

16. Counting Trees

16.1 Introduction

We will now consider two kinds of counting problems. The general form of the first kind of question we shall examine is: *If we define some kind of structure, which has size N , how many other structures of size N are there?*

Here are some questions of this form:

1. How many subsets of an M -element set are there of size N ?
2. How many graphs are there on V vertices with N edges?
3. How many trees are there with N vertices?
4. How many trees on N vertices have exactly k leaves?

Another kind of question arises when there is some sort of symmetry among the structures we want to consider. We say that two structures have the same **pattern** when one can be gotten from the other by a symmetry operation (more on this in notes 17). We can then ask how many different patterns of structures we can have with certain given parameters.

Thus, for example we can ask: *how many different patterns of graphs with N edges on V vertices are there?* In this case, since the N edges are in fact just pairs of vertices, we can permute the vertices in these pairs as our symmetry operation. The symmetry comes from the fact that our graph always has N edges; just which vertices they are attached to is what changes.

There is a fundamental difference between these two kinds of questions. One is counting instances of a structure, the other is counting patterns of structures. We shall look at some simple cases involving graphs to illustrate these differences more clearly.

We will 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 the section, the first two are straightforward.

The answer to question 1 is a basic result from combinatorics. The number of subsets of an M element set of size N is the binomial coefficient $\binom{M}{N}$, which is

$$\frac{M!}{N!(M-N)!}.$$

(We can simplify our notation by using the symbol $M_{(N)}$ to denote a descending product of N terms starting with M . Thus,

$$M_{(N)} = M * (M - 1) * \dots * (M - N + 1) = \frac{M!}{(M - N)!}$$

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

To answer question 2, we begin by considering the cases of $N = 1$ and $N = 2$, and comparing the number of graphs we get to the number of patterns we get.

We first look at how many graphs on V vertices there are with 1 edge. Our edge will be between some two vertices in V, so this is a question of picking an unordered set of size 2 from a set of size V. The answer is the binomial coefficient $\binom{V}{2}$. Some simple algebra (or exercise 1 in notes 14) will show you that this is equal to $V(V - 1)/2$. As far as patterns are concerned, all of these graphs have the same pattern, that of a single edge.

Similarly, there are $\frac{V(V - 1)(V(V - 1) - 1)}{2}$ different graphs on V vertices with two edges (see exercises for more info). On the other hand, there are only 2 patterns: either the two edges can be disjoint or they can form a path of length two. So the answer to the pattern question for $N = 2$ is 2. Here is an example of the two patterns for $V = 10$:



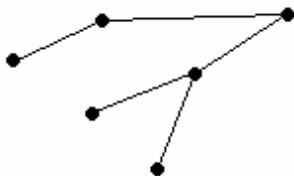
The number of graphs on V vertices and N edges, then, is the number of ways of picking N edges out of the set of all possible edges (which has size $V(V - 1)/2$). Thus, it is the binomial coefficient $\binom{V(V - 1)/2}{N}$, which in our notation from above we can write as $\frac{(V(V - 1)/2)_{(N)}}{N!}$.

We now apply some of these ideas to trees in order to answer questions 3 and 4 (we have some experience with trees from notes 2 and 6).

16.2 Counting Trees

Before looking at trees, let us recall what they are.

A **tree** is a connected graph without any cycles. That is, there is a path from any vertex to any other, but any path from a vertex to itself must traverse every edge contained in it an even number of times. Here is an example of a tree:



We will say that an “empty graph” is a graph on V vertices with no edges. The empty graph has $|V|$ (the size of V) connected components. Each edge that can be added to a graph G provides a path from one of its vertices to the other. If there was already a path between these vertices (so that they were in the same connected component) then the new edge completes a cycle, and we will not have a tree.

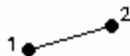
Otherwise, each new edge joins two previously unconnected components of G into one, so that after $|V| - 1$ edges are added to the empty graph, we will have a tree.¹

Thus, every tree on n vertices has $n - 1$ edges. We could have defined 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.

