448 · Inorder Successor in BST

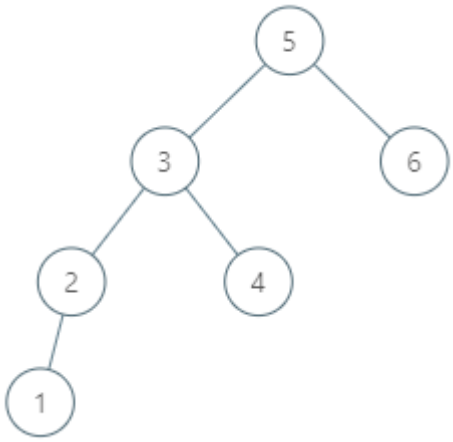
Description

Given a binary search tree (See Definition) and a node in it, find the in-order successor of that node in the BST.

If the given node has no in-order successor in the tree, return `null`.

It's guaranteed *p* is one node in the given tree. (You can directly compare the memory address to find p)

The successor of a node `p` is the node with the smallest key greater than `p.val`.



Input: root = [5,3,6,2,4,null,null,1], p = 4
Output: 5
**Explanation: Self explanatory. Traverse Inorderly manner.
**

1. If the given node has no in-order successor in the tree, return `null`.
2. It's guaranteed that the values of the tree are unique.

```
class Solution:
    """
    @param: root: The root of the BST.
    @param: p: You need find the successor node of p.
    @return: Successor of p.
    """
    def inorderSuccessor(self, root, p):
        # write your code here
        if root is None:
```

```
            return None
        if root.left is root.right:
            return None
        ans = []
        self.helper(root,p,ans)
        return ans[0]
    def helper(self,root,p,ans):
        if root is None:
            return
        self.helper(root.left,p,ans)
        if root.val>p.val and len(ans)==0:
            ans.append(root)
        self.helper(root.right,p,ans)
```