

18.212: Algebraic Combinatorics

Andrew Lin

Spring 2019

This class is being taught by **Professor Postnikov**.

April 5, 2019

Recall that we started talking about the Cayley formula last week: the number of labeled trees on n vertices is n^{n-2} . We gave a proof last time, and we'll review it in more detail today.

Consider the polynomial

$$F_n(x_1, \dots, x_n) = \sum_{T \text{ tree on } [n]} \prod_{i=1}^n x_i^{\deg(i)-1}.$$

The degree of a vertex in any graph is the number of edges adjacent to that vertex. Then we wish to show the following result:

Theorem 1

$$F_n = (x_1 + \dots + x_n)^{n-2}.$$

Proof. We want to show that $R_n = F_n - (x_1 + \dots + x_n)^{n-2}$ is identically zero. We do this by induction: this is true for $n = 1, 2$.

First of all, R_n is either a homogeneous polynomial in our variables x_1, \dots, x_n of degree $n - 2$ or zero. This is because each monomial in the sum has total degree $\sum(\deg(i)) - n$, which is $(2n - 2) - n = n - 2$.

Now, if we pick any $1 \leq i \leq n$ and evaluate R_n at $x_i = 0$, then we have

$$R_{n-1}(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)(x_1 + \dots + x_{i-1} + x_{i+1} + \dots + x_n).$$

(we can denote this "skipping" of x_i as \hat{x}_i as well). This is because all terms disappear except those where i has degree 1, so it's a leaf: this can be constructed by having a labeled tree on the other $n - 1$ vertices and then adding the leaf anywhere!

But the right hand side is 0 by induction, so that means that f when evaluated at $x_i = 0$ is always 0. Now assume that our polynomial f is not equal to zero: then it contains at least one monomial of the form $x_1^{a_1} \dots x_n^{a_n}$. Since the polynomial R_n has degree $n - 2$, $a_1 + \dots + a_n \leq n - 1$: this means there exists an i with $a_i = 0$ in that monomial. But when we evaluate this at $x_i = 0$, that monomial survives, and therefore $R_n \neq 0$: contradiction! So R_n must just be the zero polynomial, which yields what we want.

(Alternatively, x_i is a factor for all i , so R_n must contain a factor $x_1 x_2 \dots x_n$, which has degree larger than R_n unless $R_n = 0$.) □

Corollary 2

Setting all $x_i = 1$, the number of trees on n vertices is n^{n-2} .

We have barely used any properties of trees here, so it's a bit hard to believe! In combinatorics, we like to give a bijective proof, so let's try to construct a bijection between trees and objects whose object is n^{n-2} .

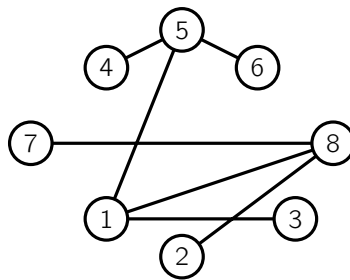
There are several bijective proofs that exist: let's go through a few of them.

Proof by Prufer, 1918. We're going to set up a bijection between trees T on n vertices and sequences (c_1, \dots, c_{n-2}) , where all $c_i \in [n]$. Clearly the latter has n^{n-2} possible elements. This sequence is called the **Prufer code**.

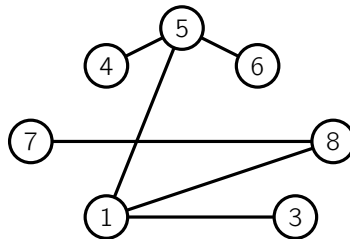
We'll write out the procedure and then do an example.

- Find the leaf (vertex with degree 1) v with the minimal possible label. Any tree with at least 1 vertex contains a leaf.
- Record (in the code) the label that v is attached to in our tree.
- Remove v (and the adjacent edge) from T .
- Repeat these three steps a total of $n - 2$ times!

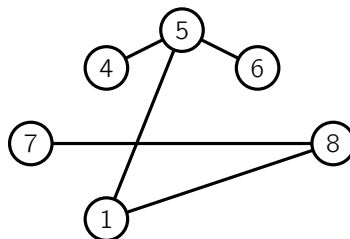
As an example, consider the following tree:



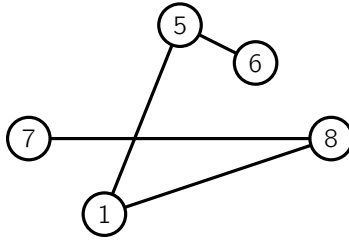
Initially, the smallest leaf has label 2, and it is attached to 8, so we write down $\boxed{8}$: now we erase label 2.



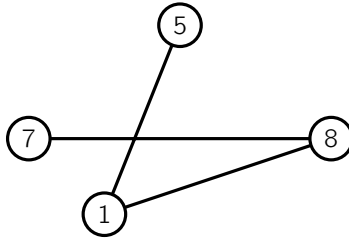
Now, the minimal leaf is 3, and it is attached to 1, so we write down $\boxed{1}$.



