3.7 COUNTING LABELED TREES: PRÜFER ENCODING

In 1875, Arthur Cayley presented a paper to the British Association describing a method for counting certain hydrocarbons containing a given number of carbon atoms. In the same paper, Cayley also counted the number of n-vertex trees with the standard vertex labels $1, 2, \ldots, n$. Two labeled trees are considered the same if their respective edge-sets are identical. For example, in Figure 3.7.1, the two labeled 4-vertex trees are different, even though their underlying unlabeled trees are both isomorphic to the path graph P_4.

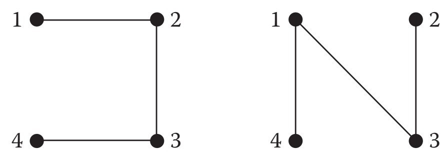


Figure 3.7.1 Two different labeled trees.

The number of n-vertex labeled trees is n^{n-2}, for $n \geq 2$, and is known as ***Cayley's Formula***. A number of different proofs have been given for this result, and the one presented here, due to H. Prüfer, is considered among the most elegant. The strategy of the proof is to establish a one-to-one correspondence between the set of standard-labeled trees with n vertices and certain finite sequences of numbers.

Prüfer Encoding

DEFINITION: A ***Prüfer sequence*** of length $n-2$, for $n \geq 2$, is any sequence of integers between 1 and n, with repetitions allowed.

The following encoding procedure constructs a Prüfer sequence from a given standard labeled tree, and thus, defines a function $f_e : \mathcal{T}_n \to \mathcal{P}_{n-2}$ from the set $\mathcal{T}_n$ of trees on n labeled vertices to the set $\mathcal{P}_{n-2}$ of Prüfer sequences of length $n-2$.

Algorithm 3.7.1: Prüfer Encode
Input: an n-vertex tree with a standard 1-based vertex-labeling.
Output: a Prüfer sequence of length $n-2$.
Initialize T to be the given tree.
For $i = 1$ to $n-2$
Let v be the 1-valent vertex with the smallest label.
Let s_i be the label of the neighbor of v.
$T := T - v$.
Return sequence $\langle s_1, s_2 \ldots, s_{n-2} \rangle$.

Example 3.7.1: The encoding procedure for the tree shown in Figure 3.7.2 is illustrated with the two figures that follow. The first figure shows the first two iterations of the construction, and the second figure shows iterations 3, 4, and 5. The portion of the Prüfer sequence constructed after each iteration is also shown.

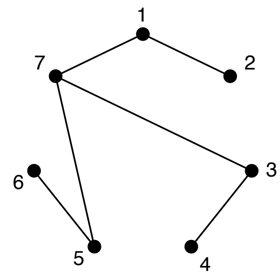


Figure 3.7.2 A labeled tree to be encoded into a Prüfer sequence S.

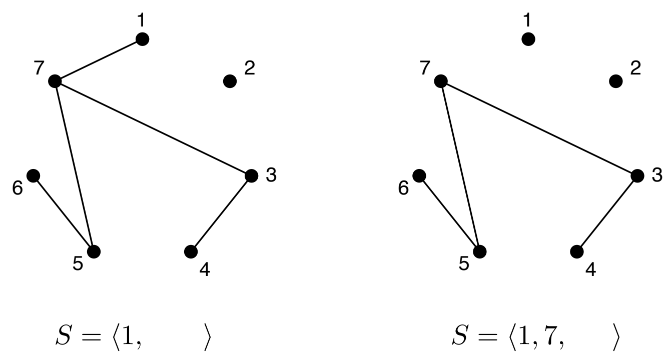


Figure 3.7.3 First two iterations of the Prüfer encoding.

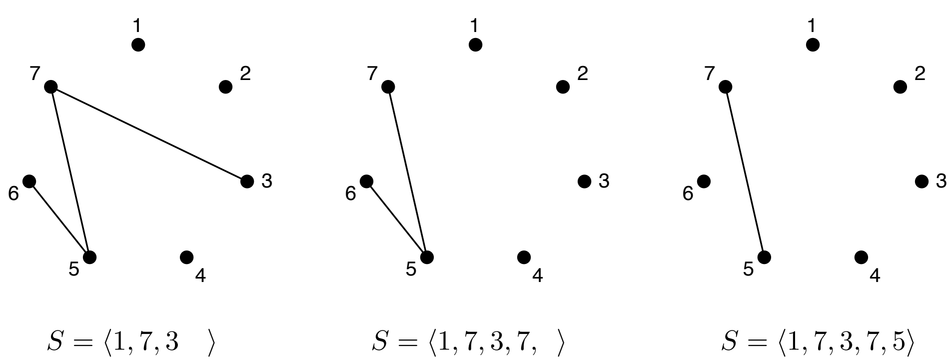


Figure 3.7.4 Iterations 3, 4, and 5 of the Prüfer encoding.

