A *Prüfer sequence* is a sequence of $n-2$ numbers, each being one of the numbers 1 through n. The set of these sequences is denoted by P_n. The set number of trees with n labelled vertices is denoted by T_n.

Theorem 3.9. *(Cayley's Formula) The number of trees with n labelled vertices is* n^{n-2}.

Proof. We are going to find a bijection between the set of Prüfer sequences and the set of spanning trees. The following is an algorithm that can be used to encode any tree into a Prüfer sequence:
1. Take any tree, $T \in T_n$, whose vertices are labeled from 1 to n in any manner.
2. Take the vertex with the smallest label whose degree is equal to 1, delete it from the tree and write down the value of its only neighbor. (Note: above we showed that any tree must have at least two vertices of degree 1.)
3. Repeat this process with the new, smaller tree. Continue until only one vertex remains.
This algorithm will give us a sequence of $n-1$ terms. Since we already know the number of vertices in our graph by the length of our sequence, we can drop the last term as it is redundant. So now we have a sequence of $n-2$ elements encoded from our tree.
(every vertex has degree equal to 1 + a, where a is the number of times that vertex appears in our sequence.)

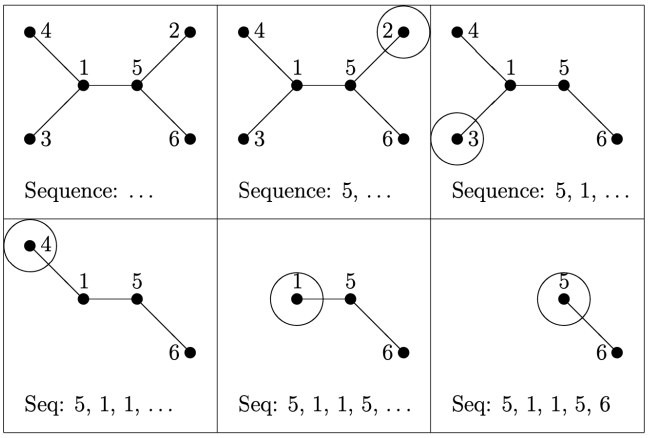


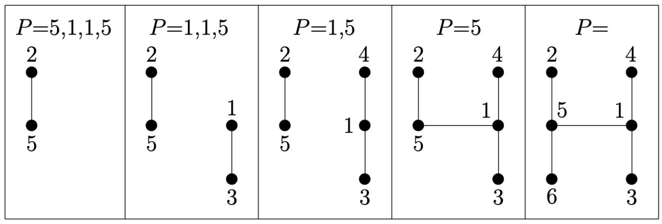
Fig. 3.1. Encoding of a tree by Prüfer sequence

Each tree gives a unique Prüfer sequece (Reconstruction of tree from a Prüfer sequece) : Given a Prüfer sequence, let R be the list of numbers from 1 to n (in increasing order). Draw n vertices with labels 1 to n. Then perform the following steps.
1. Draw an edge between the vertex labeled by the first number in R that is NOT in P and the vertex labeled by the first number in P.
2. Update R by removing its first number (the minimum) and also update P by removing its first number. Then repeat step 1.
3. Do this until there is no number left in P. Finally, remember to connect the remaining isolated vertex (only one, why?) with the vertex labelled with the last number of the orginal sequence P.

So, we have a bijective function between $T_n \to P_n$. Since, there are n^{n-2} Prüfer sequence for any given n, so $|P_n| = n^{n-2}$, and thus $|T_n| = n^{n-2}$.

Tree reconstruction illustrated: click here

Let's reconstruct our original tree from our sequence, $P = 5, 1, 1, 5$:



□

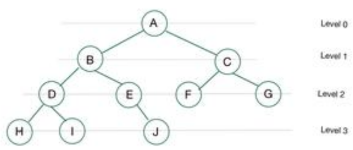
Example 3.10. *Supplementary materials on Prüfer sequence.* click here

Corollary 3.11. *The number of spanning tree in a complete graph with n labelled vertices is n^{n-2}.*

Definition 3.12. *An edge in a spanning tree T is called a **branch** of the spanning tree T. If an edge of a connected graph G is not a branch of a spanning tree T of G then the edge is called a **chord** of T. A fundamental circuit is obtained by adding a chord to a spanning tree.*

3.1. Binary trees.

Definition 3.13. *A rooted tree is a tree in which one vertex has been designated the root. The level of a vertex in a rooted tree refers to its distance from the root. Two ends of an edge clearly lies in two consecutive level, i.e., if lower level end of an edge has level k then the other end has level $k + 1$. The former vertex is called the parent vertex and the later its child. In the context of rooted trees, the term "node" is more in usage than "vertex" in the literatures. And Pendant vertices in a tree are also called leaves. An internal node/vertex is any vertex of a rooted tree that has child nodes.*
A binary tree is a rooted tree whose vertices has degree at most 3 and root has degree exactly 2.



Various types of Binary trees

- **Full Binary Tree:** A binary tree where each node has either 0 or 2 children.
- **Complete Binary Tree:** A complete binary tree is a binary tree in which every level, except possibly the last, is completely filled, and all nodes in the last level are as far left as possible

- **Perfect Binary Tree:** A binary tree in which all the internal nodes have two children, and all leaf nodes are at the same level.

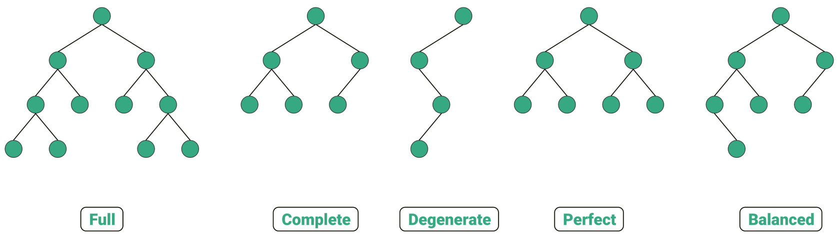


Fig. 3.2. Various types of binary trees

- **Height of the tree:** The height of a rooted tree is the number of edges in the longest path from the root to a leaf. A tree with only one vertex has a height of 0.
- **Number of vertex:** A binary tree of height h has at most $2^{h+1}-1$ nodes.
- **Number of leaves:** In a full binary tree, the number of leaves is $L = \frac{N+1}{2}$, where N is the total number of nodes.

Theorem 3.14. *The number of leaves in a binary tree with n vertices is at least 2 and at most* $\lceil \frac{n}{2} \rceil$

Proof. The minimum number is guaranteed by Theorem 3.4.

Let T denote a tree with n vertices. Let $I(T)$ and $L(T)$ denote the set of internal vertices and leaves of T, respectively.
Then

$$|I(T)| + |L(T)| = |T| = n \tag{3.1}$$

We will use a double counting argument. Let S denote the set of pair of vertices (v_p, v_c), where v_p is parent of v_c. Since in a binary tree each internal vertex has at most 2 child,

$$|S| \leq 2|I(t)|$$

Further, each child has one parent except the root; consequently,

$$|S| = n - 1$$

Equating the above two counts of $|S|$, we obtain, $|I(T)| \geq \frac{n-1}{2}$. It follows that

$$|L(T)| \leq n - \frac{n-1}{2} = \frac{n+1}{2}$$

Since number of leaves can only be an integer, hence the theorem.

□

3.2. **Practice problem set 3.** TBA