A tree is a collection of elements together with a hierarchical relationship between the elements.

**DEF:**
- A single element (node) is a tree.
- If $n$ is a node and $T_1, T_2, \ldots, T_n$ are trees then $n$ related to $T_1, \ldots, T_n$ drawn as $n$ is called the root of the tree. $T_1$ to $T_n$ are called the subtrees of $n$.

**Example**

- The root of each subtree $T_1, T_2, \ldots, T_n$ is called a child of $n$.
- $n$ is called the parent of the roots of each $T_1, \ldots, T_n$.
- Nodes with the same parent are called siblings.
- Nodes with no children are called leaves.

- The degree of a node is the number of children a node has.
- The degree of a tree is the highest degree of a node in the tree.

- A path in a tree is a sequence of nodes $(n_0, n_1, n_2, \ldots, n_n)$ where $n_{i+1}$ is the parent of $n_i$. 

"Ternary tree"
• The depth of a node is the length of the path from that node to the root.
  - NOTE: there is exactly one path from every node to the root.

• The height of a tree is the length of the longest path from any node to the root.

• Given 2 nodes a and b,
  - a is called an ancestor of b if there is a path from b to a.
  - b is called a descendant of a if there is a path from b to a.
  - NOTE: The root of a tree is every node's ancestor.

Applications in Comp. Sci.
File system.

Expressions: “3 + 9* (5 - 8)”

Tree Traversals.
• Sequentially do something with every element of a tree.

```
pre-order (T)
{ print root of T
  for every subtree T of T, post-order(T')
}
```

```
post-order (T)
{ for every subtree T of T, post-order(T')
}
```
pre-order = h, a, i, k, f, b, c, d, e, g
post-order = i, k, a, b, f, d, e, g, c, h