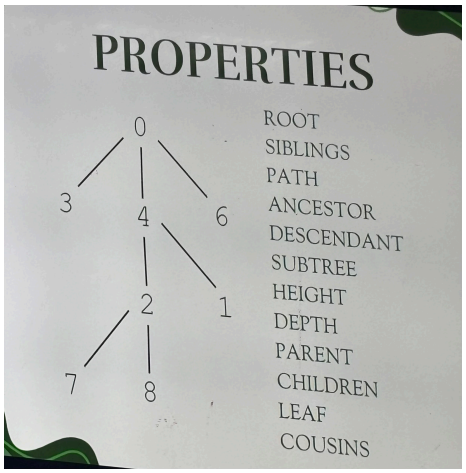

Property	Definition	Example
Root	The very top node. The one with no parent. Think “the original ancestor.”	0
Parent	A node that has one or more children connected below it.	Parent of 7, 8 is 2
Children	Nodes directly under a parent. Each child has exactly one parent.	7, 8 are children of 2
Siblings	Nodes that share the same parent. Basically, children of the same node.	7, 8 are siblings
Leaf	A node with no children. It’s at the end of a branch and can’t go any further down.	7, 8, 1, 6, 3
Path	The route you take from one node to another by following edges. Like breadcrumbs through the tree.	0 → 4 → 1
Ancestor	Any node you pass through when going up the tree. A node’s parent, grandparent, great-grandparent, etc.	For 2, ancestors are 4 and 0
Descendant	The opposite. Any node you reach by going down the tree from a given node.	For 4, descendants are 1, 2, 7, 8
Subtree	A smaller tree that starts at some node and includes all its descendants. Every node can be the root of its own subtree.	Subtree rooted at 2 = {7, 8}

Height	How tall the tree (or a node) is, the count of edges on the longest path down to a leaf. ● Height of a leaf = 0. ● Height of the whole tree = height of the root.	Height of 4 = 1, height of the entire tree = 3
Depth	How far a node is from the root, the count of edges from the root down to that node. ● Depth of the root = 0.	Depth of 4 = 2
Cousins	Nodes on the same level (same depth) but with different parents.	No cousins, put kid at 3, only then its children will be cousins with 2 and 1

Depth V.S Height:
Depth: From root → down to a node
Height: From a node → down to the deepest leaf

Favorite property: **root**

Because the root is the only part of the whole structure that isn't confused about where it belongs, and also everything else exists because of it. Without the root, the tree wouldn't even be a tree.