These notes are from https://mathematicaster.org/teaching/graphs2022/prufer.pdf

How many trees are there on a fixed vertex set? For a nonempty set V, let $\mathcal{T}_V$ be the set of all trees on vertex-set V.

Theorem 1 (Cayley's formula)**.** *If* $|V| = n \geq 2$*, then* $|\mathcal{T}_V| = n^{n-2}$.

Phrased differently, the above theorem states that for each $n \geq 2$, the clique K_n has exactly n^{n-2} spanning trees.

Note that we are counting "labeled trees" and not "trees up to isomorphism". For example, there are 3 trees on vertex-set [3], but there is only one tree on three vertices up to isomorphism.[1]

In order to prove this, we will define a bijection from $\mathcal{T}_V$ to V^{n-2} which will imply Cayley's formula since $|V| = n$ and so $|V^{n-2}| = n^{n-2}$. This bijection works by mapping a tree to what is called a Prüfer code. In order to define this, we will need to enforce an ordering on V. To this end, we say that a set V is *ordered* if we can label $V = \{v_1, \dots, v_n\}$ such that $v_1 < \cdots < v_n$ (note that $<$ does not need to be our usual notion of ordering of real numbers; it is simply a way to enforce some ordering on the elements). Note that any finite set can be turned into an ordered set by enforcing some arbitrary order, so we do not lose any generality by looking at ordered sets.

For an ordered set V with $|V| = n \geq 2$, we define a function

$$\text{Prüfer}_V \colon \mathcal{T}_V \to V^{n-2}$$

recursively as follows:

1. If $n = 2$, then define $\text{Prüfer}_V(T) = ()$; the empty-sequence.

2. If $n \geq 3$ and $T \in \mathcal{T}_V$, let $\ell \in V$ be the smallest (with respect to the ordering on V) leaf of T. Let $v \in V$ be the unique neighbor of ℓ in T and define

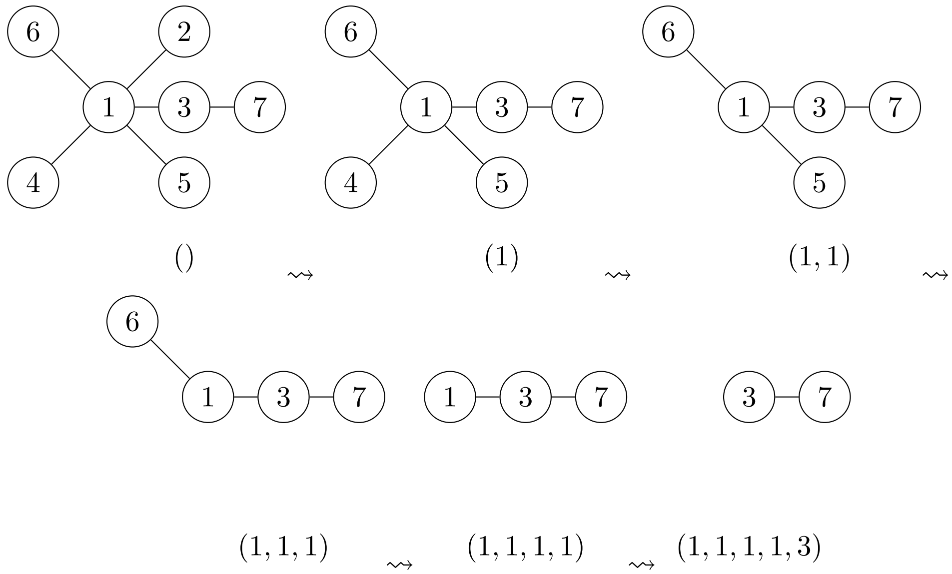
$$\text{Prüfer}_V(T) = \big(v, \text{Prüfer}_{V \setminus \{\ell\}}(T - \ell)\big).$$

Technically speaking, as we just defined things, the output of Prüfer_V would be something like $(2, (2, (1, (4, ()))))$, but we will simply drop parentheses and write $(2, 2, 1, 4)$. With this identification, we see that Prüfer_V is well-defined. For a tree $T \in \mathcal{T}_V$, $\text{Prüfer}_V(T)$ is known as the Prüfer code of T.

