

```

1  /*Author: Bochen (mddboc@foxmail.com)
2  Last Modified: Tue Apr 10 22:28:44 CST 2018*/
3
4  /*Given an array where elements are sorted in ascending order, convert it to a
height balanced BST.
5
6  ..... 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.
7
8
9  ..... Example:
10
11 ..... Given the sorted array: [-10,-3,0,5,9],
12
13 ..... One possible answer is: [0,-3,9,-10,null,5], which represents the following
height balanced BST:
14
15 .....
16 ..... / \
17 ..... -3  9
18 ..... /  \
19 ..... -10 5*/
20
21
22 import java.util.*;
23 import java.lang.Math;
24 import java.lang.System;
25 import java.lang.Integer;
26
27
28 public class Main {
29
30     .... public static void main(String[] args) throws ArithmeticException {
31
32     ..... TreeNode root = new TreeNode(1);
33     ..... root.left = new TreeNode(2);
34     ..... root.right = new TreeNode(2);
35     ..... root.left.left = new TreeNode(3);
36     ..... root.left.right = new TreeNode(4);
37     ..... root.right.left = new TreeNode(4);
38     ..... root.right.right = new TreeNode(3);
39
40     ..... boolean result = new Solution().isSymmetric(root);
41
42     ..... System.out.println(result);
43     .... }
44
45 }
46
47
48 class ListNode {
49     .... int val;
50     .... ListNode next;
51
52     .... ListNode(int x) {
53     ..... val = x;
54     .... }
55 }
56
57
58 class TreeNode {
59     .... int val;
60     .... TreeNode left;
61     .... TreeNode right;
62
63     .... TreeNode(int x) {
64     ..... val = x;
65     .... }
66 }
67
68
69 class Solution {

```

```
70     public TreeNode sortedArrayToBST(int[] nums) {
71
72         if (nums == null || nums.length < 1)
73             return null;
74
75         int numsLength = nums.length;
76
77         return sortedArrayToBSTHelper(nums, 0, numsLength - 1);
78     }
79
80     private TreeNode sortedArrayToBSTHelper(int[] nums, int startIndex, int
        endIndex) {
81         if (startIndex > endIndex) {
82             return null;
83         }
84
85         int mid = (startIndex + endIndex) / 2;
86         TreeNode root = new TreeNode(nums[mid]);
87
88         root.left = sortedArrayToBSTHelper(nums, startIndex, mid - 1);
89         root.right = sortedArrayToBSTHelper(nums, mid + 1, endIndex);
90
91         return root;
92     }
93 }
```