

```

1  /*Author: Bochen (mddboc@foxmail.com)
2  Last Modified: Tue Apr 10 22:28:44 CST 2018*/
3
4  /*Given a binary tree, find its minimum depth.
5
6  .....The minimum depth is the number of nodes along the shortest path from the
       root node down to the nearest leaf node.*/
7
8
9  import java.util.*;
10 import java.lang.Math;
11 import java.lang.System;
12 import java.lang.Integer;
13
14
15 public class Main {
16
17     ....public static void main(String[] args) throws ArithmeticException {
18
19         ....TreeNode root = new TreeNode(1);
20         ....root.left = new TreeNode(2);
21         ....root.right = new TreeNode(2);
22         ....root.left.left = new TreeNode(3);
23         ....root.left.right = new TreeNode(4);
24         ....root.right.left = new TreeNode(4);
25         ....root.right.right = new TreeNode(3);
26
27         ....boolean result = new Solution().isSymmetric(root);
28
29         ....System.out.println(result);
30     }
31
32 }
33
34
35 class ListNode {
36     ....int val;
37     ....ListNode next;
38
39     ....ListNode(int x) {
40         ....val = x;
41     }
42 }
43
44
45 class TreeNode {
46     ....int val;
47     ....TreeNode left;
48     ....TreeNode right;
49
50     ....TreeNode(int x) {
51         ....val = x;
52     }
53 }
54
55
56 class Solution {
57     ....public int minDepth(TreeNode root) {
58
59         ....if (root == null) {
60             ....return 0;
61         }
62
63         ....if (root.left == null) {
64             ....return 1 + minDepth(root.right);
65         }
66         ....else if (root.right == null) {
67             ....return 1 + minDepth(root.left);
68         }
69         ....else {
70             ....return 1 + Math.min(minDepth(root.left), minDepth(root.right));
71         }
72     }

```

