Lowest Common Ancestor

BINARY SEARCH TREE

Understanding Binary Search Tree

- BST is an extension of Binary Tree with two extra conditions.
  - Left child is always smaller than the root.
  - Right child is always greater than the root.

Why the word LOWEST is important

- The root is always an ancestor of all the child nodes no matter where the nodes are positioned.
- So, if we are sure that the child nodes are present in the tree, we can just return the root node.

The Idea

In BST, the common ancestor for two nodes is the node with value that is between the given values.

A common doubt

- What if the given nodes form parent and child in the binary tree?
  - According to the definition of binary tree, the parent is always its own child.

The Algorithm

1. We start from the root.
2. We check if the input values lean toward the same direction.
3. If they don’t, return the root.
4. If they do lean in any direction, we call the function with the root node changed to the new node.
5. Go to step 2.

The Code

```
class theCod3 {
    public Node LCA(Node root, Node p, Node q) {
        if (p.val > root.val && q.val > root.val) {
                    return LCA(root.right, p, q);
        } else if (p.val < root.val && q.val < root.val) {
            return LCA(root.left, p, q);
        }
        return root;
    }
}
```