**Question:** You are given the roots of two binary trees. Write a recursive function **is\_identical(tree1\_root, tree2\_root)** that finds whether these trees are identical or not. Return true if the trees have the same structure and the same values at corresponding nodes, otherwise false.

**Note:** The function should not use any loops. You may use any number of helper functions as needed. Assume the TreeNode class is provided, where each node contains an integer value in the element attribute and pointers to the left and right child nodes.

| **Sample Input (only roots of trees will be given)** | **Sample Output** |
| --- | --- |
| 1 1  / \ / \  2 3 2 3  / \ / \  4 5 4 5 | true |
| 1 1  / \ / \  2 3 3 2  / \ \ /  4 5 5 4 | false |

**Question:** You are given the root of a binary tree. Write a recursive function is\_symmetric(root) that checks whether the tree is symmetric around its centre. Return True if the tree is symmetric, otherwise return False. A binary tree is symmetric if the left subtree is a mirror reflection of the right subtree.

**Note:** The function should not use any loops. You may use any number of helper functions as needed. Assume the TreeNode class is provided, where each node contains an integer value in the element attribute and pointers to the left and right child nodes.

| **Sample Input (only the root of the tree will be given)** | **Sample Output** |
| --- | --- |
| 1  / \  2 2  / \ / \  3 4 4 3 | True |
| 1  / \  2 2  \ \  3 3 | False |