**Alice and Merg Game**

You are given two binary search trees (BSTs), T1 and T2, which may contain overlapping nodes. Write code in your preferred programming language to merge these two BSTs into a single BST. The merged BST should include all the nodes from T1 and T2 with their values in sorted order.

**Input Format:**

* First line contains the values of all the nodes in the binary tree T1 in pre-order format where true suggests the node exists and false suggests it is NULL.
* Second line contains the values of all the nodes in the binary tree T2 in pre-order format where true suggests the node exists and false suggests it is NULL.

**Output Format:**

* Print the merged BST in pre-order format where true suggests the node exists and false suggests it is NULL.

**Constraints:**

* Each tree may contain duplicate values.
* Assume that there are no duplicate values between T1 and T2.
* You should not modify the structure of either tree; instead, construct a new tree.

**Hints:**

* Consider using an iterative approach to merge the trees while maintaining the BST properties.
* Utilize in-order traversal to collect nodes in sorted order and construct the merged BST.
* This problem challenges you to not only understand BST properties but also to merge two tree structures while ensuring the resulting tree maintains the BST property and is efficiently constructed.

**Sample Input Test Case 1:**

arduino

Copy code

8 true 5 true 2 false false true 7 false false

10 true 15 true 12 false false true 18 false false

**Sample Output Test Case 1:**

arduino

Copy code

10 true 8 true 5 true 2 false false true 7 false false false true 15 true 12 false false true 18 false false

**Sample Input Test Case 2:**

arduino

Copy code

10 true 5 true 3 false false true 7 false false

15 true 12 true 11 false false true 18 false false

**Sample Output Test Case 2:**

arduino

Copy code

15 true 10 true 5 true 3 false false true 7 false false true 12 true 11 false false false true 18 false false

**Explanation:** In the merged BST, all nodes from both trees are included and sorted. Note that the value 4 is chosen as the root since it's the median value between the minimum value in T2 and the maximum value in T1.

**Extra Test Cases:**

**Input Test Case 3:**

arduino

Copy code

7 true 3 true 2 false false true 5 false false

8 true 6 true 4 false false true 9 false false

**Output Test Case 3:**

arduino

Copy code

8 true 7 true 6 true 3 true 2 false false true 5 false false true 4 false false true 9 false false

**Input Test Case 4:**

arduino

Copy code

5 true 3 true 2 false false true 4 false false

6 true 7 true 8 false false true 9 false false

**Output Test Case 4:**

arduino

Copy code

6 true 5 true 3 true 2 false false true 4 false false true 7 true 8 false false true 9 false false

**Input Test Case 5:**

arduino

Copy code

10 true 5 true 2 false false true 7 false false

10 true 5 true 2 false false true 7 false false

**Output Test Case 5:**

arduino

Copy code

10 true 10 true 5 true 5 true 2 false false true 2 false false true 7 false false true 7 false false

Ensure to construct the merged BST using linked lists only