Symmetric Tree Documentation

Problem Description

Given the root of a binary tree, check whether it is a mirror of itself (i.e., symmetric around its center).

Example 1

Input: root = [1, 2, 2, 3, 4, 4, 3]

Output: true

Example 2

Input: root = [1, 2, 2, null, 3, null, 3]

Output: false

Constraints

The number of nodes in the tree is in the range [1, 1000].

Node values are in the range [-100, 100].

Follow-up

- Could you solve it both recursively and iteratively?

Explanation

Recursive Approach

The isMirror helper function is used to compare two subtrees. The function checks:

1. If both subtrees are empty, they are symmetric.
2. If only one subtree is empty, they are not symmetric.
3. If the values of the current nodes are equal, and the left subtree of the first tree is a mirror of the right subtree of the second tree, and vice versa, then the trees are symmetric.

Iterative Approach

The iterative approach uses a queue to perform a level-order traversal:

1. Start by enqueuing the left and right children of the root.
2. *For each pair of nodes dequeued:*

- If both nodes are null, continue to the next pair.
- If only one is null or their values do not match, return False.
- Enqueue the children in the order that ensures mirrored comparison: left's left with right's right, and left's right with right's left.

This approach continues until the queue is empty, ensuring that the tree is symmetric if no discrepancies are found.