**难度: 简单**

实现一个函数，检查二叉树是否平衡。在这个问题中，平衡树的定义如下：任意一个节点，其两棵子树的高度差不超过 1。

示例 1:

给定二叉树 [3,9,20,null,null,15,7]

3

/ \

9 20

/ \

15 7

返回 true 。

示例 2:

给定二叉树 [1,2,2,3,3,null,null,4,4]

1

/ \

2 2

/ \

3 3

/ \

4 4

返回 false 。

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/check-balance-lcci

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路:**

1. 定义一个结果变量

2. 获取左边的深度

3. 获取右边的深度

4. 比较左右深度相减是否大于1

如果大于1,把定义的结果变量变成false

没有大于1, 就返回最大值就行了

|  |
| --- |
| //定义一个变量  boolean result = true;  public boolean isBalanced(TreeNode root) {  getDept(root,0);  return result;  }  private int getDept(TreeNode root, int dept) {  //如果为空返回深度  if (root == null) {  return dept;  }  //获取左边的深度, 深度+1  int leftDept = getDept(root.left, dept + 1);  //获取右边的深度, 深度+1  int rightDept = getDept(root.right, dept + 1);  //判断左右深度是否大于1  if (Math.abs(leftDept - rightDept) > 1) {  result = false;  }  //返回最大值的深度  return Math.max(leftDept, rightDept);  } |