potential choice as**

$$(x_1 + x_2 + \dots + x_n),$$

where each term x_j represents an individual choice, and the plus signs each represent the word “or”.

Then repeating this choice $n-2$ times can be represented as

$$(x_1 + x_2 + \dots + x_n)^{n-2},$$

where each monomial obtained by multiplying this out will represent a sequence of $n-2$ choices.

This expression loses some of the information about the choices, namely the ordering in which they are made, but it is quite useful in letting us keep track of how many times a given set of choices can be made, as we shall see.

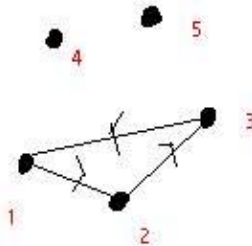
Notice that if we replace all the x_j 's by 1's, our expression counts the number of our possible choices, which is n^{n-2} .

We will now describe a given sequence of choices graphically:

Let $f(j) = x_k$ indicate that we chose our k -th object on our j -th choice.

Then we can draw a directed graph, and put in a directed edge (j,k) from vertex j to vertex k , for each such choice. Remember that we make $n-2$ choices and each can be any object.

In the example pictured below we have $f(1) = 2$, $f(2) = 3$, $f(3) = 1$, and $n=5$. This graph corresponds to a monomial $x_1x_2x_3$.



The resulting graph will have the following properties

1. There will be exactly one edge from each vertex with index up to $n-2$, and none from the last two vertices.
2. It can have directed cycles or even loops.

Our plan is to make each such graph into a tree in a reversible way.

Now a tree is different from one of our graphs in the following respects.

First it is an undirected graph.

We can change this by introducing a direction to each edge, namely toward the last vertex. If we do so every vertex other than the last will have an edge from it.

The difference between our graphs and trees is then the following:

Our graphs have no edge coming from vertex $n-1$ while a directed tree has one.

Our graphs can have loops and directed cycles, trees cannot.

There may be no edge coming into vertex n in one of our graphs, but there must be at least one in every directed tree.

And our graphs have $n-2$ edges while trees have $n-1$ of them.

We will convert one of our graphs into a tree by adding to it a directed path from vertex $n-1$ to vertex n that passes through and destroys every cycle in our graph.

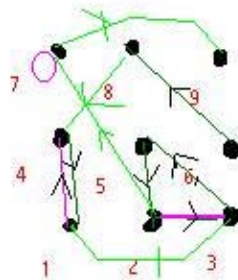
This leaves us with three questions: **how do we order the cycles on this path?**
How do we pass through a cycle to destroy it? And, how do we reverse this process to regain our graph from the tree it creates?

We label each cycle in one of our graphs by **the smallest index** of the vertices in it. (thus the cycle 1 to 3 to 5 to 1 gets the label 1.) Then **we order the cycles by their labels and traverse them in that order.**

We destroy a cycle by entering it immediately after the label vertex and exiting at the label vertex, omitting the edge from the label vertex to its successor in the cycle.

In our example we enter at 3 go to 5 and then 1 and then leave the cycle, omitting the edge (1,3).

So we will have a path that goes from vertex $n-1$ to the first cycle then the next, etc. and finally ends at vertex n .



In the illustration the graph has edges 14, 23, 35, 41, 52, 68, and 77. The path from $n-1$ to n , here from 8 to 9, goes 84135279, so that edges 84, 13, 27 and 79 are added, and 14 23 and 77 are omitted.

The tree then has edges 13, 27, 35, 41, 52, 68, 79 and 84. From the tree we can find the path from $n-1$ to n , here 84135279 again, and deduce that the graph had a cycle 141 another 2352 and a 77 loop, which gives us the graph back again.

Do we really get a tree after introducing this path? Well, there are no cycles left and there are $n-1$ vertices, and that defines a tree.

Notice that the smallest vertex index on the path from $n-1$ to n in the resulting tree will mark the end of the first cycle destroyed, the smallest index beyond it marks the end of the next tree, and so on.

This means that given a tree, we can examine the path in it from vertex $n-1$ to n , find the smallest vertex in it and make it the end of the first cycle, and so on until we have reconstructed our original graph.

And that concludes our proof.

4. How many leaves has an average tree?

By the way, the function $(x_1 + \dots + x_n)^{n-2}$ whose monomials obtained by the distributive law from this expression each correspond to one of our graphs, now has its monomials each corresponding to a tree.

And the monomial that corresponds to a tree retains valuable information about the tree. The power of x_k that occurs on it represents the number of edges that a directed into the k -th vertex in our directed tree, and one less than that for the n -th vertex, since we added an edge directed to it in the path we used to convert a graph to a tree.

This is in every case, one less than the total degree of the k -th vertex in the tree. The difference of one comes from the edge coming out of each of the vertices except for the last two in the graph; and corresponds to an edge added in the tree-making path for the last two vertices.

Thus we conclude that $(x_1 + \dots + x_n)^{n-2}$ is the sum over all trees T on the n vertices of the product over all of the vertices k of $x_k^{d(k,T)-1}$, where $d(k,T)$ is the degree of vertex k in tree T .

If we set all the x_k 's to one we get our formula. But this expression contains lots more information than this.

For example suppose we want the probability that a given vertex, say the n -th, is a leaf of the tree. In trees with this property we have $d(n,T)=1$, or $d(n,T)-1 = 0$, and x_n therefore does not appear in the corresponding monomial.

This means we can count all such trees by setting x_n to 0, and all the other x_k to 1. We get $(n-1)^{n-2}$.

From this fact we can deduce that the proportion of trees for which vertex n is a leaf is

$$\frac{(n-1)}{n} \text{ or } (1 - \frac{1}{n})^{n-2}.$$

For large n $(1-1/n)^n$ is very like $1/e$ so this expression is close to $1/e$ as well. It follows that on the average, a tree on n vertices has

roughly n/e leaves for n reasonably large.

Exercise: 1. What is the expected number of vertices of degree 2 in a tree on n vertices? Of degree k ? give exact expressions and estimates in terms of e . (Hint: differentiate k times and then set $x_n = 0$.)

2. In how many trees are both vertices a and b leaves attached to the same other vertex? What is the expected number of such pairs among all trees? (Pretend the set of all trees is a uniform sample space.)