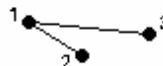
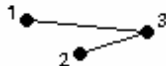
We now ask: *how many trees are there on n vertices?*

We can guess a formula by looking at the answer for small values of n .

It is clear that there is only one tree with two vertices, $\{ (1,2) \}$:

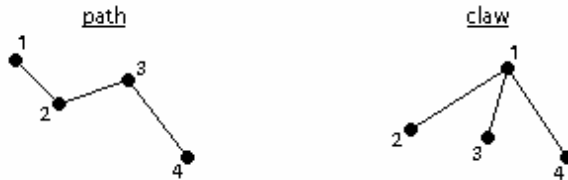


With three vertices, all trees are paths of length two. There are three of them, namely $\{ (1,2), (2,3) \}$, $\{ (1,3), (2,3) \}$, and $\{ (1,2), (1,3) \}$:



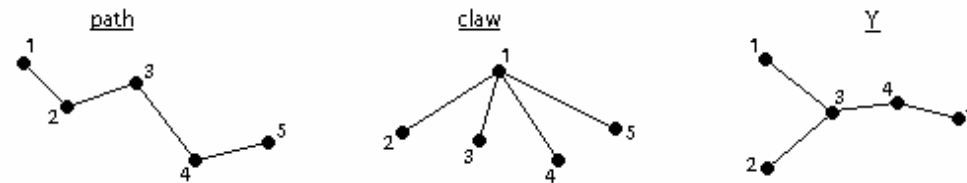
With four vertices, there are two patterns of trees; a path of length three and a “claw” consisting of one vertex linked to each of the others, as in $\{ (1,2), (1,3), (1,4) \}$. Here is an example of one of each pattern:

¹ Thus, we can see that a tree is a graph on V vertices that has the minimum number of edges necessary to be a connected graph.



There are 4 possible claws, one with each vertex as center. For the paths, there are $\binom{4}{2}$, or 6, pairs of endpoints for the paths, and $2!$, or 2, ways to arrange the middle vertices for each of these², giving us 12 possible paths altogether. Adding together the claws and the paths we get a total of 16 possible trees on 4 vertices.

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 possible claws, one for each vertex. There are $\binom{5}{2} * 3!$, or 60, paths, since there are $\binom{5}{2}$ possible pairs of endpoints and $3!$ ways to arrange the intermediate vertices. There are also $\binom{5}{2} * 3!$ Y trees, since there are $\binom{5}{2}$ ways to choose the top vertices of the Y and $3!$ ways to arrange the rest. This gives us $5 + 60 + 60 = 125$ total trees on 5 vertices.

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

$$F(2) = 1 = 2^0, F(3) = 3 = 3^1, F(4) = 16 = 4^2, F(5) = 125 = 5^3$$

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

16.3 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 (don't worry, we're not leaving it as an exercise this time).

² If we have n objects, then there are $n!$ ways to arrange them, because the first one can be any of the n , the second one can be any of the n except the first one, and so on. This means there are n possibilities for the first one, $n-1$ for the second, and so on. This gives us $n*(n-1)*...*1$ possible arrangements, which is $n!$

We will do so by defining other structures whose size we know to be n^{n-2} , and then show that we can assign a unique tree to each of them, and vice versa. This is known as finding a **bijection** between two sets, and is a very common technique in combinatorics.

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 this. If we throw it back and pick again then there are n possible outcomes as well. Thus, if we pick objects independently in this manner, a total of $n-2$ times, there will be n^{n-2} different ways to do this.

