uc3m | Universidad Carlos III de Madrid

Data Structures and Algorithms.
Author: Isabel Segura Bedmar

Unit 5 – Trees

Problem - Implementation of a Binary Tree. This implementation should include the following methods: size, height, depth, preorder, inorder, postorder and levelorder, studied in class.

Problem - Implementation of a Binary Search Tree. This implementation should include the following methods: search, insert and remove, studied in class.

Problem: Implement an iterative method that returns the smallest element in the tree.

Problem: Implement an iterative method that returns the maximum element in the tree.

Problem: Implement a recursive method that sums all the elements in the tree and returns this result.

Problem: Implement a recursive method that visits all the nodes and prints those whose grandparent's element is multiple of 10.

Problem: Implement an iterative method that takes a binary search node and returns its predecessor node from its left subtree.

Problem: Implement an iterative method that takes a binary search node and returns its successor node from its right subtree.

Problem: Implement a new version of the remove method, where the node's element to be removed is replaced by using its predecessor instead of using its successor in the tree.

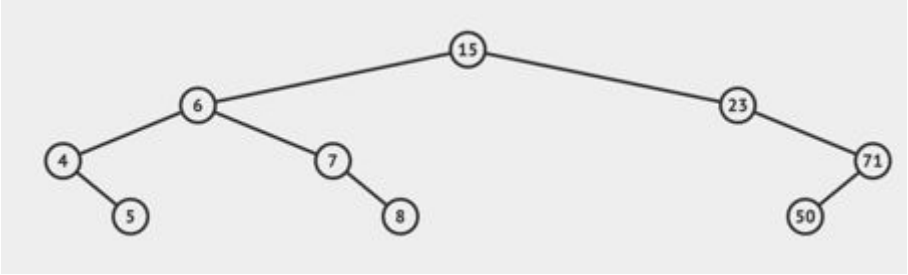
Problem: Implement a method that takes a binary search node and returns its size balance factor. The size balance factor of a node is the difference between size of the left subtree and the size of the right subtree

Problem: Implement a method that takes a binary search node and returns its height balance factor. The height balance factor of a node is the difference between height of the left subtree and the height of the right subtree

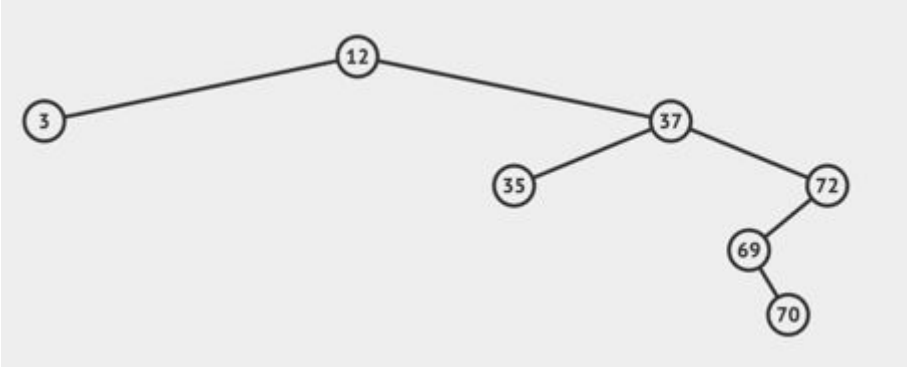
Problem: Implement a method that checks if the tree is size balanced. A BST is size balanced if all its nodes have a size balance factor less or equal to 1.

Problem: Implement a method that checks if the tree is height balanced (AVL). A BST is an AVL if all its nodes have a height balance factor less or equal to 1.

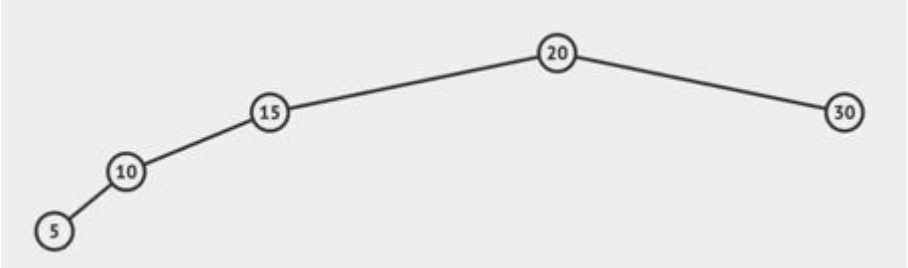
Problem: Is it a size-balanced binary search tree (BST)?. If it is not, please balance it.



