Insert:

Delete:

1. Recursively find the node that has the same value as the key, while setting the left/right nodes equal to the returned subtree
2. Once the node is found, have to handle the below 4 cases

* node doesn't have left or right - return null
* node only has left subtree- return the left subtree
* node only has right subtree- return the right subtree
* node has both left and right - find the minimum value in the right subtree, set that value to the currently found node, then recursively delete the minimum value in the right subtree

public class Solution {

public TreeNode deleteNode(TreeNode root, int key) {

if (root == null) return null;

if (key < root.val) root.left = deleteNode(root.left, key);

else if (key > root.val) root.right = deleteNode(root.right, key);

else if (root.left == null) return root.right;

else if (root.right == null) return root.left;

else {

root.val = findMin(root.right);

root.right = deleteNode(root.right, root.val);

}

return root;

}

private int findMin(TreeNode node) {

while (node.left != null) node = node.left;

return node.val;

}

}

Search:

Verification: