6. Prim's algorithm (from [228]) provides another method for finding minimum weight spanning trees.

   Prim's Algorithm

   Given: A connected, weighted graph G.

   i. Choose a vertex v, and mark it.
   ii. From among all edges that have one marked end vertex and one unmarked end vertex, choose an edge e of minimum weight. Mark the edge e, and also mark its unmarked end vertex.
   iii. If every vertex of G is marked, then the set of marked edges forms a minimum weight spanning tree. If not, repeat step ii.

   Use Prim's algorithm to find minimum weight spanning trees for the graphs in Figure 1.44. As you work, compare the stages to those of Kruskal's algorithm.

7. Give an example of a connected, weighted graph G having (i) a cycle with two identical weights, which is neither the smallest nor the largest weight in the graph, and (ii) a unique minimum weight spanning tree which contains exactly one of these two identical weights.

1.3.4 Counting Trees

As for everything else, so for a mathematical theory: beauty can be perceived but not explained.

— Arthur Cayley [214]

In this section we discuss two beautiful results on counting the number of spanning trees in a graph. The next chapter studies general techniques for counting arrangements of objects, so these results are a sneak preview.

Cayley's Tree Formula

Cayley's Tree Formula gives us a way to count the number of different labeled trees on n vertices. In this problem we think of the vertices as being fixed, and we consider all the ways to draw a tree on those fixed vertices. Figure 1.45 shows three different labeled trees on three vertices, and in fact, these are the only three.

There are 16 different labeled trees on four vertices, and they are shown in Figure 1.46.

As an exercise, the ambitious student should try drawing all of the labeled trees on five vertices. The cautious ambitious student might wish to look ahead at Cayley's formula before embarking on such a task.

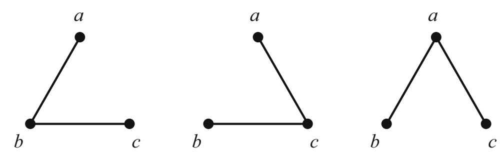

FIGURE 1.45. Labeled trees on three vertices.

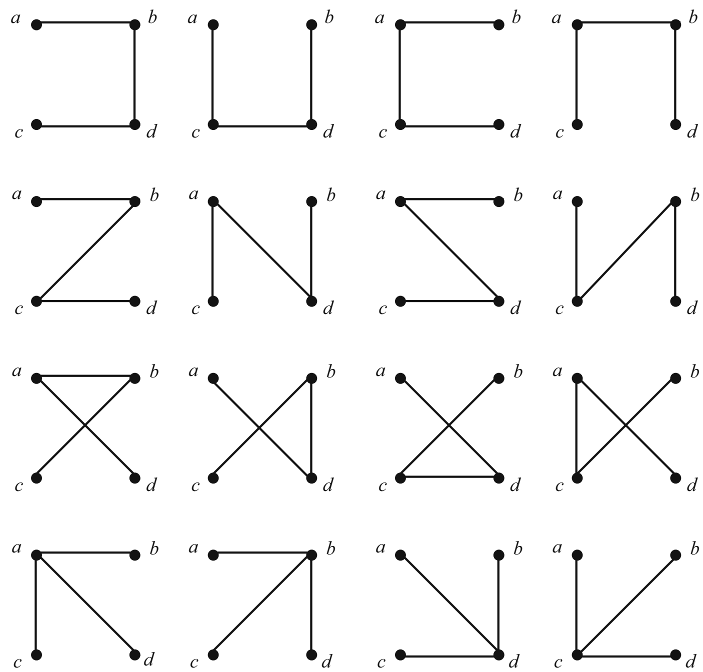

FIGURE 1.46. Labeled trees on four vertices.

Cayley proved the following theorem in 1889 [50]. The proof technique that we will describe here is due to Prüfer[7] [229]. Prüfer's method is almost as noteworthy as the result itself. He counted the labeled trees by placing them in one-to-one correspondence with a set whose size is easy to determine—the set of all sequences of length $n-2$ whose entries come from the set $\{1, \ldots, n\}$. There are n^{n-2} such sequences.

