```java
// Function to find the lowest common ancestor in a BST.
// Node{int data, Node left, Node right}
Node LCA(Node root, int n1, int n2) {
    Queue<Node> queue = new LinkedList<>();
    queue.add(root);
    while (!queue.isEmpty()) {
        Node u = queue.remove();
        if (u.data == n1 || u.data == n2) {
            return u;
        }
        if (u.data > Math.min(n1, n2) && u.data < Math.max(n1, n2)) {
            return u;
        }
        if (u.left != null) {
            queue.add(u.left);
        }
        if (u.right != null) {
            queue.add(u.right);
        }
    }
    return null;
}
```