

# 99. 恢复二叉搜索树

二叉搜索树中的两个节点被错误地交换。

请在不改变其结构的情况下，恢复这棵树。

- 使用  $O(n)$  空间复杂度的解法很容易实现。
- 你能想出一个只使用常数空间的解决方案吗？

输入: [1,3,null,null,2]

```

    1
   /
  3
   \
    2

```

输出: [3,1,null,null,2]

```

    3
   /
  1
   \
    2

```

输入: [3,1,4,null,null,2]

```

    3
   / \
  1   4
   /
  2

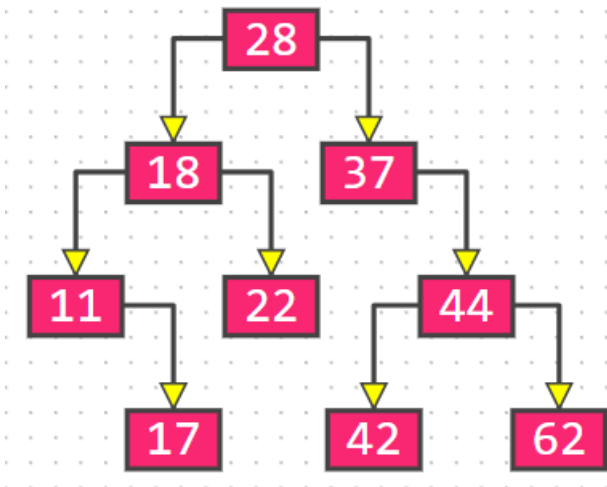
```

输出: [2,1,4,null,null,3]

```

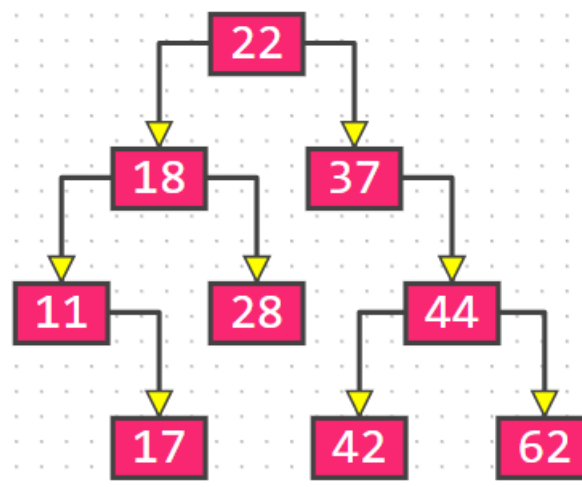
    2
   / \
  1   4
   /
  3

```



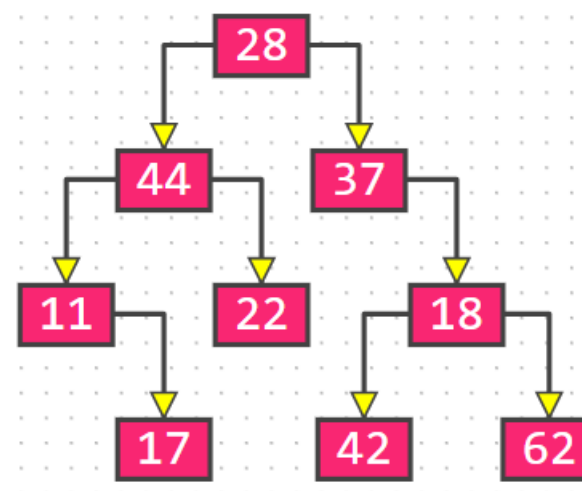
■ 中序遍历的结果是升序的

□ 11, 17, 18, 22, 28, 37, 42, 44, 62



■ 中序遍历的结果

□ 11, 17, 18, 28, 22, 37, 42, 44, 62



■ 中序遍历的结果

□ 11, 17, 44, 22, 28, 37, 42, 18, 62

■ 第1个错误节点: 第1个逆序对中的较大节点

■ 第2个错误节点: 最后1个逆序对中的较小节点

# 二叉树的Morris遍历

- 使用Morris方法遍历二叉树，可以实现时间复杂度 $O(n)$ 、空间复杂度 $O(1)$
- 这里只演示二叉树的Morris中序遍历。前序遍历、后序遍历在此基础上做一些调整即可

■ 执行步骤（假设遍历到当前节点是N）

① 如果 $N.left \neq null$ ，找到N的前驱节点P

□ 如果 $P.right == null$

✓  $P.right = N$

✓  $N = N.left$

✓ 回到①

□ 如果 $P.right == N$

✓  $P.right = null$

✓ 打印N

✓  $N = N.right$

✓ 回到①

② 如果 $N.left == null$

□ 打印N

□  $N = N.right$

□ 回到①

③ 重复①、②直到 $N == null$

