



5810. Merge BSTs to Create Single BST

[My Submissions \(/contest/weekly-contest-249/problems/merge-bsts-to-create-single-bst/submissions/\)](/contest/weekly-contest-249/problems/merge-bsts-to-create-single-bst/submissions/)

[Back to Contest \(/contest/weekly-contest-249/\)](/contest/weekly-contest-249/)

You are given n **BST (binary search tree) root nodes** for n separate BSTs stored in an array `trees` (**0-indexed**). Each BST in `trees` has **at most 3 nodes**, and no two roots have the same value. In one operation, you can:

- Select two **distinct** indices i and j such that the value stored at one of the **leaves** of `trees[i]` is equal to the **root value** of `trees[j]`.
- Replace the leaf node in `trees[i]` with `trees[j]`.
- Remove `trees[j]` from `trees`.

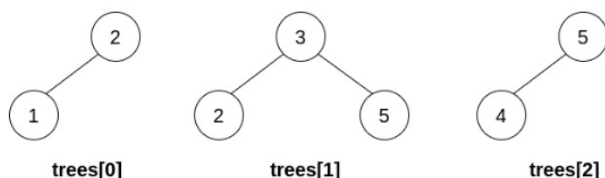
Return the **root** of the resulting BST if it is possible to form a valid BST after performing $n - 1$ operations, or `null` if it is impossible to create a valid BST.

A BST (binary search tree) is a binary tree where each node satisfies the following property:

- Every node in the node's left subtree has a value **strictly less** than the node's value.
- Every node in the node's right subtree has a value **strictly greater** than the node's value.

A leaf is a node that has no children.

Example 1:



User Accepted: 0

User Tried: 0

Total Accepted: 0

Total Submissions: 0

Difficulty: **Hard**

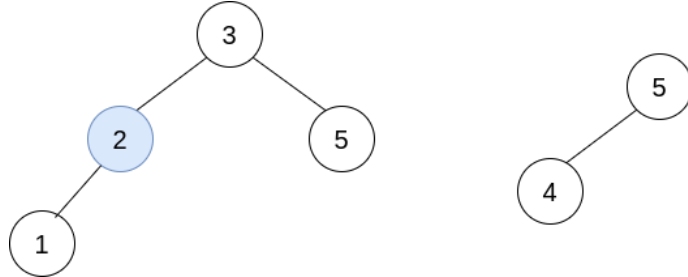
Input: trees = [[2,1],[3,2,5],[5,4]]

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

Explanation:

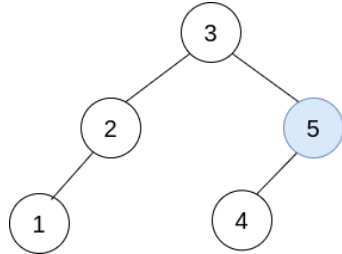
In the first operation, pick $i=1$ and $j=0$, and merge trees[0] into trees[1].

Delete trees[0], so trees = [[3,2,5,1],[5,4]].



In the second operation, pick $i=0$ and $j=1$, and merge trees[1] into trees[0].

Delete trees[1], so trees = [[3,2,5,1,null,4]].



The resulting tree, shown above, is a valid BST, so return its root.

Example 2:



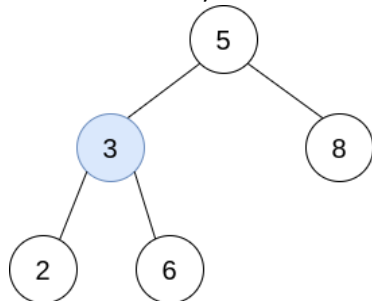
Input: trees = [[5,3,8],[3,2,6]]

Output: []

Explanation:

Pick $i=0$ and $j=1$ and merge trees[1] into trees[0].

Delete trees[1], so trees = [[5,3,8,2,6]].



The resulting tree is shown above. This is the only valid operation that can be performed, but the resulting tr

Example 3:

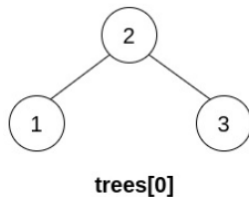


Input: trees = [[5,4],[3]]

Output: []

Explanation: It is impossible to perform any operations.

Example 4:



Input: trees = [[2,1,3]]

Output: [2,1,3]

Explanation: There is only one tree, and it is already a valid BST, so return its root.

Constraints:

- $n == \text{trees.length}$
- $1 \leq n \leq 5 \times 10^4$
- The number of nodes in each tree is in the range $[1, 3]$.
- No two roots of trees have the same value.
- All the trees in the input are **valid BSTs**.
- $1 \leq \text{TreeNode.val} \leq 5 \times 10^4$.

JavaScript



```

1 ▾ /**
2  * Definition for a binary tree node.
3  * function TreeNode(val, left, right) {
4  *     this.val = (val===undefined ? 0 : val)
5  *     this.left = (left===undefined ? null : left)
6  *     this.right = (right===undefined ? null : right)
7  * }
8  */
9 ▾ /**
10  * @param {TreeNode[]} trees
11  * @return {TreeNode}
12  */
13 ▾ var canMerge = function(trees) {
14
15     };

```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)

Copyright © 2021 LeetCode

[Help Center \(/support\)](#) | [Jobs \(/jobs\)](#) | [Bug Bounty \(/bugbounty\)](#) | [Online Interview \(/interview/\)](#) | [Students \(/student\)](#) | [Terms \(/terms\)](#) |

[Privacy Policy \(/privacy\)](#)

 [United States \(/region\)](#)