Trees

Overview

Trees are a data structure that utilizes nodes and parent-child relationships between nodes to store data in a tree-like structure. Here is an example class for a binary tree (tree with two children nodes).

```
public class Tree {
    Node root;
    private class Node {
        int val;
        Node left;
        Node right;

        public Node(int v, Node l, Node r) {
            val = v;
            left = l;
            right = r;
        }
    }

    public Tree(Node n) {
        root = n;
    }

    ... // rest of class
}
```

It may seem strange that we wrap the Node class within the Tree class, but it makes operating on the tree a lot easier when the data structure is implemented this way. The Tree class only keeps record of the root of the tree, while each individual Node will keep track of the structure of the tree.

Trees alone aren't very helpful, but they are the building blocks of several useful data structures including, Binary Search Trees, Balanced Trees, Game Trees. They are also useful in understanding search algorithms in graphs.

TL;DR

Trees by themselves aren't very useful, but are important building blocks of other data structures.