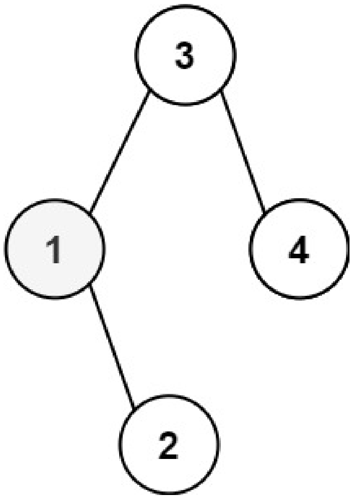
# 230. Kth Smallest Element in a BST

| | |
|---|---|
| Created | @September 18, 2022 3:34 PM |
| Difficulty | Medium |
| LC Url | https://leetcode.com/problems/kth-smallest-element-in-a-bst/ |
| Importance | |
| Tag | BST NEET Recursion Tree |
| Video | https://www.youtube.com/watch?v=5LUXSvjmGCw |

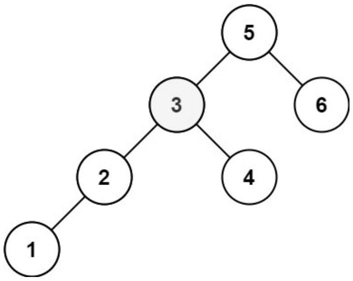
Given the `root` of a binary search tree, and an integer `k`, return *the* `k th` *smallest value (**1-indexed**) of all the values of the nodes in the tree*.

**Example 1:**



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

**Example 2:**



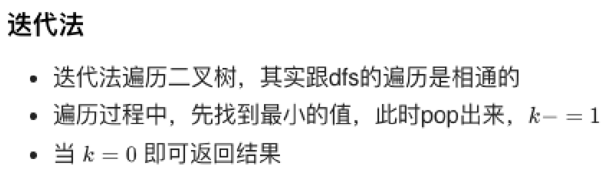
```
Input: root = [5,3,6,2,4,null,null,1], k = 3
Output: 3
```

**Constraints:**

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

**Follow up:** If the BST is modified often (i.e., we can do insert and delete operations) and you need to find the kth smallest frequently, how would you optimize?

# Solution



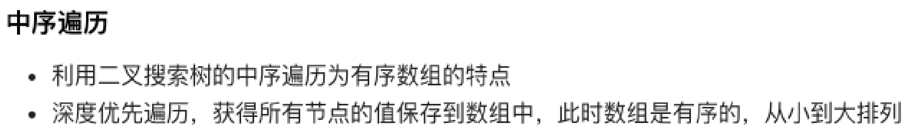
### 迭代法

- 迭代法遍历二叉树，其实跟dfs的遍历是相通的
- 遍历过程中，先找到最小的值，此时pop出来，$k-=1$
- 当 $k=0$ 即可返回结果

```
# 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 kthSmallest(self, root: Optional[TreeNode], k: int) -> int:
        stack = []
        while root or stack:
            while root:
                stack.append(root)
                root = root.left
            root = stack.pop()
            k -= 1
            if k == 0:
                return root.val
            root = root.right

        # 链接：https://leetcode.cn/problems/kth-smallest-element-in-a-bst/solution/er-cha-sou-suo-shu-zhong-di-kxiao-de-yua-8o07/
```

https://leetcode.cn/problems/kth-smallest-element-in-a-bst/solution/chi-xiao-dou-nojie-ti-python-dfszhong-xu-m5ql/



### 中序遍历

- 利用二叉搜索树的中序遍历为有序数组的特点
- 深度优先遍历，获得所有节点的值保存到数组中，此时数组是有序的，从小到大排列

```
# 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 kthSmallest(self, root: Optional[TreeNode], k: int) -> int:

        def dfs(node, res):

            if node is None:
                return
            dfs(node.left, res)
            res.append(node.val)
            dfs(node.right, res)

        res = []
        dfs(root, res)
        return res[k-1]


作者：niconiconi-12
链接：https://leetcode.cn/problems/kth-smallest-element-in-a-bst/solution/chi-xiao-dou-nojie-ti-python-dfszhong-xu-m5ql/
来源：力扣（LeetCode）
著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。
```