Informally, to build the Prüfer code associated with T, we delete the smallest leaf of T, write down the neighbor of that leaf and repeat until there are just two vertices left. Here are two examples (I apologize in advance for my poorly drawn pictures):

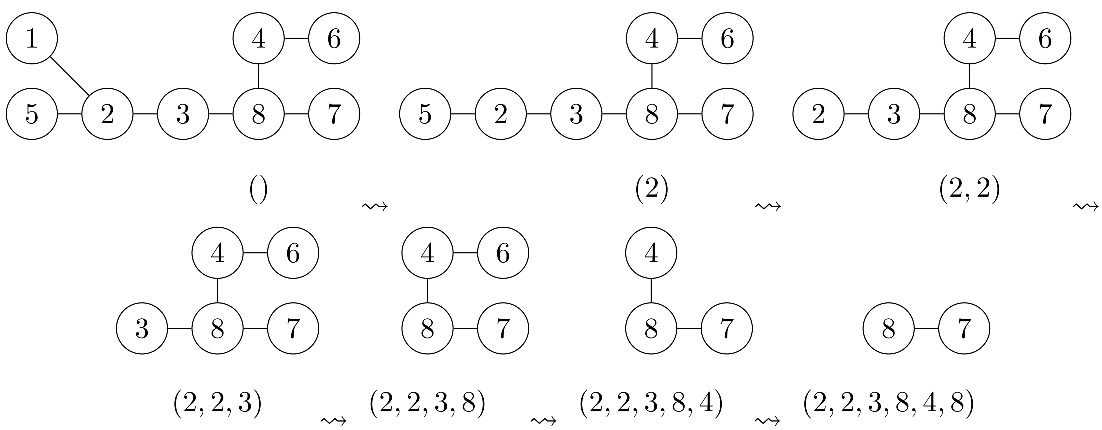
1. Example 1:

[1] This issue gets worse as n gets larger. For example, Cayley's formula states that there are $7^5 = 17807$ trees with vertex-set [7]; however, there are only 11 trees on 7 vertices up to isomorphism. There is no known closed-form expression for the number of trees on n vertices up to isomorphism. See https://oeis.org/A000055 for a list of some known values, though. With other ideas that you may encounter in a general enumerative combinatorics class (Catalan numbers and rooted plane trees), one can show that there are at most $\frac{1}{n+1}\binom{2n}{n} \leq 4^n$ trees on n vertices up to isomorphism (though this is far from tight), which is much, much, much smaller than n^{n-2} for large n. I've included supplementary notes about this topic on the webpage if you're interested in learning more.



Thus, for this tree $T \in \mathcal{T}_{[7]}$, we have $\text{Prüfer}_{[7]}(T) = (1, 1, 1, 1, 3)$.

2. Example 2:



Thus, for this tree $T \in \mathcal{T}_{[8]}$, we have $\text{Prüfer}_{[8]}(T) = (2, 2, 3, 8, 4, 8)$.

We claim that Prüfer_V is a bijection, which will establish Cayley's formula. In order to do so, we will need to understand a little about the structure of $\text{Prüfer}_V(T)$.

Lemma 2. *Fix any ordered set V with $|V| = n \geq 2$. For every $v \in V$ and any $T \in \mathcal{T}_V$, v appears exactly $\deg_T v - 1$ many times in the sequence $\text{Prüfer}_V(T)$.*

Proof. We prove this by induction on n.

The base case of $n = 2$ is clear since the only tree on two vertices is a single edge and its Prüfer code is the empty list.

