Description Solution Discuss (999+) Submissions

# 426. Convert Binary Search Tree to Sorted Doubly Linked List

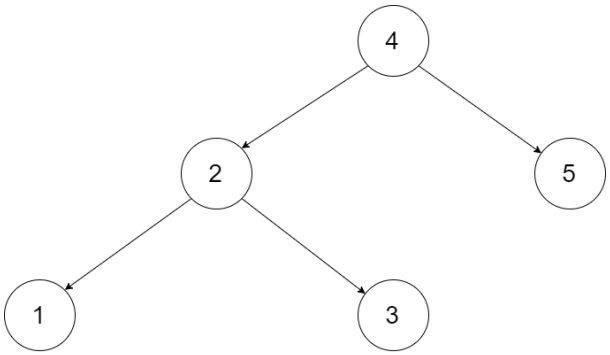
Medium 2374 184 Add to List Share

Convert a **Binary Search Tree** to a sorted **Circular Doubly-Linked List** in place.

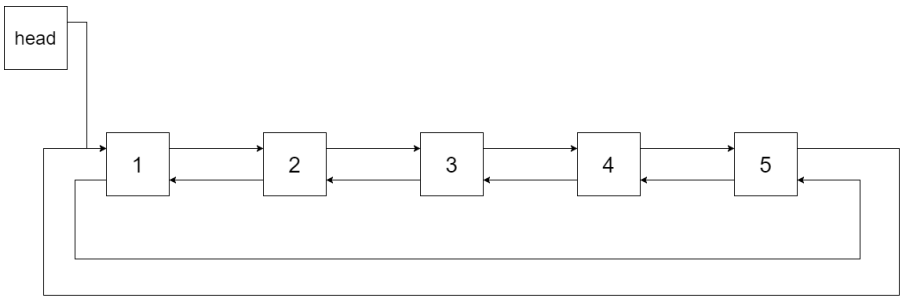
You can think of the left and right pointers as synonymous to the predecessor and successor pointers in a doubly-linked list. For a circular doubly linked list, the predecessor of the first element is the last element, and the successor of the last element is the first element.

We want to do the transformation **in place**. After the transformation, the left pointer of the tree node should point to its predecessor, and the right pointer should point to its successor. You should return the pointer to the smallest element of the linked list.

**Example 1:**




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




**Output:** [1,2,3,4,5]

**Explanation:** The figure below shows the transformed BST. The solid line indicates the successor relationship, while the

```
>right,back,f

33
34          if(bc
35 ▾        {
36              r
back;
37              b
root;
38          }
39          else
40 ▾        {
41              f
42          }
43
44          back
45
46          conve
>left,back,fr
47      }
48
49 ▾    Node*
treeToDoublyL
{
50          if(!r
51              r
52
53          Node*
54          Node*
55
56
 convertor(rc
57          back-
58          front
59          retur
60      }
```

Your previous code was re

Testcase Run Code Resu

Accepted Runtime

Your input [3]

Output [3]

Expected [3]

Problems Pick One Prev 426/2493 Next xample ases Run Code