Trees: Definition and Terminology
Goals

Know the definition of a tree.

Know the following basic tree terminology:

root, child, parent, leaf, height, depth, subtree, descendant, ancestor
Trees in nature
Trees in computer science
public class ListNode {
    int value;
    ListNode next;
}

Linked List Node
public class TreeNode {
    int value;
    TreeNode next1;
    TreeNode next2;
}

A tree is like a linked list, except each node can have more than one successor (next).
public class ListNode {
    int value;
    ListNode next;
}

LinkedList

head

ListNode

value

next

ListNode

value

next

...
public class List {
    int value;
    List next;
}

The node is the list. next points to the tail of the list (also a list!)
public class Tree {
    int value;
    Tree left;
    Tree right;
}

The node is the tree.
left points to the left child of the tree (also a tree!)
right points to the right child of the tree (also a tree!)
Tree - Definition

**Tree**: like a linked list, but:
- Each node may have zero or more successors *(children)*
- Each node has exactly one predecessor *(parent)* except the root, which has none
- All nodes are reachable from root

**Binary tree**: A tree, but:
- Each node can have at most two children *(left child, right child)*
**Tree Terminology**

**M** is the **root** of this tree

**N** is the **left child** of **P**

**S** is the **right child** of **P**

**P** is the **parent** of **N**
M is the root of this tree
N is the left child of P
S is the right child of P
P is the parent of N
M and G are ancestors of D
P, N, S are descendants of W

A node's ancestors are the nodes on the path to the root
A node's descendants include its children, its children's children, and so on.
**M** is the root of this tree

**N** is the left child of **P**

**S** is the right child of **P**

**P** is the parent of **N**

**M** and **G** are ancestors of **D**

**P**, **N**, and **S** are descendants of **W**

The subtree rooted at **G** contains **G**, **D**, **J**, **B**, and **H**.

**G** is the root of the left subtree of **M**

---

A **subtree** is a subset of a tree containing a node and its descendants.
A leaf is a node with no children.

**Leaf**

M is the root of this tree
N is the left child of P
S is the right child of P
P is the parent of N
M and G are ancestors of D
P, N, S are descendants of W
The subtree rooted at G contains G, D, J, B, and H.
G is the root of the left subtree of M
B, H, J, N, S are leaves
M is the root of this tree
N is the left child of P
S is the right child of P
P is the parent of N
M and G are ancestors of D
P, N, S are descendants of W
The subtree rooted at G contains G, D, J, B, and H.
G is the root of the left subtree of M
B, H, J, N, S are leaves
The subtree rooted at W has height 2

The height of a tree is the length of the path from the root to the deepest leaf.
M is the root of this tree
N is the left child of P
S is the right child of P
P is the parent of N
M and G are ancestors of D
P, N, S are descendants of W
The subtree rooted at G contains G, D, J, B, and H.
G is the root of the left subtree of M
B, H, J, N, S are leaves
The subtree rooted at W has height 2
J is at depth 2

The depth of a node is the length of the path from the root to that node.