There is a useful way to describe this process. We can give a name to each choice; we will say that x_j represents 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 one of the objects, and the plus signs each represent the word "or". Therefore, this expression represents one choice between the n different objects. Repeating this choice $n - 2$ times can be represented as

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

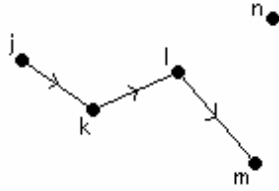
where each term obtained by multiplying this out (called a **monomial**) will represent a sequence of $n - 2$ choices. For example, if $n = 10$, the term $x_1^3 x_2 x_5 x_6 x_{10}^4$ represents choosing x_1 3 times, x_2 once, x_5 once, x_6 once, and x_{10} four times, though not necessarily in this order.

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.

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

We will now describe a given sequence of choices graphically.

In order to do so, we have to make one new definition, that of a directed graph. A **directed graph** has the same elements as a regular graph, namely a set of vertices and edges, with one important difference. In a directed graph, the edges have a direction, meaning that they point from one vertex to the other. Therefore, unlike in a normal graph, when we write the pairs of vertices that represent an edge, the order matters. So the edge (j,k) represents an edge pointing from vertex j to vertex k . Here is a drawing of the directed graph $((j,k), (k,l), (l,m))$ on the vertices $\{j,k,l,m,n\}$:

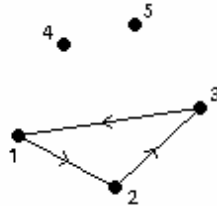


We will use directed graphs to help describe our choices.

Let $f(j) = x_k$ indicate that we chose the k -th object for 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 of the n objects.

In the example pictured below, we have $f(1) = 2, f(2) = 3, f(3) = 1$, and $n = 5$. This graph corresponds to the term $x_1x_2x_3$, or choosing O_1 first, O_2 second, and O_3 third:



The directed graph that we form from our $n - 2$ choices in this way will have the following properties:

1. There will be exactly one edge directed from each vertex with index $\leq n - 2$, and none from the last two vertices. (This is because there are only $n - 2$ choices, and each edge is directed from the vertex of the choice number, and towards the vertex of the object selected).
2. It can have directed cycles or even loops (since one could pick the j -th object on the j -th choice).

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

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

First, a tree is an undirected graph. We can change this by introducing a direction to each edge of the tree, namely towards the last vertex in V . If we do so, every vertex other than the last will have an edge directed from it.

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

- a) Our graphs have no edge directed from vertex $n - 1$ while a directed tree does.
- b) Our graphs can have loops and directed cycles while trees cannot.

- c) There may be no edge directed into vertex n in one of our graphs, but there must be at least one in every directed tree (since every vertex in a tree must have at least one edge, and there is no edge directed from vertex n).
- d) 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 neutralizes (eliminates) every cycle in our graph.

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

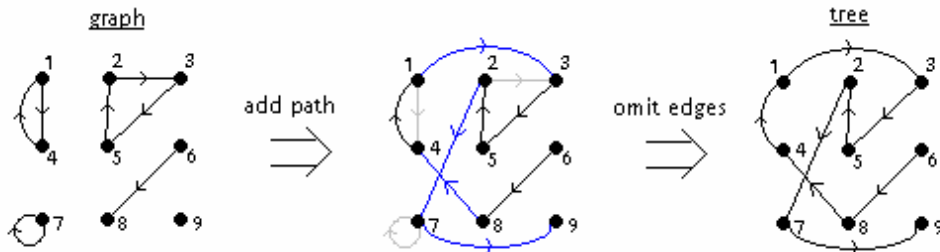
We label each cycle in one of our graphs by the smallest index of the vertices in it. Thus, for example, the cycle $\{ (1,3), (3,5), (5,1) \}$ gets the label 1. We then order the cycles by these labels and traverse them in ascending order.

Here is how we neutralize a cycle. We have our path from $n - 1$ to n enter the cycle at the vertex immediately after the label vertex of the cycle and exit again at the label vertex. We then omit the edge that is directed from the label vertex to the vertex after it in the cycle. In the above example, we enter the cycle at vertex 3 and leave the cycle at vertex 1, omitting the edge $(1,3)$ (We show the cycle-neutralizing path in blue):



So, we will have a path that goes from vertex $n - 1$, neutralizes all the cycles in order, and finally ends at vertex n .

