* Use to represent hierarchical data. For example, file structure, company structure, family tree, etc. Another is to organize data for quick search, insertion, deletion.
* Tree is a collection of nodes, linked together to represent hierarchy that happen naturally. The top node is called the root.
* Terminiology to remember:
  + Root
  + Children
  + Parent
  + Sibling – Have same parent
  + Leaf – Has no children
  + If you can go from A to B in a traversal (which is one way only)
    - A is ancestor of B
    - B is descendant of A
  + Cousin – Nodes have same grandparent
* With a tree with N nodes, you will have N – 1 edges. Edges are the arrows that point to the next node. These edges are incoming edges, and you have the – 1 in there, because the root doesn’t have an incoming edge.
* Depth – Length of path from root to x, or number of edges in path from root to x.
* Height – Number of edges in longest path from x to a leaf. The height of the tree itself is height of the root node.
* Most common and best way to implement a tree is a Linked List

**Binary Tree**

* A tree such that each node can have at most 2 children.
* A strict/proper binary tree is one with either 2 or 0 children.
* A complete binary tre