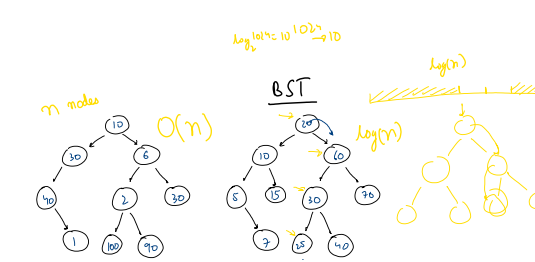
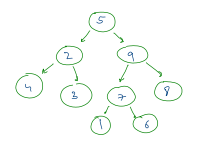
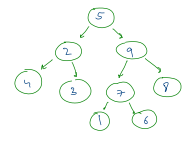
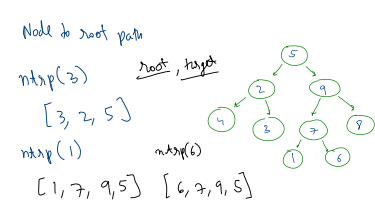
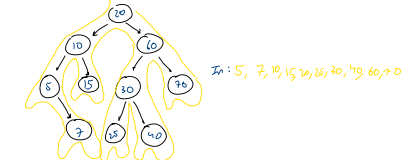
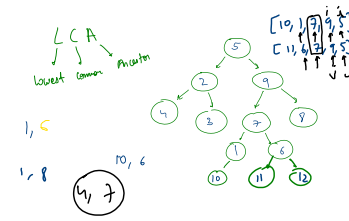
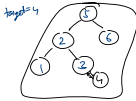


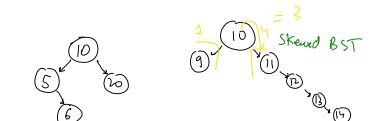
```

function lowestCommonAncestor(root, p, q) {
    if (!root || !p || !q) return null;
    if (root === p || root === q) return root;
    if (root.val < p.val && root.val < q.val) return lowestCommonAncestor(root.left, p, q);
    if (root.val > p.val && root.val > q.val) return lowestCommonAncestor(root.right, p, q);
    return root;
}

```



For all nodes, left subtree should have smaller value, right subtree should have larger value



Inorder is always sorted in BST

