Trees

A tree is a mathematical structure that contains not only a set of data elements, but also connections between elements, giving a tree-like appearance. Trees were first studied by Cayley (1857). We call these data elements with connection information **nodes**.

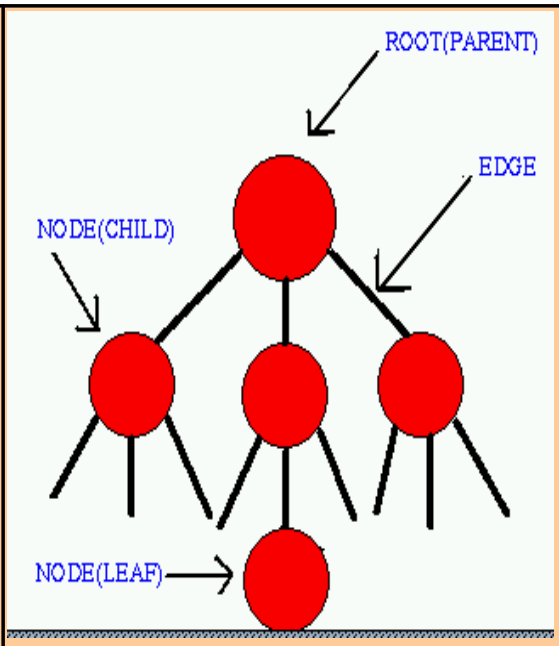
Trees may also be viewed as **graphs**, a mathematical structure consisting of **vertices** with connections between them. In this context, the connections between vertices are called **edges**.

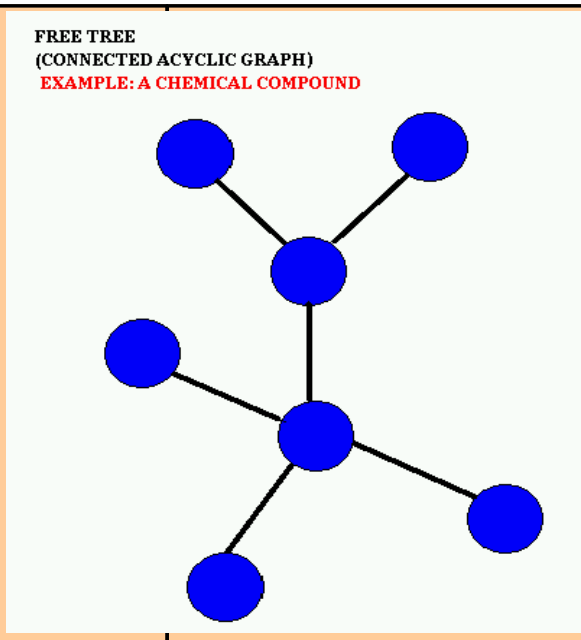
In C and C++, the connections between the nodes are specified by pointers, much like those used to create linked lists.

A set of nodes with connections between them is called a **free tree**, if there is no particular distinguishing node. A **rooted tree**, on the other hand, singles out a particular node called a root which is the entry point to that structure.

Tree Data structure

Terminology
Nodes
Leaf Nodes
Degree of a node
Level of a node
Height of a node
Depth of a node
Height of a tree





Ordered Rooted Trees

An ordered tree consists of a root node and any number of children which are ordered from "oldest" to "youngest", hence the name. Each of the children may be the root of sub-trees.

Position Trees

This type of tree superficially resembles an ordered tree. However, the children are not identified by their age, but by their position. That is, two ordered trees consisting of only two nodes are identical: they both consist of a root and an oldest child. Two two-node position trees, on the other hand, may be quite different. They both have roots, but one may have a child in position #1, and the other a child in position #2.

In general, a k-ary position tree is a position tree with a branch factor of k. The most important of these are **binary trees** (k=2).