In our next example, the graph has edges $\{ (1,4), (2,3), (3,5), (4,1), (5,2), (6,8), (7,7) \}$. The path from $n - 1$ to n , here from 8 to 9, goes through all the vertices of the graph in this order: 8 4 1 3 5 2 7 9. The edges $(8,4)$, $(1,3)$, $(2,7)$, and $(7,9)$ are added as part of this path and the edges $(1,4)$, $(2,3)$, and $(7,7)$ are omitted in the process of neutralizing the cycles. (In the middle drawing, the added path is in blue and the omitted edges are in grey):



Are we really guaranteed to get a tree after introducing the path from $n - 1$ to n and neutralizing the cycles? Well, if we look at the procedure outlined above, we see that

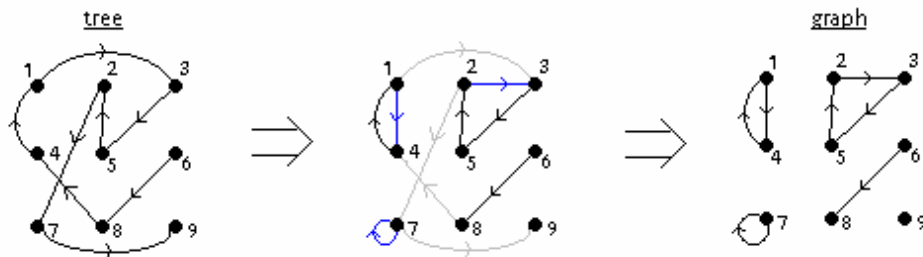
the graph that results has no cycles and $n - 1$ edges, which as we said in the previous section, defines a tree.

Now, how do we get from a tree back to one of our graphs in a well-determined fashion?

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 neutralized. The next smallest index on the path marks the end of the second cycle, and so on.

This means that given a tree, we can examine the path in it from vertex $n - 1$ to n , and find the smallest vertex in it. We know that the first edge of our path connects to the cycle with the smallest index, at the vertex that the label vertex is directed towards. This means, that we can close our first cycle by simply drawing an edge directed from the smallest vertex in the path to the second vertex of the path. We know that the path leaves this cycle and goes to the second cycle, entering it at the vertex right after the label vertex. This tells us which edge to complete this cycle as well. We continue in this manner until we have reconstructed all of the cycles. We then omit every edge of the path that is not in one of our cycles. This gives us back our original graph.

If we look at the tree that we constructed in our last example, we can illustrate this process. We see that this tree has a path going from $n - 1$ to n (in this case from 8 to 9) of the form: 8 4 1 3 5 2 7 9. The smallest vertex on this path is 1. Since the first edge of our path is (8,4), this means that vertex 1 must be directed towards 4, so we add the vertex (1,4). This completes a cycle from 1 to 4 to 1. The next smallest vertex is 2. Since the first edge coming out of the first cycle is (1,3), this means that the next edge we add is (2,3). This completes our second cycle, which goes from 2 to 3 to 5 to 2. We continue in this manner, skipping vertices 2,3,4, and 5 because they are already in cycles, and vertex 6 because it is not on the path. When we finish the process, we omit edges (8,4), (1,3), (2,7), and (7,9) because they are on the path from 8 to 9 but not part of any cycle. In this manner, we get back our original graph:



Thus, we have shown that every set of $n - 2$ choices from n objects can be uniquely represented as a graph that in turn can be uniquely represented as a tree, and vice versa. This gives us our bijection between the number of trees on n vertices and the number of ways to make $n - 2$ choices of n objects, and thus concludes our proof.

□

16.4 How many leaves are on an average tree?

We have shown that each monomial obtained by applying the distributive law to the expression $(x_1 + x_2 + \dots + x_n)^{n-2}$ corresponds to a set of $n - 2$ choices of n objects, which we represented as a graph. The previous section tells us now that each term also corresponds to a tree.