Now suppose that $n \geq 3$, let V be any ordered set with $|V| = n$ and fix any $T \in \mathcal{T}_V$. Fix any $v \in V$; we need to show that v appears exactly $\deg_T v - 1$ many times in $\text{Prüfer}_V(T)$.

Suppose first that v is the smallest leaf of T; then $\deg_T v = 1$ and v never appears in $\text{Prüfer}_V(T)$. Next, suppose that $\ell \neq v$ is the smallest leaf of T and that u is the unique neighbor of ℓ. Then we have

$$\text{Prüfer}_V(T) = \big(u, \text{Prüfer}_{V \setminus \{\ell\}}(T - \ell)\big).$$

Since $v \neq \ell$, $v \in V \setminus \{\ell\}$ and so, by the induction hypothesis, v appears exactly $\deg_{T-\ell} v - 1$ times in $\text{Prüfer}_{V\setminus\{\ell\}}(T-\ell)$. If $v \neq u$, then $\deg_T v = \deg_{T-\ell} v$ and so we are done. Otherwise, $v = u$ and so $\deg_T v = \deg_{T-\ell} v + 1$, and so we are done since $v = u$ is also the first entry of $\text{Prüfer}_V(T)$. □

With this lemma in hand, we are ready to prove that Prüfer_V is a bijection and thus establish Cayley's formula.

Theorem 3. *For an ordered set V with $|V| = n \geq 2$, Prüfer_V is a bijection from $\mathcal{T}_V$ to V^{n-2}.*

Proof. We prove this by induction on n.

The base case of $n = 2$ is clear. If $|V| = 2$, then there is exactly one tree, and Prüfer_V maps this one tree to the empty sequence, which is the only element of V^0.

Suppose now that $n \geq 3$ and let V be any ordered set with $|V| = n$.

We show first that Prüfer_V is surjective. To this end, fix any sequence $(s_1, \ldots, s_{n-2}) \in V^{n-2}$. Let $v \in V$ be the smallest element with $v \notin \{s_1, \ldots, s_{n-2}\}$. Note that v exists since $|V| = n$ (so there are at least two elements not in $\{s_1, \ldots, s_{n-2}\}$). In particular, $(s_2, \ldots, s_{n-2}) \in (V \setminus \{v\})^{n-3}$. By the induction hypothesis, there is a tree $T' \in \mathcal{T}_{V\setminus\{v\}}$ with $\text{Prüfer}_{V\setminus\{v\}}(T') = (s_2, \ldots, s_{n-2})$. Form the tree T on vertex set V by attaching v to the vertex s_1. By construction, v is a leaf of T and v's unique neighbor is s_1; thus we will have succeeded in showing that $\text{Prüfer}_V(T) = (s_1, \ldots, s_{n-2})$ if we can show that v is the smallest leaf of T. Suppose that u is the smallest leaf of T and suppose for the sake of contradiction that $u \neq v$ (and thus $u < v$). But then u is also a leaf of $T - v$. Thus, by Lemma 2, we know that u never appears in $\text{Prüfer}_{V\setminus\{v\}}(T-v)$. Additionally, $u \neq s_1$ since s_1 cannot be a leaf since $n \geq 3$. Thus, $u \notin \{s_1, \ldots, s_{n-2}\}$; a contradiction to the definition of v.

We show now that Prüfer_V is injective. Suppose that $T, S \in \mathcal{T}_V$ have

$$\text{Prüfer}_V(T) = \text{Prüfer}_V(S) = (v_1, \ldots, v_{n-2});$$

we need to show that $T = S$. Suppose that t is the smallest leaf of T and that s is the smallest leaf of S. Thus, the unique neighbor of t in T is v_1 and the unique neighbor of s in S is v_1. We claim that $t = s$. Since $(v_1, \ldots, v_{n-2}) = \text{Prüfer}_V(T)$, t does not appear in $(v_1, \ldots, v_{n-2})$. Thus, t must be a leaf of S. Since s is the smallest leaf of S, we must then have $t \geq s$. A symmetric argument implies that $s \geq t$ and so $t = s$ as desired.

