

[Courses](#)[Practice](#)[Roadmap](#)[Pro](#)

Algorithms and Data Structures for Beginners

16 / 35

16 - Binary Tree

☐

Mark Lesson Complete

[View Code](#)

[Prev](#)

[Next](#)

20 Breadth-First Search

Binary Trees

Concept

The Binary Tree is another data structure that involves nodes and pointers. We talked about nodes in the linked list chapter. We connected these nodes in a straight line with `next` and `prev` pointers. Nodes in a binary tree also have at most two pointers, but we call them the **left child** and the **right child** pointers. The first node in a binary tree is referred to as the **root** node. We draw the pointers down instead of a straight line.

The value of a node can be any data type. A `TreeNode` class would look like the following. Notice how much of the implementation is similar to a `ListNode` discussed in the linked list chapter, except now we have `left_child` and `right_child`

```
class TreeNode:
    constructor(val):
        val = val
        left_child = any value (primitive, Object etc)
        right_child = any value (primitive, Object etc)
```

If a node does not have any children, it is classified as a **leaf** node. If a node has even a single child, either left or right, it would be classified as a **non-leaf** node.

Unlike linked lists, binary tree node pointers can only point in one direction. As such, cycles are not allowed in binary trees. Mathematically speaking, a binary tree is an undirected graph with no cycles. This means that a leaf node is always guaranteed to exist.

The same applies to the decision trees that are used in recursion.

The following section demonstrates how binary trees are drawn and their terminology that is crucial to understand binary tree problems in interviews.