The monomial that corresponds to a tree retains valuable information about the tree. The power of x_k that occurs in it represents the number of edges that are directed into the k -th vertex in our directed tree.³ This holds for every vertex of the tree except vertex n , which has one additional edge directed towards it since we added an edge directed towards it in the path we used to convert the graph into a tree.

Every vertex in the tree has one edge directed away from it, except for the n -th vertex, which has no edges directed away from it. We see, therefore, that the degree of the k -th vertex of the tree is equal to the one plus the power of x_k in the monomial corresponding to the tree (recall that the degree of a vertex is the number of edges connected to it).

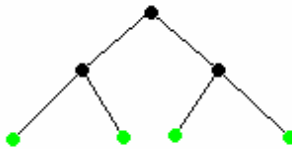
We will say that $d(k, T)$ is the degree of vertex k in tree T . The previous paragraph tells us, then, that the monomial corresponding to T is equal to the product over all vertices k of $x_k^{d(k, T)-1}$. Thus, we conclude that $(x_1 + x_2 + \dots + x_n)^{n-2}$ is equal to the sum over all trees T on n vertices of the product over all of the vertices k of $x_k^{d(k, T)-1}$. We can write this more formally as:

$$(x_1 + \dots + x_n)^{n-2} = \sum_n \left(\prod_k x_k^{d(k, T)-1} \right)$$

If we set each x_k equal to 1, then we get our formula from section 16.2 for the number of trees on n vertices. However, the above expression contains lots more information than this.

For example, we can use it to find the number of leaves for an average tree.

We define a **leaf** of a tree to be a vertex that only has one edge. Thus, for example, the tree below has 4 leaves (here in green):



³ The power of x_k here is the number of times that that object is chosen. In our graph, this corresponds to the number of directed edges that point towards vertex k . When we transform our graph into a tree, for every edge that we remove, we add another edge that is directed towards the vertex that the removed edge was directed towards.

We are looking for the probability that a given vertex, say the j -th, is a leaf of the tree T . Based on our definition of a leaf, if vertex n is a leaf, then $d(j, T) = 1$, meaning that $d(j, T) - 1 = 0$, and therefore x_j does not appear in the corresponding monomial.

This means that we can count all trees where j is a leaf by setting x_j to 0 and all other x_k to 1 in the left side of the above expression. This is just saying that we can pick any combination of the objects, as long as none of them is object j . Because of our bijection, this tells us that there are $(n-1)^{n-2}$ trees where j is a leaf.

We then get that the proportion of trees for which j is a leaf is

$$\frac{(n-1)^{n-2}}{n^{n-2}} = \left(\frac{n-1}{n}\right)^{n-2} = \left(1 - \frac{1}{n}\right)^{n-2}$$

For large n , $\left(1 - \frac{1}{n}\right)^n$, is very close to $\frac{1}{e}$ (see the exercise for more info), so the above expression is close to $\frac{1}{e}$ as well. It follows that on average, a tree on n vertices has roughly $\frac{n}{e}$ leaves, for n reasonably large.

Exercises

Exercise 1 Show that the number of different graphs on V vertices with 2 edges is given by $\frac{V(V-1)(V(V-1)-1)}{2}$.

Exercise 2 Show that for large n , $\left(1 - \frac{1}{n}\right)^n$ is very close to $\frac{1}{e}$.

Hint: One expression for e , is that $e = \lim_{x \rightarrow \infty} \left(1 + \frac{1}{x}\right)^x$.

Exercise 3 What is the expected number of vertices of degree 2 on a tree on n vertices? Of degree k ? Give exact expressions and estimates in terms of e .

Exercise 4 In how many trees are vertices a and b both leaves attached to the same vertex c ? What is the expected number of such pairs among all trees? (Pretend the set of all trees is a uniform sample space).

Additional Sources

Mathworld-A Wolfram Web Resource. <http://mathworld.wolfram.com>

~Edited by Jacob Green