```java
/*Author: Bochen (mddboc@foxmail.com)
Last Modified: Tue Apr 10 22:28:44 CST 2018*/

/*
Given a binary tree, find its maximum depth.

        The maximum depth is the number of nodes along the longest path from the root node down to the farthest leaf node.

        For example:
        Given binary tree [3,9,20,null,null,15,7],

        3
        / \
        9  20
        /  \
        15   7
        return its depth = 3.*/


import java.util.*;
import java.lang.Math;
import java.lang.System;
import java.lang.Integer;


public class Main {

    public static void main(String[] args) throws ArithmeticException {

        String input = "ab";

        boolean answer = new Solution().repeatedSubstringPattern(input);

        System.out.println("haha");
    }

}


class ListNode {
    int val;
    ListNode next;

    ListNode(int x) {
        val = x;
    }
}


class TreeNode {
    int val;
    TreeNode left;
    TreeNode right;

    TreeNode(int x) {
        val = x;
    }
}


class Solution {
    public int maxDepth(TreeNode root) {

        if (root == null) {
            return 0;
        }

        if (root.left != null && root.right != null) {
            return 1 + Math.max(maxDepth(root.left), maxDepth(root.right));
        } else {
            if (root.left != null) {
                return 1 + maxDepth(root.left);
            } else if (root.right != null) {
                return 1 + maxDepth(root.right);
            } else {
                return 1;
            }
        }
    }
}
```