Recover Binary Search Tree

Question: Two elements of a binary search tree (BST) are swapped by mistake.

Recover the tree without changing its structure.

Note: A solution using O(n) space is pretty straight forward. Could you devise a constant space solution?

Solutions:

```
class TreeNode:
    def __init__(self, x):
        self.val = x
        self.left = None
        self.right = None


class Solution:
    # @param root, a tree node
    # @return a tree node
    def FindTwoNodes(self, root):
            if root:
                self.FindTwoNodes(root.left)
                if self.prev and self.prev.val > root.val:
                    self.n2 = root
                    if self.n1 == None: self.n1 = self.prev
                self.prev = root
                self.FindTwoNodes(root.right)
    def recoverTree(self, root):
```

```
self.n1 = self.n2 = None
self.prev = None
self.FindTwoNodes(root)
self.n1.val, self.n2.val = self.n2.val, self.n1.val
return root
```