Theorem 1.18 (Cayley's Tree Formula)**.** *There are n^{n-2} distinct labeled trees of order n.*

The algorithm below gives the steps that Prüfer used to assign a particular sequence to a given tree, T, whose vertices are labeled $1, \ldots, n$. Each labeled tree is assigned a unique sequence.

[7]With a name like that he was destined for mathematical greatness!

Prüfer's Method for Assigning a Sequence to a Labeled Tree

Given: A tree T, with vertices labeled $1, \ldots, n$.

1. Let $i = 0$, and let $T_0 = T$.
2. Find the leaf on T_i with the smallest label and call it v.
3. Record in the sequence the label of v's neighbor.
4. Remove v from T_i to create a new tree T_{i+1}.
5. If $T_{i+1} = K_2$, then stop. Otherwise, increment i by 1 and go back to step 2.

Let us run through this algorithm with a particular graph. In Figure 1.47, tree $T = T_0$ has 7 vertices, labeled as shown. The first step is finding the leaf with smallest label: This would be 2. The neighbor of vertex 2 is the vertex labeled 4. Therefore, 4 is the first entry in the sequence. Removing vertex 2 produces tree T_1. The leaf with smallest label in T_1 is 4, and its neighbor is 3. Therefore, we put 3 in the sequence and delete 4 from T_1. Vertex 5 is the smallest leaf in tree $T_2 = T_1 - \{4\}$, and its neighbor is 1. So our sequence so far is 4, 3, 1. In $T_3 = T_2 - \{5\}$ the smallest leaf is vertex 6, whose neighbor is 3. In $T_4 = T_3 - \{6\}$, the smallest leaf is vertex 3, whose neighbor is 1. Since $T_5 = K_2$, we stop here. Our resulting sequence is 4, 3, 1, 3, 1.

Notice that in the previous example, none of the leaves of the original tree T appears in the sequence. More generally, each vertex v appears in the sequence exactly $\deg(v) - 1$ times. This is not a coincidence (see Exercise 1). We now present Prüfer's algorithm for assigning trees to sequences. Each sequence gets assigned a unique tree.

Prüfer's Method for Assigning a Labeled Tree to a Sequence

Given: A sequence $\sigma = a_1, a_2, \ldots, a_k$ of entries from the set $\{1, \ldots, k+2\}$.

1. Draw $k+2$ vertices; label them $v_1, v_2, \ldots, v_{k+2}$. Let $S = \{1, 2, \ldots, k+2\}$.
2. Let $i = 0$, let $\sigma_0 = \sigma$, and let $S_0 = S$.
3. Let j be the smallest number in S_i that does not appear in the sequence σ_i.
4. Place an edge between vertex v_j and the vertex whose subscript appears first in the sequence σ_i.
5. Remove the first number in the sequence σ_i to create a new sequence σ_{i+1}. Remove the element j from the set S_i to create a new set S_{i+1}.
6. If the sequence σ_{i+1} is empty, place an edge between the two vertices whose subscripts are in S_{i+1}, and stop. Otherwise, increment i by 1 and return to step 3.

