101 Symmetric Tree

Problem

Given a binary tree, check whether it is a mirror of itself.

Idea

The idea is to write a recursive function isMirror() that takes two trees as an argument and returns true if trees are the mirror and false if trees are not mirrored. The isMirror() function recursively checks two roots and subtrees under the root.

Python Implementation

```
def isMirror(root1, root2):
    # If both trees are empty, then they are mirror images
    if root1 is None and root2 is None:
        return True

    """ For two trees to be mirror images,
        the following three conditions must be true
        1 - Their root node's key must be same
        2 - left subtree of left tree and right subtree
          of the right tree have to be mirror images
        3 - right subtree of left tree and left subtree
           of right tree have to be mirror images
    """
    if (root1 is not None and root2 is not None):
        if root1.key == root2.key:
            return (isMirror(root1.left, root2.right)and
                    isMirror(root1.right, root2.left))

    # If none of the above conditions is true then root1
    # and root2 are not mirror images
    return False


def isSymmetric(root):

    # Check if tree is mirror of itself
    return isMirror(root, root)
```

Time Complexity: O(N)

Auxiliary Space: O(h) where h is the maximum height of the tree

Iterative approach suing a queue:

Idea

The basic idea is to check if the left and right subtrees of the root node are mirror images of each other. To do this, we perform a level-order traversal of the binary tree using a queue. We push the root node into the queue twice, initially. We dequeue two nodes at a time from the front of the queue and check if they are mirror images of each other.

Steps

Create a stack and push the root node onto it twice.

While the stack is not empty, repeat the following steps:

a. Dequeue two nodes from thequeue, say node1 and node2.

b. If both node1 and node2 are null, continue to the next iteration.

c. If one of the nodes is null and the other is not, return false as it is not a mirror.

d. If both nodes are not null, compare their values. If they are not equal, return false.

e. Push the left child of node1 and the right child of node2 onto the queue.

f. Push the right child of node1 and the left child of node2 onto the queue.