

```

1  /*Author: Bochen (mddboc@foxmail.com)
2  Last Modified: Tue Apr 10 22:28:44 CST 2018*/
3
4  /*Given an array of integers that is already sorted in ascending order, find two
   numbers such that they add up to a specific target number.
5
6  .....The function twoSum should return indices of the two numbers such that they
   add up to the target, where index1 must be less than index2. Please note
   that your returned answers (both index1 and index2) are not zero-based.
7
8  .....You may assume that each input would have exactly one solution and you may
   not use the same element twice.
9
10 .....Input: numbers={2, 7, 11, 15}, target=9
11 .....Output: index1=1, index2=2*/
12
13
14 import java.util.*;
15 import java.lang.Math;
16 import java.lang.System;
17 import java.lang.Integer;
18
19
20 public class Main {
21
22     ....public static void main(String[] args) throws ArithmeticException {
23
24         .....TreeNode root = new TreeNode(1);
25         .....root.left = new TreeNode(2);
26         .....root.right = new TreeNode(2);
27         .....root.left.left = new TreeNode(3);
28         .....root.left.right = new TreeNode(4);
29         .....root.right.left = new TreeNode(4);
30         .....root.right.right = new TreeNode(3);
31
32         .....boolean result = new Solution().isSymmetric(root);
33
34         .....System.out.println(result);
35     ....}
36
37 }
38
39
40 class ListNode {
41     ....int val;
42     ....ListNode next;
43
44     ....ListNode(int x) {
45         .....val = x;
46     ....}
47 }
48
49
50 class TreeNode {
51     ....int val;
52     ....TreeNode left;
53     ....TreeNode right;
54
55     ....TreeNode(int x) {
56         .....val = x;
57     ....}
58 }
59
60
61 class Solution {
62     ....public int[] twoSum(int[] numbers, int target) {
63
64         .....int startPoint = 0, endPoint = numbers.length - 1;
65
66         .....while (startPoint < endPoint) {
67             .....int sum = numbers[startPoint] + numbers[endPoint];
68
69             .....if (target == sum) {

```

```
70 ..... return new int[]{startPointer + 1, endPointer + 1};
71 .....} else if (target < sum) {
72 .....    endPointer--;
73 .....} else {
74 .....    startPointer++;
75 .....}
76 .....}
77 .....
78 ..... return null;
79 .....}
80 }
```