

# Problem 958: Check Completeness of a Binary Tree

## Problem Information

**Difficulty:** Medium

**Acceptance Rate:** 58.81%

**Paid Only:** No

**Tags:** Tree, Breadth-First Search, Binary Tree

## Problem Description

Given the `root` of a binary tree, determine if it is a \_complete binary tree\_.

In a \*\*[complete binary tree]([http://en.wikipedia.org/wiki/Binary\\_tree#Types\\_of\\_binary\\_trees](http://en.wikipedia.org/wiki/Binary_tree#Types_of_binary_trees))\*\* , every level, except possibly the last, is completely filled, and all nodes in the last level are as far left as possible. It can have between `1` and `2<sup>h</sup>` nodes inclusive at the last level `h`.

**Example 1:**



**Input:** root = [1,2,3,4,5,6] **Output:** true **Explanation:** Every level before the last is full (ie. levels with node-values {1} and {2, 3}), and all nodes in the last level ({4, 5, 6}) are as far left as possible.

**Example 2:**



**Input:** root = [1,2,3,4,5,null,7] **Output:** false **Explanation:** The node with value 7 isn't as far left as possible.

**Constraints:**

\* The number of nodes in the tree is in the range `[1, 100]`. \* `1 <= Node.val <= 1000`

## Code Snippets

### C++:

```
/**
 * Definition for a binary tree node.
 * struct TreeNode {
 *     int val;
 *     TreeNode *left;
 *     TreeNode *right;
 *     TreeNode() : val(0), left(nullptr), right(nullptr) {}
 *     TreeNode(int x) : val(x), left(nullptr), right(nullptr) {}
 *     TreeNode(int x, TreeNode *left, TreeNode *right) : val(x), left(left),
 *     right(right) {}
 * };
 */
class Solution {
public:
    bool isCompleteTree(TreeNode* root) {
        }
    };
}
```

### Java:

```
/**
 * Definition for a binary tree node.
 * public class TreeNode {
 *     int val;
 *     TreeNode left;
 *     TreeNode right;
 *     TreeNode() {}
 *     TreeNode(int val) { this.val = val; }
 *     TreeNode(int val, TreeNode left, TreeNode right) {
 *         this.val = val;
 *         this.left = left;
 *         this.right = right;
 *     }
 * }
```

```
*/  
class Solution {  
public boolean isCompleteTree(TreeNode root) {  
  
}  
}  
}
```

### Python3:

```
# Definition for a binary tree node.  
# class TreeNode:  
#     def __init__(self, val=0, left=None, right=None):  
#         self.val = val  
#         self.left = left  
#         self.right = right  
class Solution:  
    def isCompleteTree(self, root: Optional[TreeNode]) -> bool:
```