Notice that if we allow the label-set for the n vertices of the tree to be *any* set of positive integers (not necessarily consecutive integers starting at 1), then the encoding proceeds exactly as before. Working with this more general label-set enables us to write the encoding algorithm recursively, shown on the next page, and leads more naturally to the inductive arguments that establish the one-to-one correspondence between labeled trees and Prüfer sequences.

The resulting Prüfer sequence will now need to indicate the label-set.

DEFINITION: A **Prüfer sequence of length $n - 2$ on a label-set** L of n positive integers is any sequence of integers from L with repetitions allowed.

Algorithm 3.7.2: Prüfer Encode (recursive) (T, L)
Input: an n-vertex labeled tree T and its label-set L of n positive integers.
Output: a Prüfer sequence of length $n-2$ of integers from L.
If T is a 2-vertex tree
Return the empty sequence
Else
Let v be the leaf vertex having the smallest label t.
Let s be the label of the neighbor of v
$P :=$ Prüfer Encode$(T - v, L - \{t\})$
Return $\langle s, P \rangle$

Proposition 3.7.1: *Let d_k be the number of occurrences of the number k in a Prüfer encoding sequence for a labeled tree T on a set L. Then the degree of the vertex with label k in T equals $d_k + 1$.*

Proof: The assertion is true for any tree on 2 vertices, because the Prüfer sequence for such a tree is the empty sequence and both vertices in T have degree 1.

Assume that the assertion is true for every n-vertex labeled tree, for some $n \geq 2$ and suppose T is an $(n+1)$-vertex labeled tree. Let v be the leaf vertex with the smallest label, let w be the neighbor of v, and let $l(w)$ be the label of w. Then the Prüfer sequence S for T consists of the label $l(w)$ followed by the Prüfer sequence S^* of the n-vertex labeled tree $T^* = T - v$.

By the inductive hypothesis, for every vertex u of the tree T^*, $deg_{T^*}(u)$ is one more than the number of occurrences of its label $l(u)$ in S^*. But for all $u \neq w$, the number of occurrences of the label $l(u)$ in S^* is the same as in S, and $\deg_T(u) = \deg_{T^*}(u)$. Furthermore, $\deg_T(w) = \deg_{T^*}(w) + 1$, and $l(w)$ has one more occurrence in S than in S^*. Thus, the condition is true for every vertex in T. ♢

Corollary 3.7.2: *If T is an n-vertex labeled tree with label-set L, then a label $k \in L$ occurs in the Prüfer sequence $f_e(T)$ if and only if the vertex in T with label k is not a leaf vertex.*

Prüfer Decoding

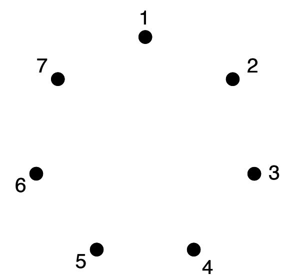
The following decoding procedure maps a given Prüfer sequence to a standard labeled tree.

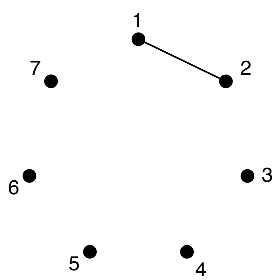
Algorithm 3.7.3: Prüfer Decode (recursive) (P, L)
Input: a Prüfer sequence $P = \langle p_1 p_2 \ldots p_{n-2} \rangle$ on a label-set L of n positive integers.
Output: a labeled n-vertex tree T on label-set L.
If P is the empty sequence (i.e., $n = 2$)
Return a 2-vertex tree on label-set L.
Else
Let k be the smallest number in L that is not in P.
Let P^* be the Prüfer sequence $\langle p_2 \ldots p_{n-2} \rangle$ on label-set $L^* = L - \{k\}$.
Let $T^* =$ Prüfer Decode(P^*, L^*).
Let T be the tree obtained by adding a new vertex with label k and an edge joining it to the vertex in T^* labeled p_1.
Return T.

