Balanced Binary Tree

Question: Given a binary tree, determine if it is height-balanced.

For this problem, a height-balanced binary tree is defined as a binary tree in which the depth of the two subtrees of every node never differ by more than 1

<u>Solutions:</u>

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None


class Solution:
    # @param root, a tree node
    # @return a boolean
    def isBalanced(self, root):
        return self.isBalancedInt(root)>=0


    def isBalancedInt(self, root):
        if root == None:
            return 0;
        left = self.isBalancedInt(root.left)
        right = self.isBalancedInt(root.right)
        if left<0 or right<0 or abs(left-right)>1:
            return -1
        return max(left,right)+1
```

```
if __name__ == '__main__':
    BT, BT.right, BT.right.left = TreeNode(1), TreeNode(2), TreeNode(3)
    print ( Solution().isBalanced(BT) )
```