Now, that we know $t = s$, we have

$$\text{Prüfer}_{V\setminus\{t\}}(T-t) = \text{Prüfer}_{V\setminus\{t\}}(S-t) = (v_2, \ldots, v_{n-2}).$$

The induction hypothesis then implies that $T - t = S - t$. Finally, since $t = s$ is attached to v_1 in both T and in S, this implies that $T = S$ as needed. □

As we've seen, there is a simple way to build the Prüfer code of a tree (this is how we defined the Prüfer code after all). Furthermore, given a sequence $P \in V^{n-2}$, we can construct a tree on vertex set V whose Prüfer code is P since Prüfer_V is bijective. But how would one efficiently build this tree?

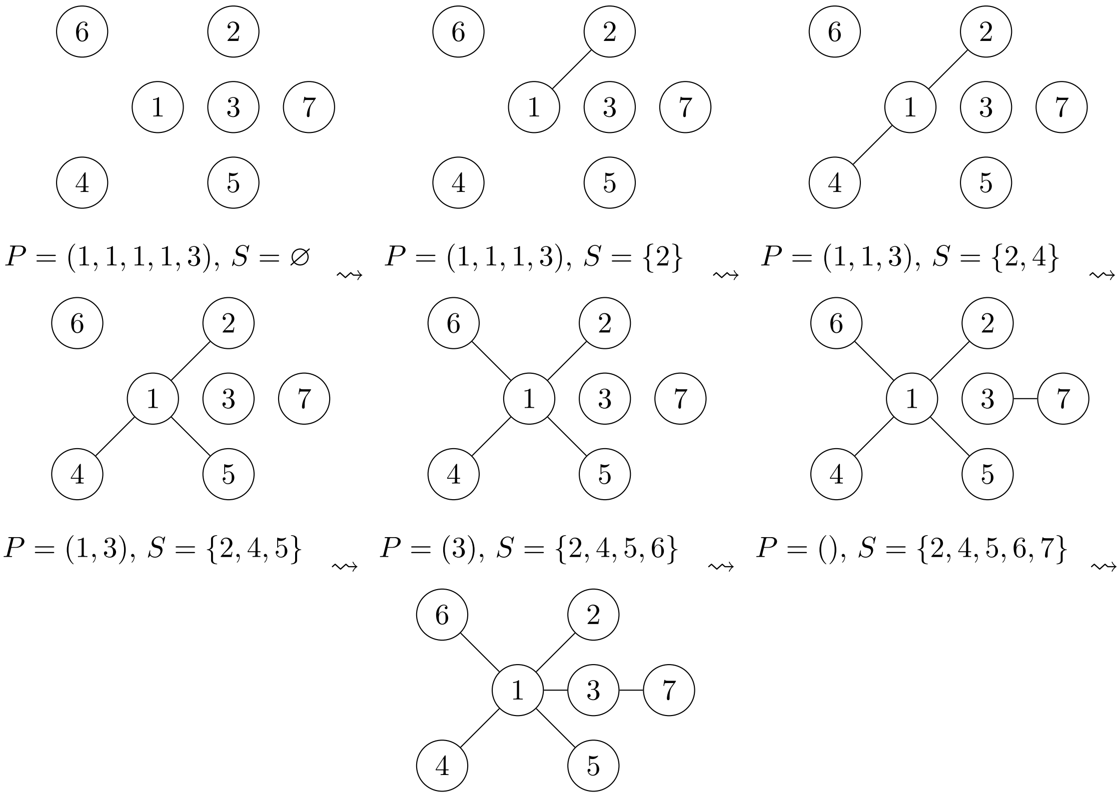
Consider a sequence $P \in V^{n-2}$. Initialize $F = (V, \varnothing)$ and $S = \varnothing$ and iterate the following until P becomes the empty sequence:

Let $v \in V$ be the smallest element which does not appear in S nor in P. Suppose that the first entry in P is p and add the edge vp to F. Remove the first entry of P (so we now have a sequence of length one fewer) and add v to S. Repeat.