Example 3.7.2: To illustrate the decoding procedure, start with the Prüfer sequence $P = \langle 1, 7, 3, 7, 5\rangle$ and label-set $L = \{1, 2, 3, 4, 5, 6, 7\}$. Proposition 3.7.1 implies that the corresponding tree has: $\deg(2) = \deg(4) = \deg(6) = 1$; $\deg(1) = \deg(3) = \deg(5) = 2$; and $\deg(7) = 3$. Among the leaf vertices, vertex 2 has the smallest label, and its neighbor must be vertex 1. Thus, an edge is drawn joining vertices 1 and 2. The number 2 is removed from the list, and the first occurrence of label 1 is removed from the sequence. The sequence of figures that follows shows each iteration of the decoding procedure. Shown in each figure are: the edges to be inserted up to that point, the label-set, and the remaining part of the Prüfer sequence.

$P = \langle 1, 7, 3, 7, 5\rangle$

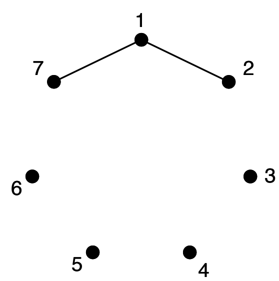
$L = \{1, 2, 3, 4, 5, 6, 7\}$





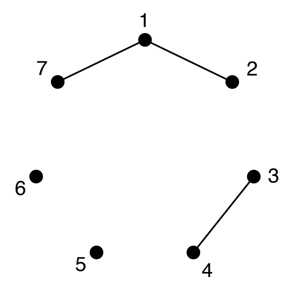
$P = \langle 7, 3, 7, 5\rangle$

$L = \{1, 3, 4, 5, 6, 7\}$



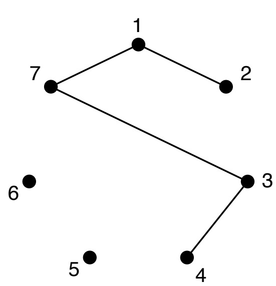
$P = \langle 3, 7, 5\rangle$

$L = \{3, 4, 5, 6, 7\}$



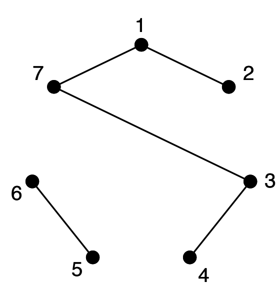
$P = \langle 7, 5\rangle$

$L = \{3, 5, 6, 7\}$



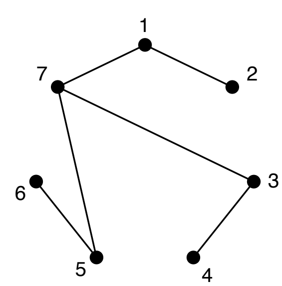
$P = \langle 5\rangle$

$L = \{5, 6, 7\}$



$P = \langle\ \rangle$

$L = \{5, 7\}$



Resulting tree T

Proposition 3.7.3: *For any label-set L of n positive integers, the decoding procedure defines a function $f_d : \mathcal{P}_{n-2} \to \mathcal{T}_n$ from the set of Prüfer sequences on L to the set of n-vertex labeled trees with label-set L.*

Proof: First observe that at each step of the procedure, there is never any choice as to which edge must be drawn. Thus, the procedure defines a function from the Prüfer sequences on L to the set of labeled graphs with label-set L. Therefore, proving the following assertion will complete the proof.

Assertion: When the decoding procedure is applied to a Prüfer sequence of length $n-2$, the graph produced is an n-vertex tree.

The assertion is trivially true for $n = 2$, since the procedure produces a single edge. Assume that the assertion is true for some $n \geq 2$, and consider a label-set L of $n+1$ positive integers and a Prüfer sequence $\langle p_1, p_2, \ldots, p_{n-1} \rangle$ on L. Let k be the smallest number in L that does not appear in $\langle p_1, p_2, \ldots, p_{n-1} \rangle$.

The first call of the procedure creates a new vertex v with label k and joins v to the vertex with label p_1. By the inductive hypothesis, Prüfer Decode($\langle p_2, \ldots, p_{n-1} \rangle, L - \{k\}$) produces an n-vertex tree. Since k is not in the label-set $L - \{k\}$, this tree has no vertex with label k. Therefore, adding the edge from v to the vertex with label p_1 does not create a cycle and the resulting graph is a tree. ♢

Notice that the tree obtained in Example 3.7.2 by the Prüfer decoding of the sequence $\langle 1, 7, 3, 7, 5 \rangle$ is the same as the tree in Example 3.7.1 that was Prüfer-encoded as $\langle 1, 7, 3, 7, 5 \rangle$. This inverse relationship between the encoding and decoding functions holds in general, as the following proposition asserts.

