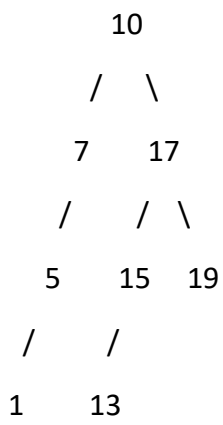
Flatten Binary Search Tree to a Sorted Linked List

CodeStudio

Given the root of a binary search tree, flatten the tree into a "linked list":

- The "linked list" should use the same Node class where the right child pointer points to the next node in the list and the left child pointer is always null.
- The "linked list" should be in the same order as a preorder traversal of the binary tree.

Example:



Output:

1 → 5 → 7 → 10 → 13 → 15 → 17 → 19 → NULL

Approach 1: This function flattens a binary search tree (BST) into a sorted linked list, by performing in-order traversal and storing the values in the 'inorderAns' vector

- **Function Purpose:** This approach flattens a BST into a sorted linked list using in-order traversal and Morris Traversal.
- **Explanation:**
    1. Perform in-order traversal using Morris Traversal and store the nodes in a vector. This vector temporarily uses additional space.
    2. Create a new linked list from the values obtained during in-order traversal.
- **Time Complexity: The time complexity is O(N), where N is the number of nodes in the tree, as it visits each node once.**
- **Space Complexity: The space complexity is O(N) due to the additional space used by the vector to store the nodes.**