2792. Count Nodes That Are Great Enough Premium

Hard Topics Hint

You are given a `root` to a binary tree and an integer `k`. A node of this tree is called **great enough** if the followings hold:

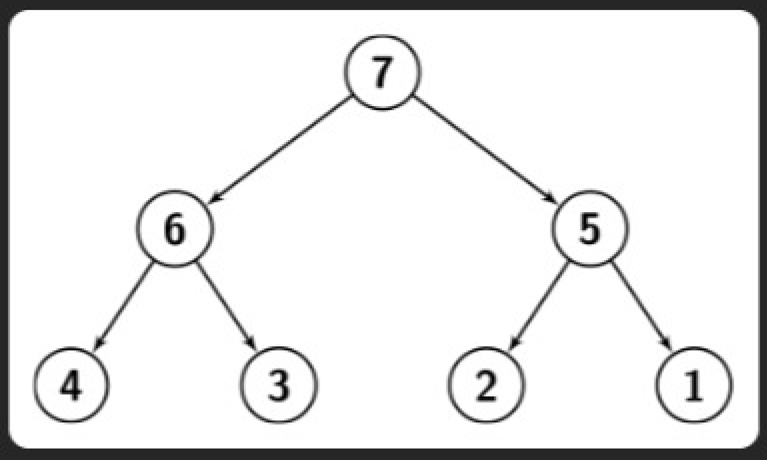
- Its subtree has **at least** `k` nodes.
- Its value is **greater** than the value of **at least** `k` nodes in its subtree.

Return *the number of nodes in this tree that are great enough.*

The node `u` is in the **subtree** of the node `v`, if `u == v` or `v` is an ancestor of `u`.

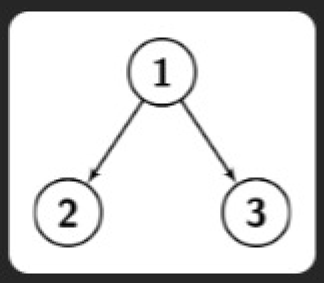
Example 1:

```
Input: root = [7,6,5,4,3,2,1], k = 2
Output: 3
Explanation: Number the nodes from 1 to 7.
The values in the subtree of node 1: {1,2,3,4,5,6,7}. Since node.val == 7, there are 6 nodes having a smaller value than its value. So it's great enough.
The values in the subtree of node 2: {3,4,6}. Since node.val == 6, there are 2 nodes having a smaller value than its value. So it's great enough.
The values in the subtree of node 3: {1,2,5}. Since node.val == 5, there are 2 nodes having a smaller value than its value. So it's great enough.
It can be shown that other nodes are not great enough.
See the picture below for a better understanding.
```



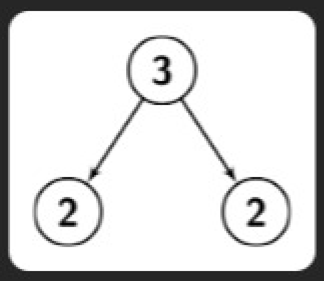
Example 2:

```
Input: root = [1,2,3], k = 1
Output: 0
Explanation: Number the nodes from 1 to 3.
The values in the subtree of node 1: {1,2,3}. Since node.val == 1, there are no nodes having a smaller value than its value. So it's not great enough.
The values in the subtree of node 2: {2}. Since node.val == 2, there are no nodes having a smaller value than its value. So it's not great enough.
The values in the subtree of node 3: {3}. Since node.val == 3, there are no nodes having a smaller value than its value. So it's not great enough.
See the picture below for a better understanding.
```



Example 3:

```
Input: root = [3,2,2], k = 2
Output: 1
Explanation: Number the nodes from 1 to 3.
The values in the subtree of node 1: {2,2,3}. Since node.val == 3, there are 2 nodes having a smaller value than its value. So it's great enough.
The values in the subtree of node 2: {2}. Since node.val == 2, there are no nodes having a smaller value than its value. So it's not great enough.
The values in the subtree of node 3: {2}. Since node.val == 2, there are no nodes having a smaller value than its value. So it's not great enough.
See the picture below for a better understanding.
```



Constraints:

- The number of nodes in the tree is in the range [1, 10^4].
- 1 <= Node.val <= 10^4
- `1 <= k <= 10`

Seen this question in a real interview before? 1/5

Yes No

Accepted 945 | Submissions 1.6K | Acceptance Rate 58.0%

Topics

Divide and Conquer Tree Depth-First Search Binary Tree

Hint 1

For each node, calculate a list of `k` values representing `k` smallest values in the subtree of that node.

Hint 2

To check if a node is great enough, get the described list in the first hint for its children and merge them. Since the resulting list may contain more than `k` elements, pick `k` smallest values and discard the extra ones.

Hint 3

Now check if the merged list has exactly `k` elements, and the current node's value is greater than the greatest element in the list, then that node is great enough.

Discussion (0)

Copyright © 2024 LeetCode All rights reserved