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

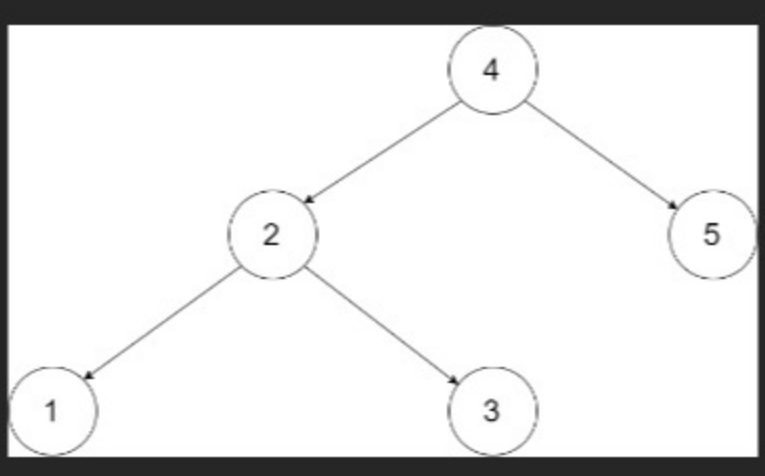
Medium | Topics | Companies

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

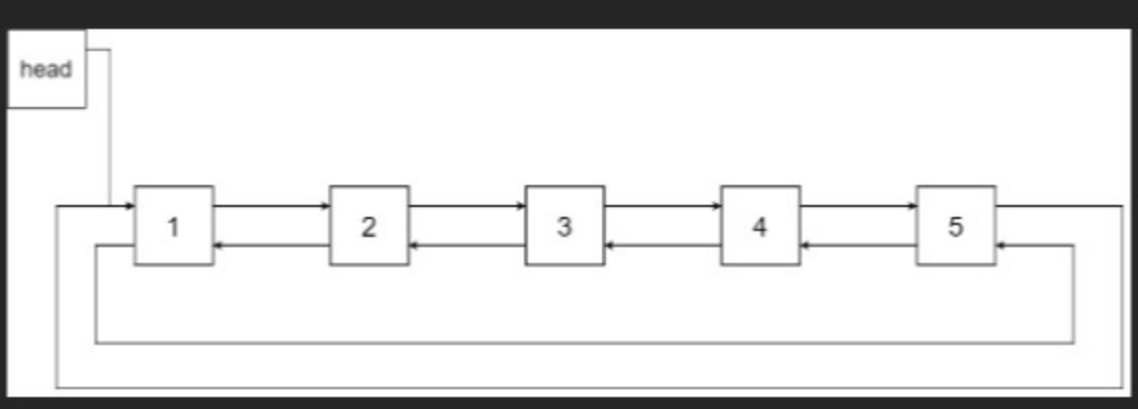
You can think of the left and right pointers as synonymous to the predecessor and successor pointers in a doubly-linked list. For a circular doubly linked list, the predecessor of the first element is the last element, and the successor of the last element is the first element.

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:

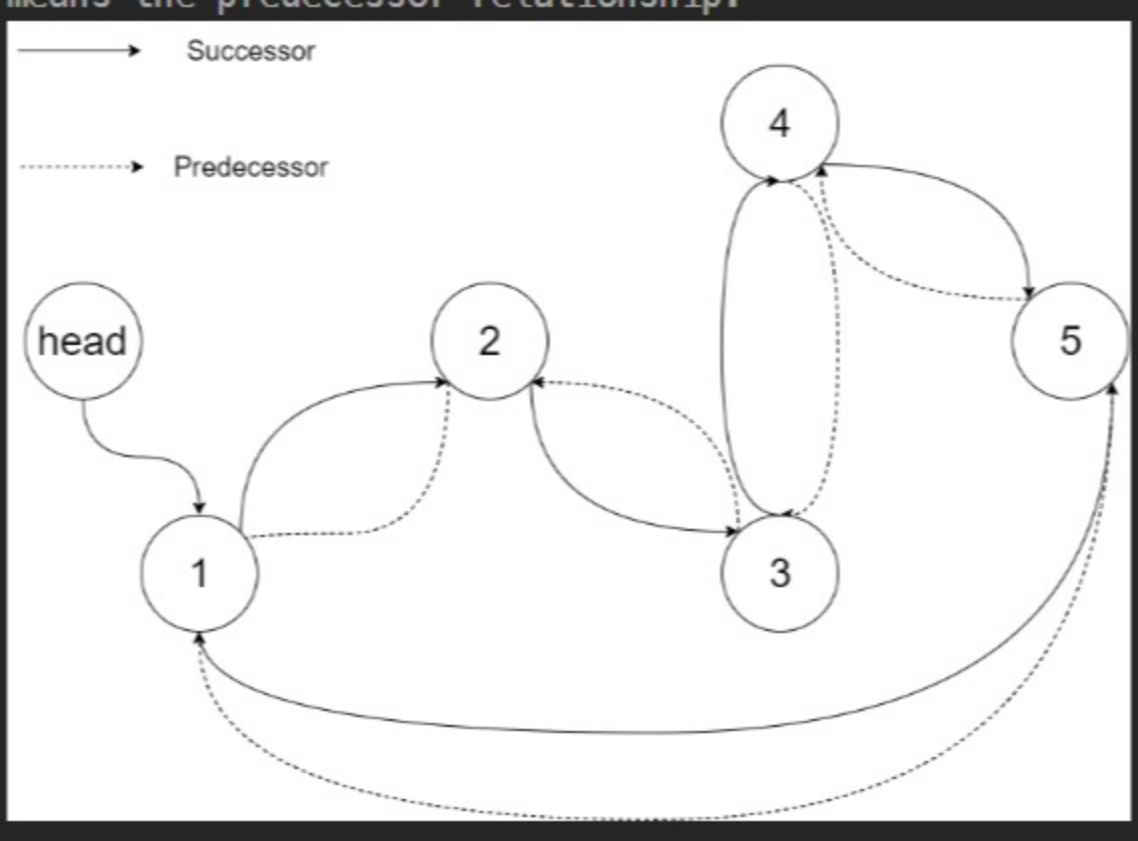


Input: root = [4,2,5,1,3]



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

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



Example 2:

Input: root = [2,1,3]
Output: [1,2,3]

Constraints:

- The number of nodes in the tree is in the range [0, 2000].
- -1000 <= Node.val <= 1000
- All the values of the tree are **unique**.

Seen this question in a real interview before? 1/5

Yes No

Accepted 314K | Submissions 482.6K | Acceptance Rate 65.1%

Topics

Linked List | Stack | Tree | Depth-First Search | Binary Search Tree | Binary Tree | Doubly-Linked List

Companies

0 - 3 months

Meta 20 | TikTok 2

6 months ago

Microsoft 4 | Amazon 2 | Nvidia 2

Similar Questions

Binary Tree Inorder Traversal Easy

Discussion (19)

Copyright © 2024 LeetCode All rights reserved