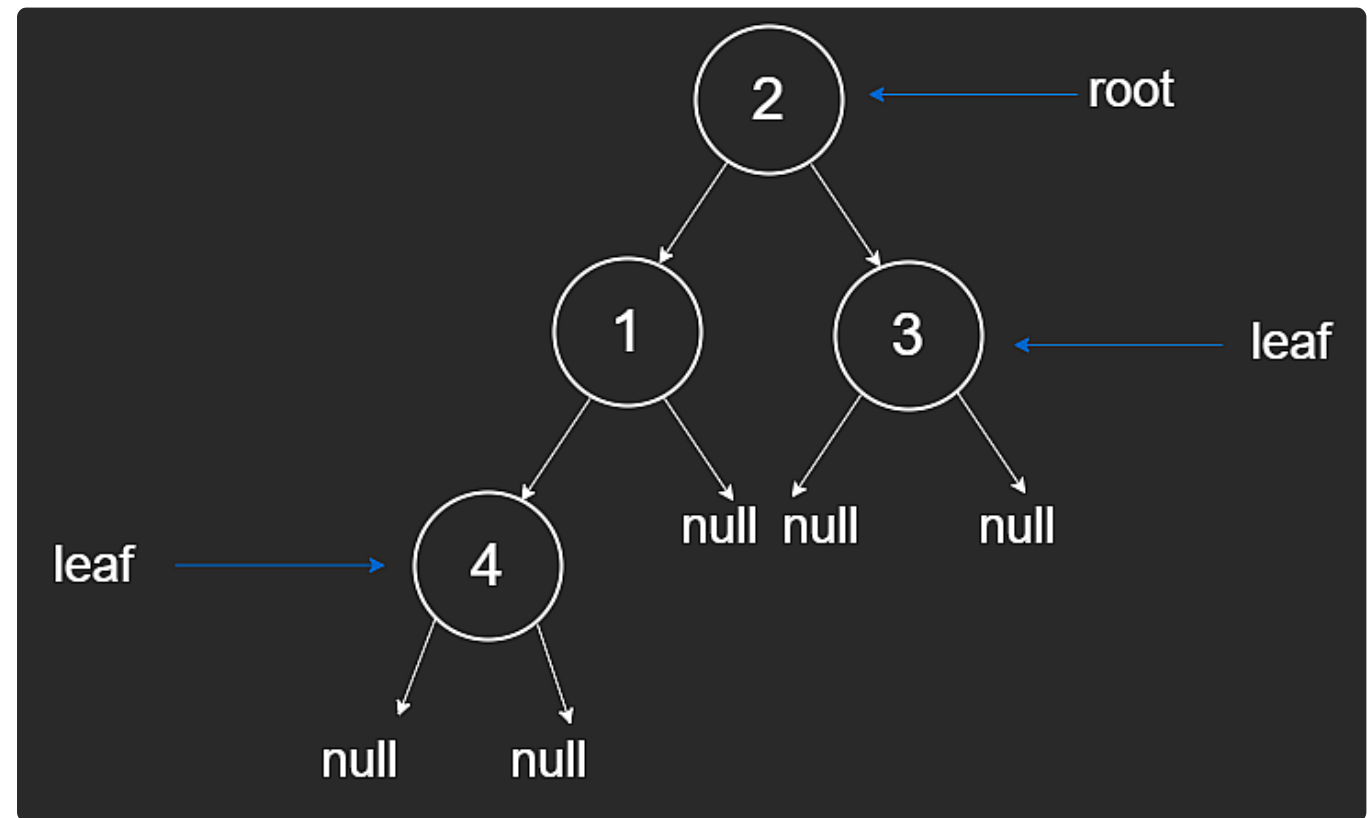
Properties

Root Node

Root node is the highest node in the tree and has no parent node. All of the nodes in the tree can be reached by the root node.