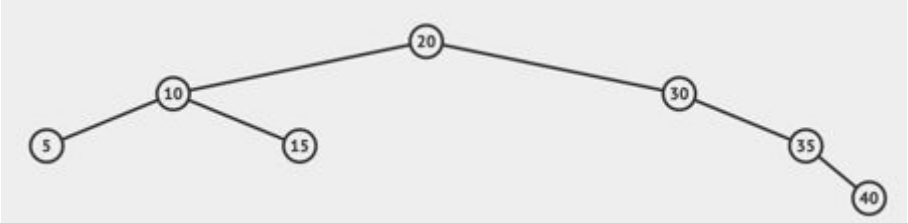
Problem: Is it a size-balanced binary search tree (BST)?. If it is not, please balance it.



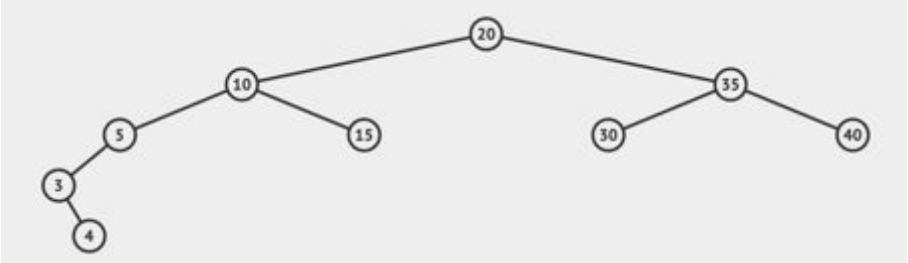
Problem: Is it a height-balanced tree?. If it is not, please balance it.



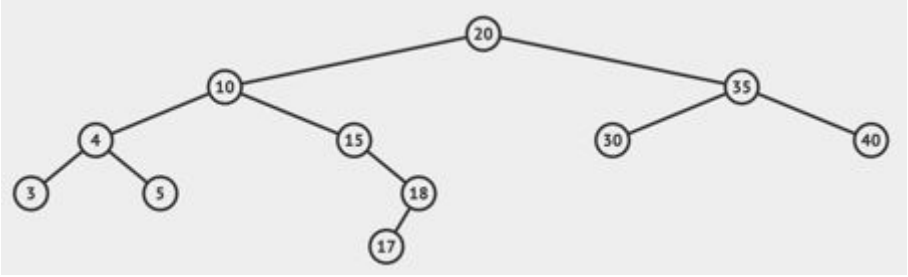
Problem: Please, transform the following tree to its height-balanced tree.



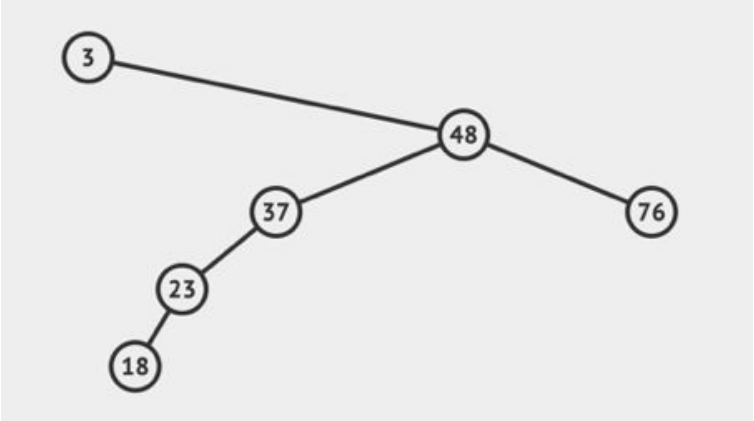
Problem: Please, transform the following tree to its height-balanced tree.



Problem: Please, transform the following tree to its height-balanced tree.



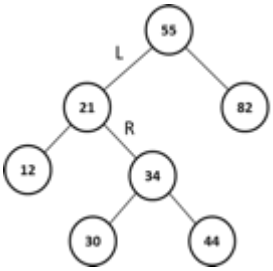
Problem: Please, transform the following tree to its height-balanced tree.



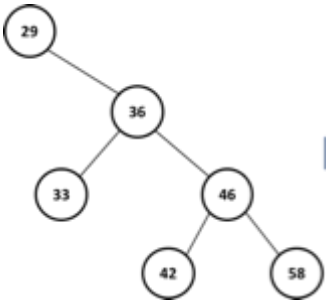
Problems:

Given the following binary search trees,

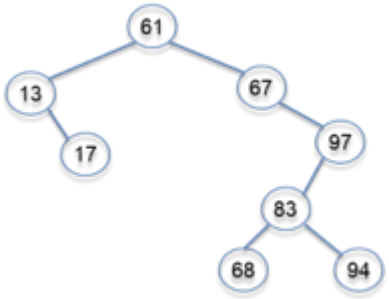
1)



2)



3)



Transform them to

a) obtain their height-balanced trees.

b) obtain their size-balanced trees.