Once P is the empty sequence (which happens after $n-2$ iterations), we will have $|S| = n-2$. Thus, $V \setminus S = \{u, v\}$ for some $u \neq v$. Adding the edge uv to F then yields our desired tree.

The proof that this algorithm works follows from unpacking the proof of surjectivity of Prüfer_V. Let's see a couple examples of this algorithm in practice with the two trees from earlier:

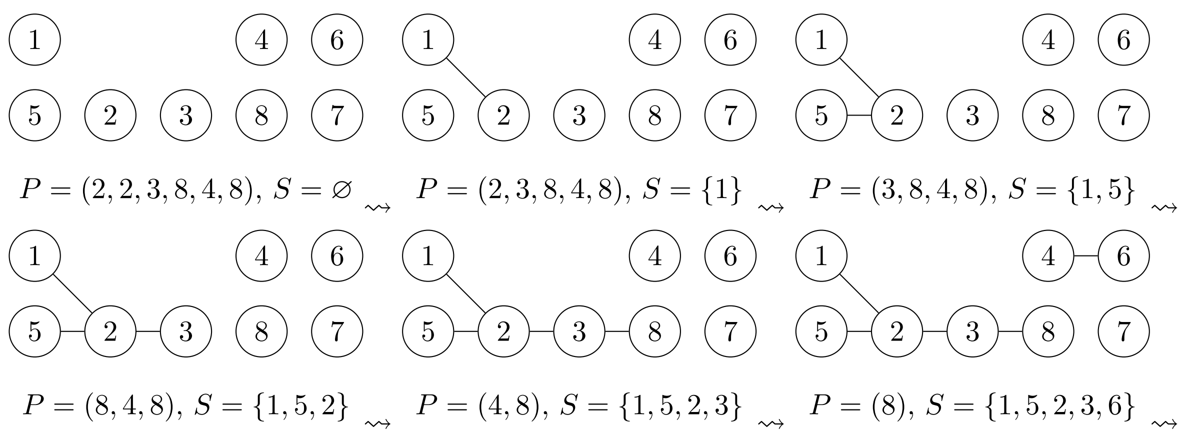
1. Example 1: $P = (1, 1, 1, 1, 3) \in [7]^5$.



$P = (1,1,1,1,3),\ S = \varnothing \rightsquigarrow P = (1,1,1,3),\ S = \{2\} \rightsquigarrow P = (1,1,3),\ S = \{2,4\} \rightsquigarrow$

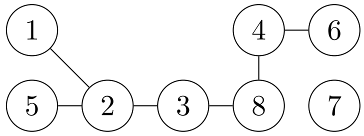
$P = (1,3),\ S = \{2,4,5\} \rightsquigarrow P = (3),\ S = \{2,4,5,6\} \rightsquigarrow P = (),\ S = \{2,4,5,6,7\} \rightsquigarrow$

2. Example 2: $P = (2, 2, 3, 8, 4, 8) \in [8]^6$.



$P = (2,2,3,8,4,8),\ S = \varnothing \rightsquigarrow P = (2,3,8,4,8),\ S = \{1\} \rightsquigarrow P = (3,8,4,8),\ S = \{1,5\} \rightsquigarrow$

$P = (8,4,8),\ S = \{1,5,2\} \rightsquigarrow P = (4,8),\ S = \{1,5,2,3\} \rightsquigarrow P = (8),\ S = \{1,5,2,3,6\} \rightsquigarrow$



$P = (),\ S = \{1, 5, 2, 3, 6, 4\} \rightsquigarrow$