# Introduction to Trees

Trees are used in many areas of computer science such as operating systems, graphics, database systems, and computer networking. Tree data structures have many things in common with their botanical namesakes. A tree data structure has roots, branches, and leaves just like an oak or maple tree would. The difference between them is that a tree data structure has its roots at the top and leaves on the bottom. In later lessons, we will look at specific examples but let's become familiar with some terminology first.

Let's watch the following video, "Introduction to Trees" (4:02), to learn more.
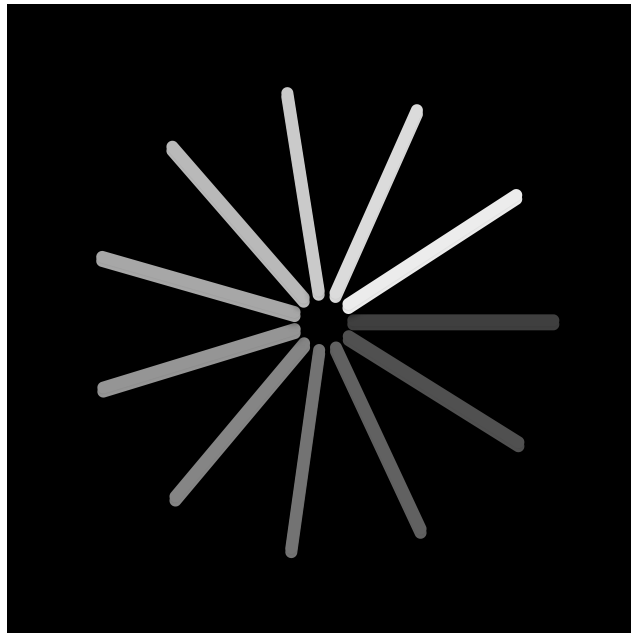
Vocabulary and Definitions

Node

A node is a fundamental part of a tree. It can have a name, which we call the "key." A node may also have additional information. We call this additional information the "payload." While the payload information is not central to many tree algorithms, it is often critical in applications that make use of trees.

Edge

An edge is another fundamental part of a tree. An edge connects two nodes to show that there is a relationship between them. Every node (except the root) is connected by exactly one incoming edge from another node. Each node may have several outgoing edges.

Root

The root of the tree is the only node in the tree that has no incoming edges.

Path

A path is an ordered list of nodes that are connected by edges. For example, Mammal $\longrightarrow$ Carnivora $\longrightarrow$ Felidae $\longrightarrow$ Felis $\longrightarrow$ Domestica is a path.

Children

The set of nodes c that have incoming edges from the same node to are said to be the children of that node.

Parent

A node is the parent of all the nodes it connects to with outgoing edges.

Sibling

Nodes in the tree that are children of the same parent are said to be siblings. The nodes etc/ and usr/ are siblings in the filesystem tree.

Subtree

A subtree is a set of nodes and edges comprised of a parent and all the descendants of that parent.

Leaf Node

A leaf node is a node that has no children.

Level

The level of a node n is the number of edges on the path from the root node to n.

Height

The height of a tree is equal to the maximum level of any node in the tree.

What is a Tree?

With the basic vocabulary now defined, we can move on to a formal definition of a tree. In fact, we will provide two definitions of a tree. One definition involves nodes and edges. The second definition, which will prove to be very useful, is a recursive definition.
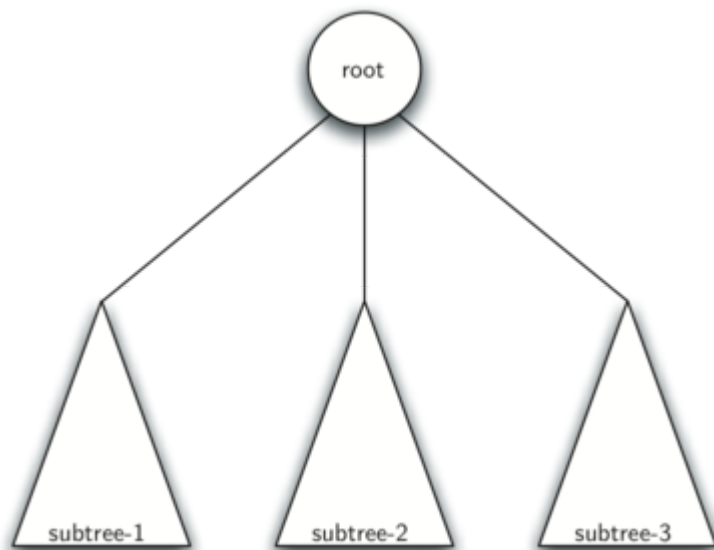
Definition One: A tree consists of a set of nodes and a set of edges that connect pairs of nodes. A tree has the following properties:

- One node of the tree is designated as the root node.
- Every node n, except the root node, is connected by an edge from exactly one other node p, where p is the parent of n.
- A unique path traverses from the root to each node.
- If each node in the tree has a maximum of two children, we say that the tree is a **binary tree**.

The figure below illustrates a tree that fits definition one. The arrowheads on the edges indicate the direction of the connection.
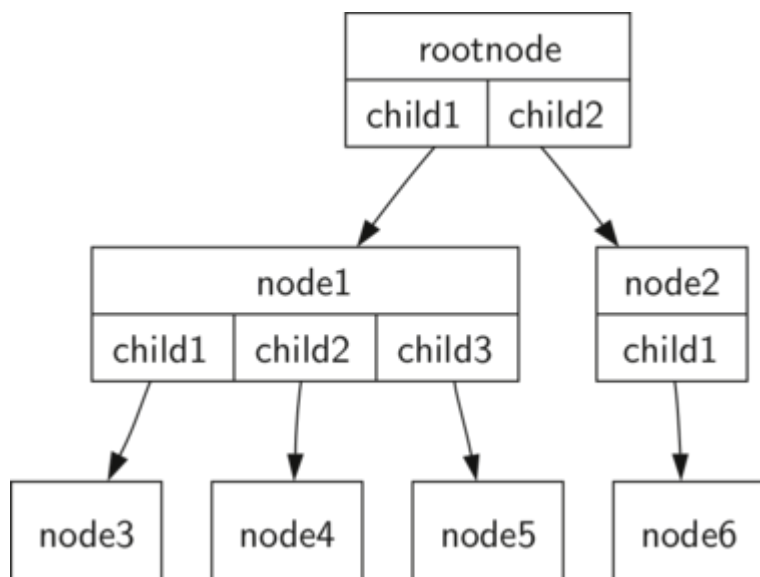
Definition Two: A tree is either empty or consists of a root and zero or more subtrees, each of which is also a tree. The root of each subtree is connected to the root of the parent tree by an edge. The figure below illustrates this recursive definition of a tree. Using the recursive definition of a tree, we know that the tree in has at least four nodes, since each of the triangles representing a subtree must have a root. It may have many more nodes than that, but we do not know unless we look deeper into the tree.