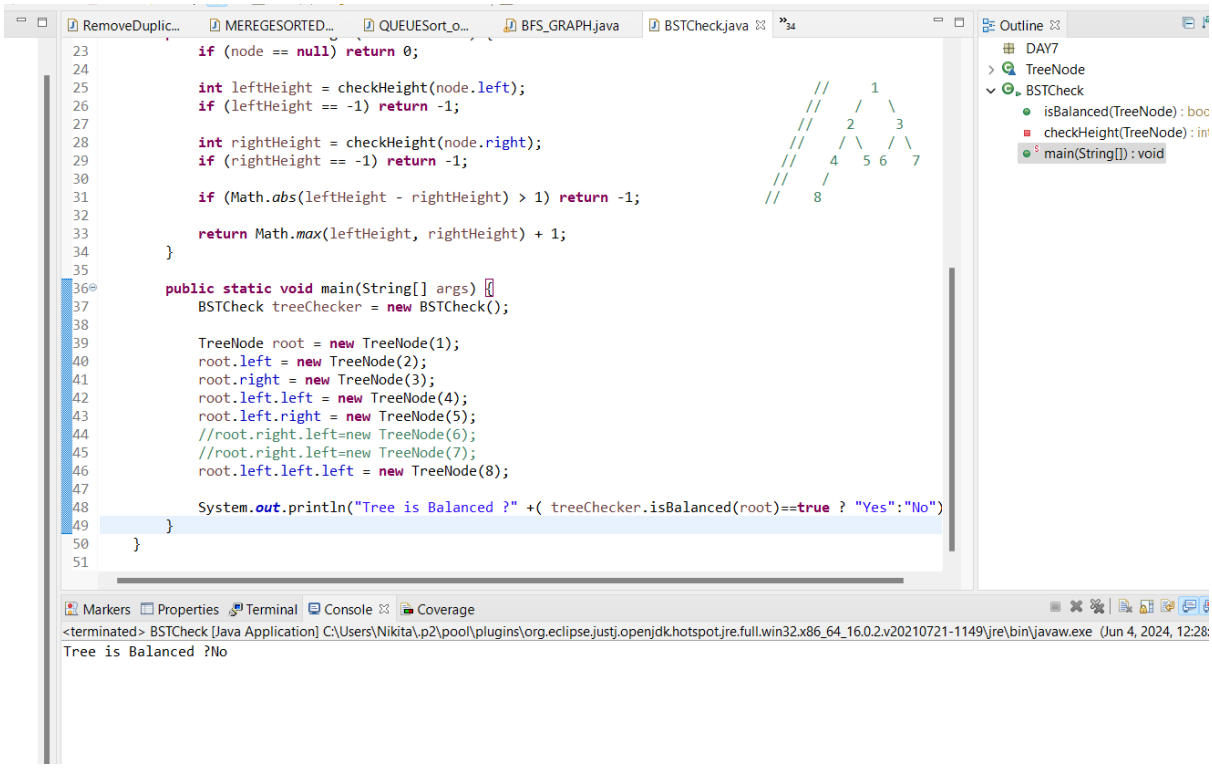
Binary Search Tree Check

```java
package DAY7;



	class TreeNode {
	    int val;
	    TreeNode left;
	    TreeNode right;
	    TreeNode(int val) {
	        this.val = val;
	        this.left = null;
	        this.right = null;
	    }
	}

	public class BSTCheck {

	    public boolean isBalanced(TreeNode root) {
	        return checkHeight(root) != -1;
	    }

	    private int checkHeight(TreeNode node) {
	        if (node == null) return 0;

	        int leftHeight = checkHeight(node.left);
//	     1
	        if (leftHeight == -1) return -1;
//	    /   \
	
//	    2     3
	        int rightHeight = checkHeight(node.right);
//	    / \   / \
	        if (rightHeight == -1) return -1;
//	    4   5 6   7

//	    /
	        if (Math.abs(leftHeight - rightHeight) > 1) return -1;
//	    8

	        return Math.max(leftHeight, rightHeight) + 1;
	    }

	    public static void main(String[] args) {
	        BSTCheck treeChecker = new BSTCheck();

	        TreeNode root = new TreeNode(1);
	        root.left = new TreeNode(2);
	        root.right = new TreeNode(3);
	        root.left.left = new TreeNode(4);
	        root.left.right = new TreeNode(5);
	        //root.right.left=new TreeNode(6);
	        //root.right.left=new TreeNode(7);
	        root.left.left.left = new TreeNode(8);

	        System.out.println("Tree is Balanced ?" +(
treeChecker.isBalanced(root)==true ? "Yes":"No"));
	    }
```

```
    }
```


RemoveDuplic...
MEREGESORTED...
QUEUESort_o...
BFS_GRAPH.java
BSTCheck.java
Outline
DAY7
TreeNode
BSTCheck
isBalanced(TreeNode) : boo
checkHeight(TreeNode) : in
main(String[]) : void
if (node == null) return 0;
int leftHeight = checkHeight(node.left);
if (leftHeight == -1) return -1;
int rightHeight = checkHeight(node.right);
if (rightHeight == -1) return -1;
if (Math.abs(leftHeight - rightHeight) > 1) return -1;
return Math.max(leftHeight, rightHeight) + 1;
public static void main(String[] args) {
BSTCheck treeChecker = new BSTCheck();
TreeNode root = new TreeNode(1);
root.left = new TreeNode(2);
root.right = new TreeNode(3);
root.left.left = new TreeNode(4);
root.left.right = new TreeNode(5);
//root.right.left=new TreeNode(6);
//root.right.left=new TreeNode(7);
root.left.left.left = new TreeNode(8);
System.out.println("Tree is Balanced ?" +(treeChecker.isBalanced(root)==true ? "Yes":"No")
1
2 3
4 5 6 7
8
Markers
Properties
Terminal
Console
Coverage
<terminated> BSTCheck [Java Application] C:\Users\Nikita\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\jre\bin\javaw.exe (Jun 4, 2024, 12:28
Tree is Balanced ?No