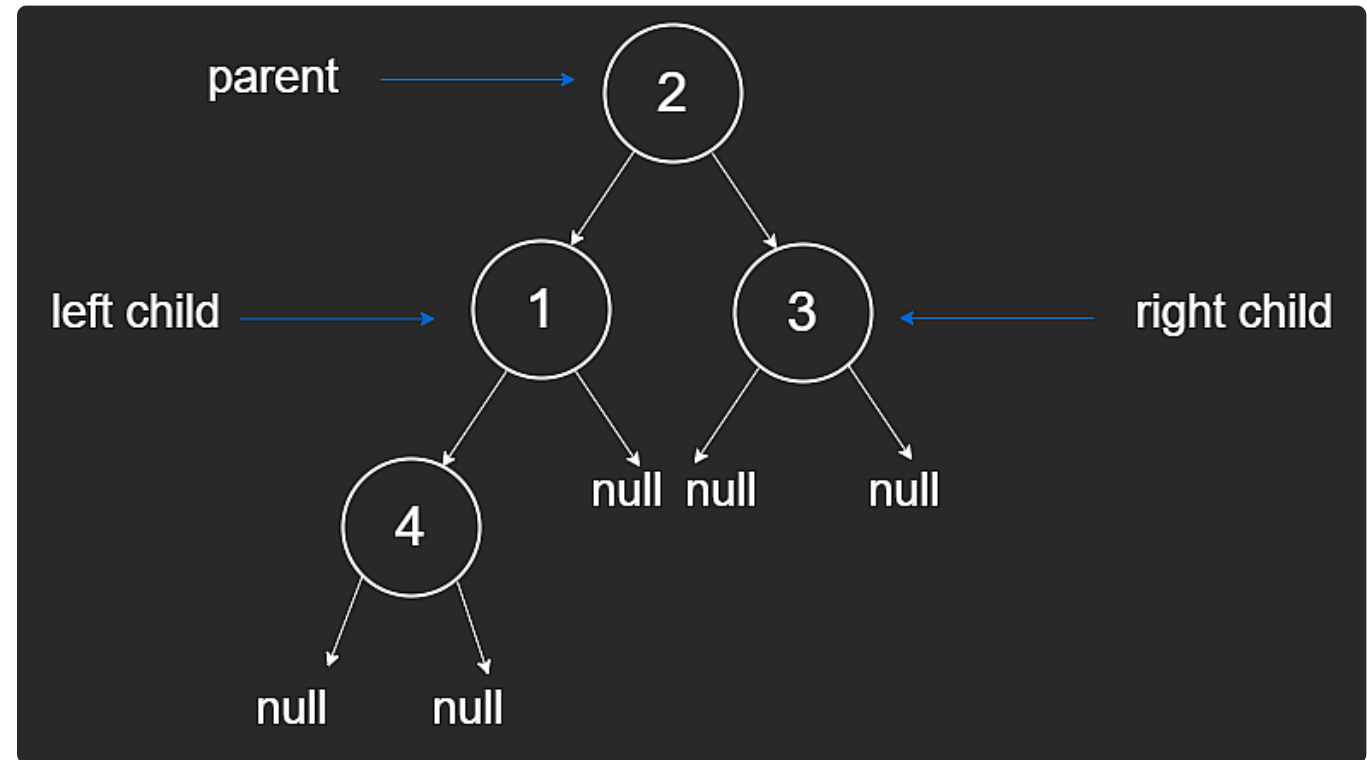
Leaf Nodes

Leaf nodes are nodes with no children. The nodes at the last level of the tree are guaranteed to be leaf nodes but they can also be found on other levels.



Children

The children of a node are its left child and right child.



Height

The height of a binary tree is measured from the root node all the way to the lowest leaf node, just like the height of anything in real life. The height of a single node tree is just 1, if the node itself is counted, or 0 if not.

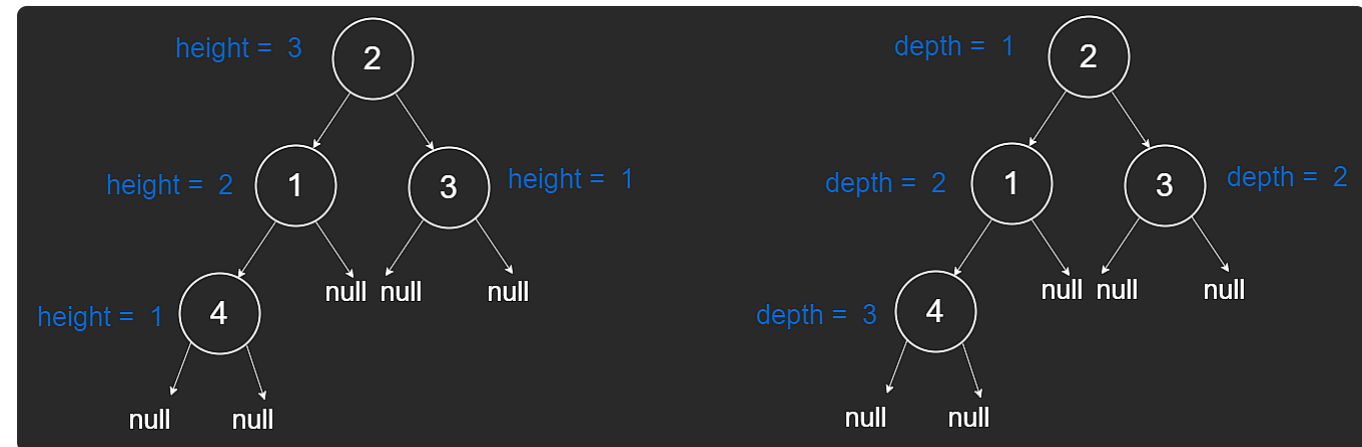
Sometimes, the height is counted by the number of edges that are in between the nodes instead of the nodes themselves. Using this method, the height will be $n-1$ where n is the number of nodes, in the path from the root to the lowest leaf.

The maximum height of the given binary tree in the visual below is 3. Alternatively, if we were counting by edges, instead of nodes, it would be 2.

The number of edges in a tree are $n - 1$, where n is the number of nodes.

Depth

Depth of a binary tree node is measured from itself all the way up to the root. As observed in the visual below, the depth at the root node is 1, with it increasing as we go down. Measure depth at a given node by looking at how many nodes are above it, including the node itself.

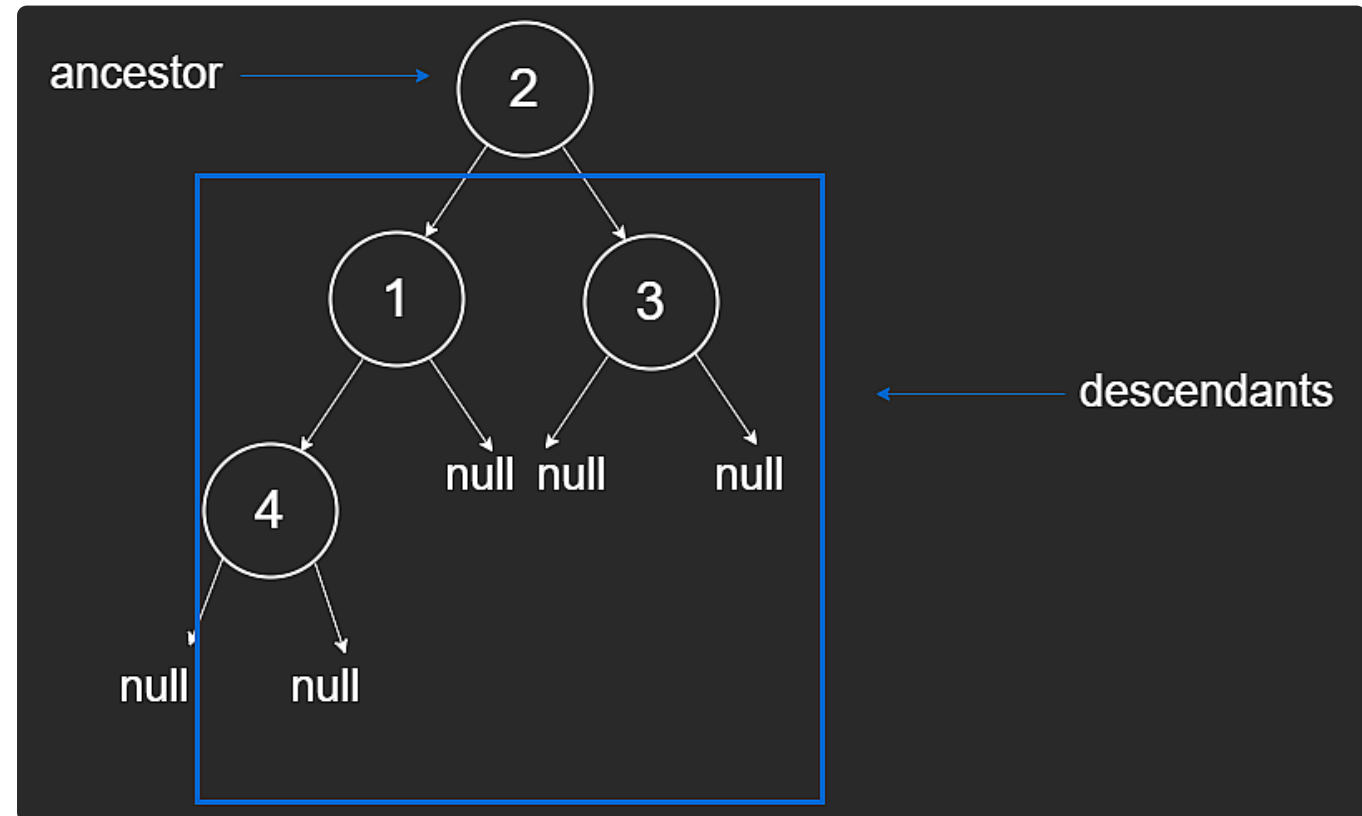


Ancestor

A node connected to all of the nodes below it is considered an ancestor to those nodes.

Descendent

The descendent of a node is either child of the node or child of some other descendent of the node.



Closing Notes

Though most of this chapter was definitions, these terms are fundamental to understand when it comes to solving binary tree questions and will help us in the

later chapters.

Copyright © 2023 NeetCode.io All rights reserved.

Contact: neetcodebusiness@gmail.com

[Github](#) [Privacy](#) [Terms](#)