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

/*Rotate an array of n elements to the right by k steps.

    For example, with n = 7 and k = 3, the array [1,2,3,4,5,6,7] is rotated to [5,6,7,1,2,3,4].

    Note:
    Try to come up as many solutions as you can, there are at least 3 different ways to solve this problem.
*/



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 {

        TreeNode root = new TreeNode(1);
        root.left = new TreeNode(2);
        root.right = new TreeNode(2);
        root.left.left = new TreeNode(3);
        root.left.right = new TreeNode(4);
        root.right.left = new TreeNode(4);
        root.right.right = new TreeNode(3);

        boolean result = new Solution().isSymmetric(root);

        System.out.println(result);
    }

}


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 void rotate(int[] nums, int k) {

        // 方法一：需要开辟新空间，更快
        /* int numsLength = nums.length;
        k = k % numsLength;

        int[] helper = new int[k];

        System.arraycopy(nums, numsLength - k, helper, 0, k);
        System.arraycopy(nums, 0, nums, k, numsLength - k);
```

```
72          System.arraycopy(helper, 0, nums, 0, k); */
73  ⟶
74  ⟶      // 方法二：不需要开辟新空间，相对慢点
75          int numsLength = nums.length;
76          k = k % numsLength;
77
78          reverse(nums, 0, numsLength - 1);
79          reverse(nums, 0, k - 1);
80          reverse(nums, k, numsLength - 1);
81      }
82
83      private void reverse(int[] nums, int startIndex, int endIndex) {
84
85          while (startIndex < endIndex){
86              int temp = nums[startIndex];
87              nums[startIndex] = nums[endIndex];
88              nums[endIndex] = temp;
89
90              startIndex++;
91              endIndex--;
92          }
93      }
94  }
```