**Trees**

- Lists, Stacks, Queues are **linear data structures**
- Trees allow for **hierarchical** relationships
  - nodes have **parent-child** relation

**General Trees (definition)**

A **tree** is either **empty** or a root node connected to 0 or more trees (called **subtrees**)

There is a **unique path** from the root to each node in the tree
Trees (jargon)

- Each node is either a **leaf** or an **internal node**
  - An internal node has one or more children
  - A leaf node (external node) has no children
- Nodes with the same parent are **siblings**

Paths

- A path from node \(v_0\) to \(v_n\) is a sequence of nodes \(v_0, v_1, v_2, \ldots, v_n\), where there is an edge from one node to the next.
- The **descendants** of a node \(v\) are all nodes reached by a path from node \(v\) to the leaf nodes.
- The **ancestors** of a node \(v\) are all nodes found on the path from the root to node \(v\).

Depth and Height

- The length of a path is the number of edges in the path.
- The **depth** (level) of a node \(v\) is the length of the path from the root node to \(v\).
- The **height** of a node \(v\) is the length of the path from \(v\) to its deepest descendant.
Tree Properties

- **root**
  - The depth of the tree is the depth of the deepest node.
  - The height of the tree is the height of the root.

How to implement general trees?

**Node:**
- data
- parent
- children array

Traversals
Traversing a tree:

A traversal is a method that “visits” every node in a tree once.
Preorder Traversal

1 algorithm preorder(p) {
2     visit(p)
3     for each child c of p {
4         preorder(c)
5     }
6 }

UNIX File System

How to compute amount of space used by files in folders and subfolders?

Postorder Traversal

1 algorithm postorder(p) {
2     for each child c of p {
3         postorder(c)
4     }
5     visit(p)
6 }