Binary Tree Terms

Binary Tree: A common tree data structure in which each node contains data, has at most two children, and the first node is called the root node. The left child's data is less than or equal to its parent's data and the right child's data is greater than its parent's data.

Node: Implemented as a **struct** or **class** and contains at least one variable for data and two pointers to other dynamically created nodes.

Root: The first node in a binary tree. **Ex) Root is 125**.

Parent: Node that is above the current node. **Ex) 155 is parent of 100**.

Children: Nodes that are directly below the current node. **Ex) 150 and 300 are children of 200**.

Subtree: Nodes beginning with the left child (left subtree) and the right child (right subtree).

Leaf: A node with no children. **Ex) 120**

Sibling: Two nodes with the same parent. **Ex) 115 and 120**.

Ancestor: Nodes above the current node.

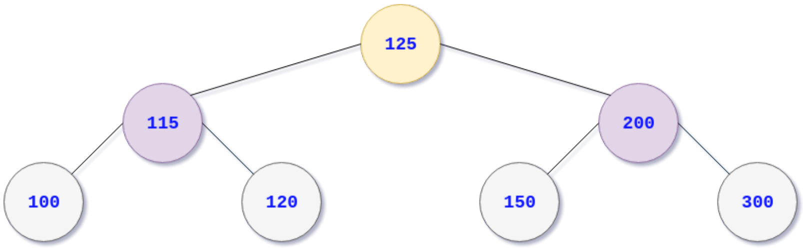
Descendant: Nodes below the current node.

Node Depth: Starting at the current node, proceed upward until you reach the root node. Count each node you pass along the way. Root node has a depth of zero. **Ex) 115 has depth 1**.

Tree Depth: Depth of the furthest leaf from the root. If root is the only node, depth of the tree is zero. **Ex) Tree = depth 2.**

Full Binary Tree: Every leaf has the same depth. Every node that is not a leaf has two children. **Ex) Tree is full**.

Complete Binary Tree: Similar to a Full Binary Tree except the depth of the leaf nodes need not be filled. In the depth with the leaf nodes, each node is as far left as possible. Therefore, not every node above the leaf nodes need to have two children. **Ex) If 300 was deleted, tree would be complete but not full**.


125
115
200
100
120
150
300