7.1 Tree Enumeration

Cayley's Formula states that with a vertex set of size n there are n^{n-2} possible trees. What this means is that there is n^{n-2} ways to form a list of length $n-2$ with entries created from a given vertex set. A list for a specific tree is its **Prüfer code**. For a given tree T with some logical ordering of vertex identifiers, we can create its **Prüfer code** by first deleting the lowest value leaf and outputting that leaf's neighbor as a value in our code. We can also use a given vertex set S and a **Prüfer code** to recreate T. See the proof in the book that for a set $S \in \mathbb{N}$ of size n, there are n^{n-2} trees with vertex set S.

Below are the algorithms for creating a **Prüfer code** from T and recreating T from a **Prüfer code**.

---

procedure CREATEPRUFER(Tree T with vertex set S)
    $a \leftarrow \emptyset$ ▷ Initialize Prüfer code to null
    for $i = 1 \ldots (n-2)$ **do**
        $l \leftarrow$ least remaining leaf in T
        $T \leftarrow (T - l)$
        $a_i \leftarrow$ remaining neighbor of l in T
    return a

---

procedure RECREATETREE(Prüfer code a created with vertex set S)
    $V(T) \leftarrow S$ ▷ Tree has vertex set S
    $E(T) \leftarrow \emptyset$ ▷ Initialize tree edges as empty
    initialize all vertices in S as unmarked
    for $i = 1 \ldots (n-2)$ **do**
        $x \leftarrow$ least unmarked vertex in S not in $a_{i \ldots (n-2)}$
        mark x in S
        $E(T) \leftarrow (x, a_i)$
    $x, y \leftarrow$ remaining unmarked vertices in S
    $E(T) \leftarrow (x, y)$
    return T

How many different ways can we create a graph given a vertex set of size n? **Cayley's Formula** states that with a vertex set of n there are n^{n-2} possible trees. A **spanning subgraph** of some graph G is a subgraph that contains all vertices in G. A **spanning tree** is a spanning subgraph that is also a tree. Another way to think about Cayley's Formula: the number of possible trees is the number of possible spanning tree configurations of complete graph. How might we compute the number of spanning trees of a general graph?

We can use a simple recurrence relation to do so. The number of possible spanning trees in a graph $\tau(G)$ is equal to the sum of the number of spanning trees of the graph with an edge removed $\tau(G - e)$ plus the the number of spanning trees of the graph with an edge contracted $\tau(G \cdot e)$. An **edge contraction** involves combining the endpoints u, v of a given edge e into a single vertex, such that the new vertex has incident edges of all original edges of both u and v except for e. We'll see more on recurrence relations in the future.

7.2 Graceful Labeling

A **graceful labeling** of a graph is a labeling of all n vertices of a graph with unique labels from $0 \ldots m$, such that each of the m edges has a unique value computed as the difference between the labels of its endpoints. A graph is **graceful** if it has a graceful labeling.

Ringel-Kotzig Conjecture: all trees are graceful. This is unproven, however, certain subsets of of trees have been proven to be. These include paths and **caterpillar graphs**. Caterpillar graphs are trees in which a single path is incident to or contains every edge in the graph. Proving this conjecture will guarantee you an A in this course (though I get to be a co-author on the paper).