

⑥ Leaf similar: check if leaf nodes of two trees are similar.

```
boolean leafSimilar(Node root1, Node root2) {  
    Stack<Node> s1 = new Stack<Node>();  
    Stack<Node> s2 = new Stack<Node>();  
  
    inOrder(root1, s1);  
    inOrder(root2, s2);  
    if (s1.size() != s2.size()) return false;  
    while (!s1.empty() || !s2.empty()) {  
        Node n1 = s1.pop();  
        Node n2 = s2.pop();  
        if (n1 == null & n2 != null)  
        if (n1.val != n2.val) return false;  
    }  
}
```

~~return s₁.empty() & s₂.empty();~~

3
void inOrder(Node n, ~~Stack~~ Stack<Node> s) {
 if (n == null) return;
 if (n.left == null & n.right == null)
 s.push();
 inOrder(n.left, s);
 inOrder(n.right, s);
}

Algo:
1. Create 2 stacks and add leaf nodes.
2. Compare the stack contents until stack is empty.
3. False, if stack sizes are different. or nodes do not match.