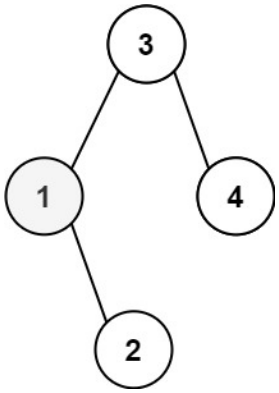
# 230. Kth Smallest Element in a BST

Given the root of a binary search tree, and an integer k, return *the* $k^{th}$ *smallest value (**1-indexed**) of all the values of the nodes in the tree*.

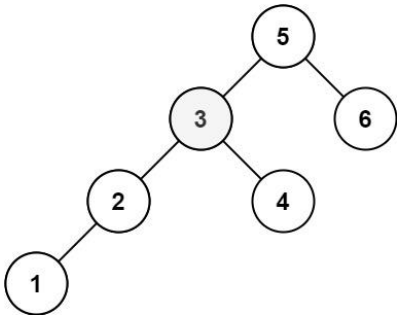
**Example 1:**




**Input:** root = [3,1,4,null,2], k = 1

**Output:** 1

**Example 2:**




**Input:** root = [5,3,6,2,4,null,null,1], k = 3

**Output:** 3

**Constraints:**

- The number of nodes in the tree is n.
- $1 <= k <= n <= 10^4$
- $0 <= Node.val <= 10^4$

```
# Definition for a binary tree node.
# class TreeNode(object):
#     def __init__(self, val=0, left=None, right=None):
#         self.val = val
#         self.left = left
#         self.right = right
class Solution(object):
    def kthSmallest(self, root, k):
        values = []
        self.inorder(root, values)
        return values[k-1]

    def inorder(self,root,values):
        if not root:
            return
        self.inorder(root.left, values)
        values.append(root.val)
        self.inorder(root.right, values)
```