Congratulations! You passed!
TO PASS 75% or higher

Keep Learning

GRADE
100%

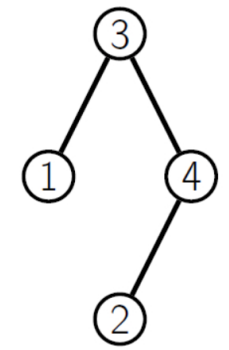
Binary Search Trees

TOTAL POINTS 4

1. **Your colleague proposed a different definition of a binary search tree: it is such binary tree with keys in the nodes that for each node the key of its left child (if exists) is less than its key, and the key of its right child (if exists) is bigger than its key. Is this a good definition for a binary search tree?** 1 / 1 point

- ( ) Yes
- (•) No

Correct
Correct! Binary search for a key in such tree might not work: try finding the key 2 in the tree below. Your first move would be to go left from the root, because $2 < 3$. However, the key 2 is in the right subtree of the root. The condition that the key of the left child of each node is less than the key of the parent, and the key of the right child is greater than the key of the parent is satisfied for all nodes in the tree. What's missing is that not just the key of the left child, but the keys of the whole left subtree of each node should be less than the key of the subtree root, and similarly for the right subtree.



2. **Suppose we enforce the AVL tree condition only for the root of the tree, but not for all the nodes. Can such tree be unbalanced?** 1 / 1 point

- (•) Yes
- ( ) No

Correct
Correct! Such tree could have height $\frac{n}{2}$ where n is the number of nodes: two chains of length $\frac{n}{2}$ having root as the only common node form such a tree.

3. **Can the Insert operation be implemented given only Split and Merge operations?** 1 / 1 point

- ( ) Yes. First create a new tree with single key - the key to be inserted. Then merge current tree with the new tree.
- ( ) No
- (•) Yes. First create a new tree with single key - the key to be inserted. Then split the current tree by this key. Then merge the left splitted part with the new tree. Then merge the result with the right splitted part.

Correct
Correct!

4. **Can the Delete operation be implemented given only Split and Merge operations?** 1 / 1 point

- (•) Yes. Suppose we are deleting key x. Split by the key twice: one split such that all the keys $< x$ go to the left left and all the keys $\geq x$ go to the right. Then split the right part of the first split such that all the keys $\leq x$ go to the left and all the keys $> x$ go to the right. Then merge the left part of the first split and the right part of the second split - thus leaving out the node with key x if there was such a node.
- ( ) Yes. Suppose we are deleting key x. Split by the key twice: one split such that all the keys $< x$ go to the left left and all the keys $\geq x$ go to the right. Then split the right part of the first split such that all

https://github.com/MalikSaurabh

the keys $\leq x$ go to the left and all the keys $> x$ go to the right. Then merge everything back.

- ○ No

✓ Correct

https://github.com/MalikSaurabh