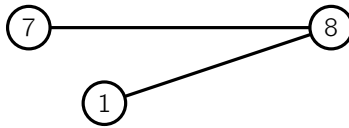
Next, the minimal leaf is 4, and it is attached to $\boxed{5}$:



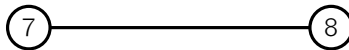
6 is attached to 5:



5 is attached to 1:



Finally, 1 is attached to 8:



and now we're done: our Prufer code is (8, 1, 5, 5, 1, 8). To show this is actually a bijection, we need to be able to invert our construction!

Notice that our vertex 5 has degree 3 originally: it appears twice in the code. Similarly, 1 has degree 3 and appears twice in the code.

Fact 3

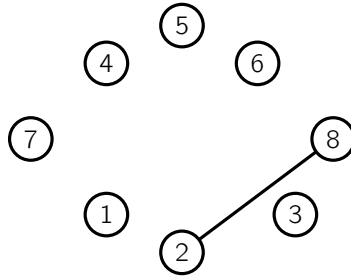
Basically, every vertex appears $\deg(v) - 1$ times in our code!

This is because every time we erase an edge to i , we record i down, except when i is a leaf: then it is removed and its neighbor is recorded instead.

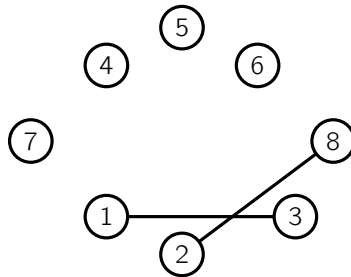
So the labels that don't show up in the Prufer code are exactly the leaves in our tree! Here's the decoding process: let's say we're given (c_1, \dots, c_{n-2}) as our initial Prufer code.

- Initially, set (c_1, \dots, c_{n-2}) to be our **Code**, and set **Labels** to be $\{1, 2, \dots, n\}$.
- Find the minimal leaf by finding the smallest element ℓ of **Labels** that is not in **Code**. Connect ℓ to the first element c of **Code** (because this was the first step of our coding process: we removed the smallest leaf and recorded its neighbor).
- Remove ℓ from **Labels** and c from **Code**.
- Repeat these steps a total of $n - 2$ times until **Code** is empty. Now we've connected everything except our final edge.
- There are now two labels left: connect them!

So let's try doing this in reverse: we start with our Prufer **Code** (8, 1, 5, 5, 1, 8) and **Labels** {1, 2, 3, 4, 5, 6, 7, 8}. The minimal element that doesn't appear is 2, so connect 2 to 8.



Now **Code** is (1, 5, 5, 1, 8) and **Labels** is {1, 3, 4, 5, 6, 7, 8}. Now connect 3 to 1:



Now just continue this process! We connect 4 to 5, and then 6 to 5, and now 5 is connected to 1, 1 is connected to 8, and we finish by connecting 8 and 7. This constructs our original tree, as desired! \square

Notice that this gives us more than just the bijection.

Proposition 4

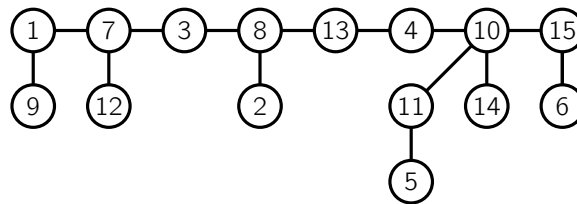
If we fix $d_1, \dots, d_n \geq 1$, then the number of labeled trees on n vertices with degrees d_1, \dots, d_n for vertices $1, 2, \dots, n$ is the number of codes where i appears d_i times:

$$\binom{n-2}{(d_1-1), (d_2-1), \dots, (d_n-1)},$$

as long as $d_1 + \dots + d_n = 2n - 2$.

There are a lot of other combinatorial proofs: here's one more.

Proof by Eğecioğlu-Remmel, 1986, based on Joyal 1981. Let's assume that node 1 is the "root of the tree:" orient all edges of our graph towards the root. Let's use this tree as an example:



\square

The path from our maximal element 15 to 1 is a partial permutation

$$P = 15, 10, 4, 13, 8, 3, 7, 1.$$

Mark all left-to-right minima: this is 15, 10, 4, 3, 1, and now we'll break up our partial permutation into cycles:

$$P = (15)(10)(4, 13, 8)(3, 7)(1).$$

Now erase our path, and replace it with a cycle! So our path now becomes a loop at 15, a loop at 10, a cycle between 4, 13, 8, a cycle between 3 and 7, and a loop at 1.

Notice that we will always have a loop at 1 and a loop at 15. Now notice that there's still exactly one edge coming out of each vertex, so now we can define a function f such that $f(i) = j$ if i points to j . This is a map from $[n]$ to $[n]$ that fixes 1 and n , so there are exactly n^{n-2} such functions.

Turns out this is a bijection between all trees on n nodes and all maps f ! This is a lot like transforming cycle notation to one-line notation and vice versa: we'll discuss this a bit more next time.

MIT OpenCourseWare
<https://ocw.mit.edu>

18.212 Algebraic Combinatorics
Spring 2019

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.