```java
/**
 * public class TreeNode {
 *   public int key;
 *   public TreeNode left;
 *   public TreeNode right;
 *   public TreeNode(int key) {
 *     this.key = key;
 *   }
 * }
 */
public class Solution {
  public boolean isBalanced(TreeNode root) {
    if (root == null) {
      return true;
    }
    if (Math.abs(getHeight(root.left) - getHeight(root.right)) > 1) {
      return false;
    }
    return isBalanced(root.right) && isBalanced(root.left);
  }
  private int getHeight(TreeNode root) {
    if (root == null) {
      return 0;
    }
    int L = getHeight(root.left);
    int R = getHeight(root.right);
    return Math.max(L, R) + 1;
  }
}

```

```java
/**
 * public class TreeNode {
 *   public int key;
 *   public TreeNode left;
 *   public TreeNode right;
 *   public TreeNode(int key) {
 *     this.key = key;
 *   }
 * }
 */
public class Solution {
  public int findHeight(TreeNode root) {
    if(root == null){
      return 0;

    }
    int leftHeight = findHeight(root.left);
    int rightHeight = findHeight(root.right);
    return Math.max(findHeight(root.left),findHeight(root.right))+1;

  }
}

```

```java
/**
 * public class TreeNode {
 *   public int key;
 *   public TreeNode left;
 *   public TreeNode right;
 *   public TreeNode(int key) {
 *     this.key = key;
 *   }
 * }
 */
public class Solution {
  public int countNodes(TreeNode root) {
    if (root == null) {
      return 0;
    }
    int left = countNodes(root.left);
    int right = countNodes(root.right);
    return countNodes(root.left) + countNodes(root.right) + 1;
  }
}

```

```java
/**
 * public class TreeNode {
 *   public int key;
 *   public TreeNode left;
 *   public TreeNode right;
 *   public TreeNode(int key) {
 *     this.key = key;
 *   }
 * }
 */
public class Solution {
  public List<Integer> inOrder(TreeNode root) {
    List<Integer> res = new ArrayList<>();
    helper(root,res);
    return res;
  }
  private void helper(TreeNode root ,List<Integer> res){
    if (root == null){
      return;
    }
    helper(root.left,res);
    res.add(root.key);
    helper(root.right,res);
  }
}
```

```java
/**
 * public class TreeNode {
 *   public int key;
 *   public TreeNode left;
 *   public TreeNode right;
 *   public TreeNode(int key) {
 *     this.key = key;
 *   }
 * }
 */
public class Solution {
  public List<Integer> preOrder(TreeNode root) {
    List<Integer> res = new ArrayList<>();
    helper(root,res);
    return res;
  }
  private void helper(TreeNode root, List<Integer> res){
    if (root == null ){
      return;
    }
    res.add(root.key);
    helper(root.left,res);
    helper(root.right,res);
  }
}

```

```java
/**
 * public class TreeNode {
 *   public int key;
 *   public TreeNode left;
 *   public TreeNode right;
 *   public TreeNode(int key) {
 *     this.key = key;
 *   }
 * }
 */
public class Solution {
  public List<Integer> postOrder(TreeNode root) {
    List<Integer> res = new ArrayList<>();
    helper(root, res);
    return res;
  }
  private void helper(TreeNode root, List<Integer> res) {
    if (root == null) {
      return;
    }
    helper(root.left, res);
    helper(root.right, res);
    res.add(root.key);
  }
}

```