Introduction to Binary Trees: Takeaways ⇱

Syntax

- Implementing a binary tree:

```python
class Node:

    def __init__(self, value=None):

        self.value = value

        self.left = None

        self.right = None

class BinaryTree:

    def __init__(self, root=None):

        self.root = root

    def insert(self, value):

        if not self.root:

            self.root = Node(value=value)

        elif not self.root.left:

            self.root.left = Node(value = value)

        elif not self.root.right:

            self.root.right = Node(value = value)
```

Concepts

- A tree data structure is naturally described using recursion.
- A node is an abstract data type that contains references to left and right nodes.
  - Nodes can be `None` or can store other data, like an integer.
  - The topmost node is known as the root node.
  - A node that has left and right references of type `None` is known as a leaf.

- A binary tree is a tree data structure in which each node has at most two children, which are referred to as the left child and the right child.
- A child node is a node that is added and that isn't the root.
- A parent node is the node the child node references. The root node is the only parent node with no parent.
- Level order insert works the following way:
    - Given a list of an integers, start at the first element and make it the root node.
    - Continue down the list adding the following two nodes at the second level, then the next four as the third level, etc. The pattern is that at level   , you will have   nodes.
- The value of every left side node corresponds to   .
- The value of every right side node corresponds to   .
- A node is an interior node if the node has a parent *and* it has both a left node or a right mode.
- Traversal methods:
    - Preorder Traversal:
        - Handles the value of the current node.
        - Recursively traverse all the nodes on the left side.
        - Recursively traverse all the nodes on the right side.

    - Inorder Traversal:
        - Recursively traverse all the nodes on the left side.
        - Handles the value of the current node.
        - Recursively traverse all the nodes on the right side.

    - Postorder Traversal:
        - Recursively traverse all the nodes on the left side.
        - Recursively traverse all the nodes on the right side.
        - Handles the value of the current node.
- The depth of a tree is defined as the maximum level of a tree. We can find the depth of the tree by recursively traversing it.
- A balanced tree is a tree in which for every node, a subtree's height does not differ by more than one.

- A complete binary tree is a tree that has all levels completely filled except the last level.
- Binary heaps allow us to query the top values of extremely large data sets.

Resources

- [Binary trees](#)
- [Tree data structures for beginners](#)