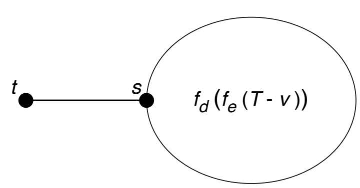
Proposition 3.7.4: *The decoding function $f_d : \mathcal{P}_{n-2} \to \mathcal{T}_n$ is the inverse of the encoding function $f_e : \mathcal{T}_n \to \mathcal{P}_{n-2}$.*

Proof: We show that for any list L of n positive integers, $f_d \circ f_e$ is the identity function on the set of n-vertex labeled trees with n distinct labels from L. We use induction on n.

Let T be a labeled tree on 2 vertices labeled s and t. Since $f_e(T)$ is the empty sequence, $f_d(f_e(T)) = T$.

Assume for some $n \geq 2$, that for n-vertex labeled tree T, $(f_d \circ f_e)(T) = T$. Let T be an $(n+1)$-vertex labeled tree, and suppose v is the leaf vertex with the smallest label t. If s is the label of the neighbor of v, then $f_e(T) = \langle s, f_e(T - v) \rangle$. It remains to show that $f_d(\langle s, f_e(T - v) \rangle) = T$.

By Corollary 3.7.2, the label of every non-leaf vertex appears in $\langle s, f_e(T - v) \rangle$, and since t is the smallest label among the leaf vertices, t is the smallest label that does not appear in $\langle s, f_e(T - v) \rangle$. Therefore, $f_d(\langle s, f_e(T - v) \rangle)$ consists of a new vertex labeled t and an edge joining it to the vertex labeled s in $f_d(f_e(T - v))$ (see figure below).



By the inductive hypothesis, $f_d(f_e(T - v)) = T - v$. Thus, $(f_d \circ f_e)(T)$ consists of the tree $T - v$, a new vertex labeled t and an edge joining that vertex to the vertex labeled s in $T - v$. That is, $(f_d \circ f_e)(T) = T$.

A similar argument shows that $f_e(f_d(P)) = P$, where P is a Prüfer sequence of length $n - 2$ on a label-set L of n positive integers. (See exercises.) ♢

Theorem 3.7.5: [***Cayley's Tree Formula***]. *The number of different trees on n labeled vertices is* n^{n-2}.

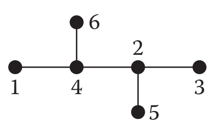
Proof: By Proposition 3.7.4, $f_e \circ f_d : \mathcal{P}_{n-2} \to \mathcal{P}_{n-2}$ and $f_d \circ f_e : \mathcal{T}_n \to \mathcal{T}_n$ are both identity functions, and hence, f_d and f_e are both bijections. This establishes a one-to-one correspondence between the trees in $\mathcal{T}_n$ and the sequences in $\mathcal{P}_{n-2}$, and, by the Rule of Product, there are n^{n-2} such sequences. ♢

Remark: A slightly different view of Cayley's Tree Formula gives us the number of different spanning trees of the complete graph K_n. The next chapter is devoted to spanning trees.

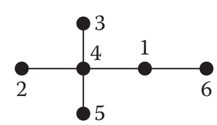
EXERCISES for Section 3.7

In Exercises 3.7.1 through 3.7.6, encode the given labeled tree as a Prüfer sequence. Then decode the resulting sequence, to demonstrate that Proposition 3.7.4 holds.

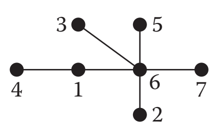
3.7.**1**$^{\mathbf{S}}$



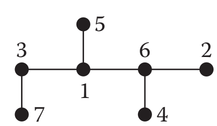
3.7.**2**



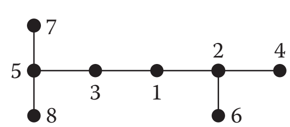
3.7.**3**



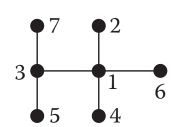
3.7.**4**



3.7.**5**



3.7.**6**$^{\mathbf{S}}$



In Exercises 3.7.7 through 3.7.12, construct the labeled tree corresponding to the given Prüfer sequence.

3.7.**7** $(6, 7, 4, 4, 4, 2)$.

3.7.**8** $(2, 1, 1, 3, 5, 5)$.

3.7.**9** $(1, 3, 7, 2, 1)$.

3.7.**10**$^{\mathbf{S}}$ $(1, 3, 2, 3, 5)$.

3.7.**11** $(1, 1, 5, 1, 5)$.

3.7.**12** $(1, 1, 5, 2, 5)$.