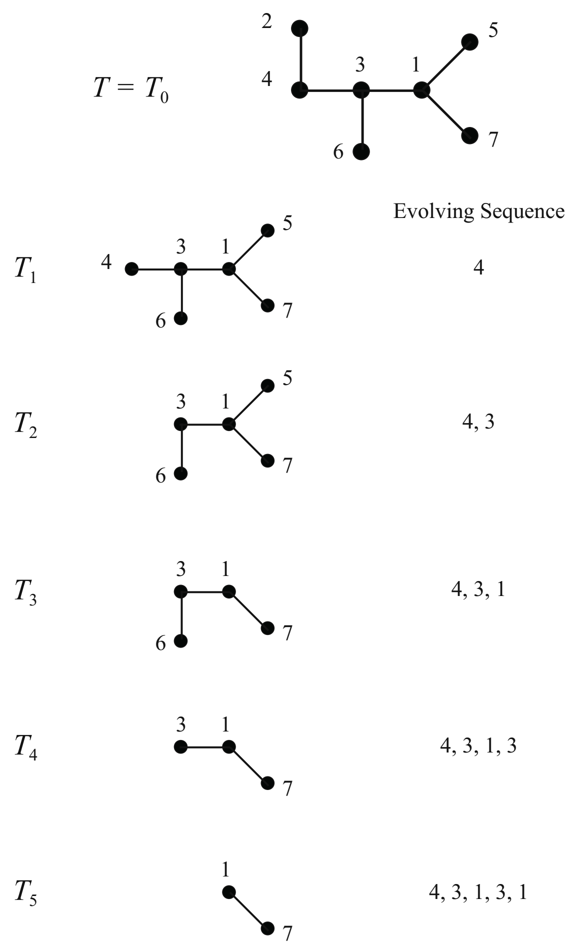

FIGURE 1.47. Creating a Prüfer sequence.

Let us apply this algorithm to a particular example. Let $\sigma = 4, 3, 1, 3, 1$ be our initial sequence to which we wish to assign a particular labeled tree. Since there are five terms in the sequence, our labels will come from the set $S = \{1, 2, 3, 4, 5, 6, 7\}$. After drawing the seven vertices, we look in the set $S = S_0$ to find the smallest subscript that does not appear in the sequence $\sigma = \sigma_0$. Subscript 2 is the one, and so we place an edge between vertices v_2 and v_4, the first subscript in the sequence. We now remove the first term from the sequence and the label v_2 from the set, forming a new sequence $\sigma_1 = 3, 1, 3, 1$ and a new set $S_1 = \{1, 3, 4, 5, 6, 7\}$. The remaining steps in the process are shown in Figure 1.48.

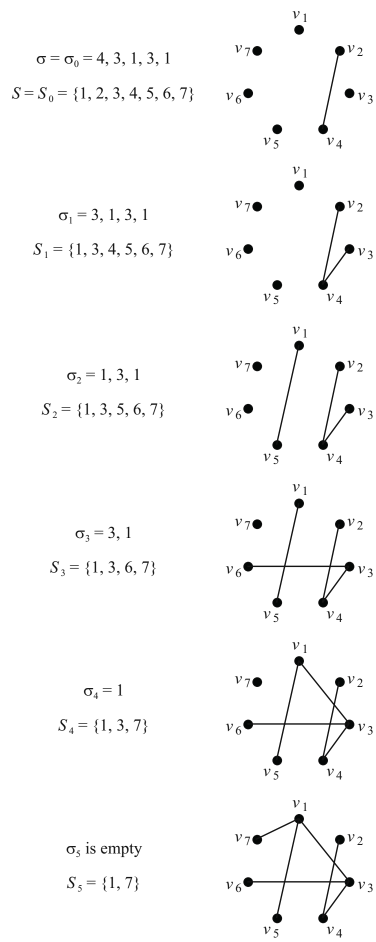

FIGURE 1.48. Building a labeled tree.

You will notice that the tree that was created from the sequence σ in the second example is the very same tree that created the sequence σ in the first example. Score one for Prüfer!

Matrix Tree Theorem

The second major result that we present in this section is the Matrix Tree Theorem, and like Cayley's Theorem, it provides a way of counting spanning trees of labeled graphs. While Cayley's Theorem in essence gives us a count on the number of spanning trees of complete labeled graphs, the Matrix Tree Theorem applies to labeled graphs in general. The theorem was proved in 1847 by Kirchhoff [175], and it demonstrates a wonderful connection between spanning trees and matrices.