

Data Structure Algorithms & Applications

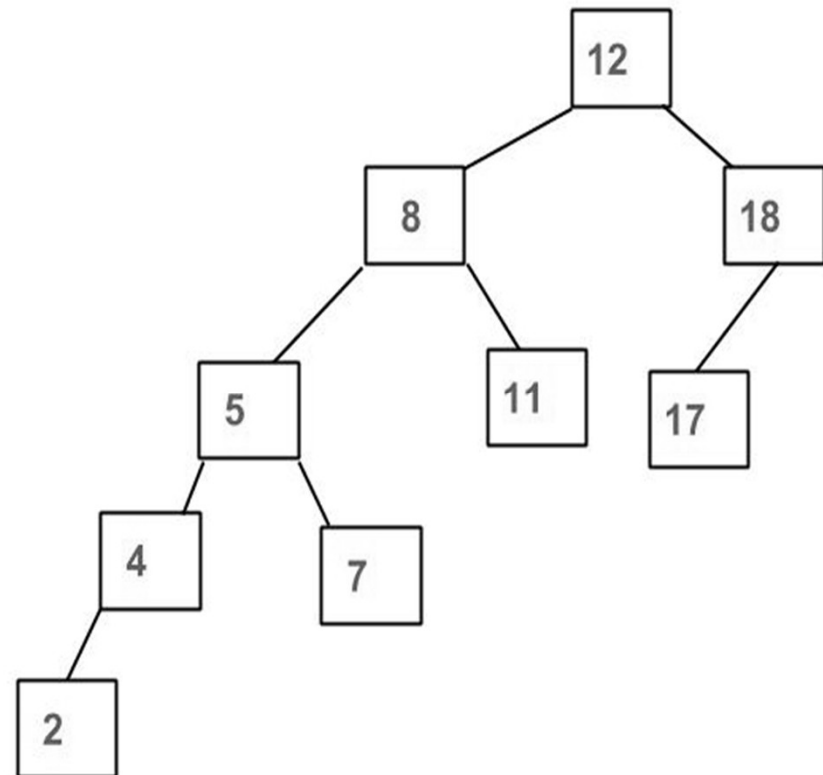
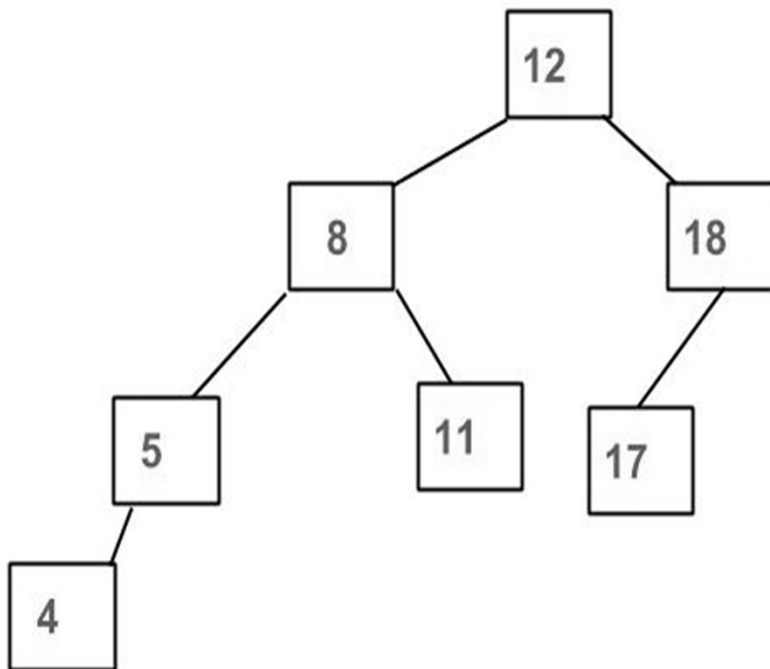
CT-159

Prepared by
Muhammad Kamran

AVL Tree

- AVL tree is a self-balancing Binary Search Tree (**BST**) where the difference between heights of left and right subtrees cannot be more than **one** for all nodes.

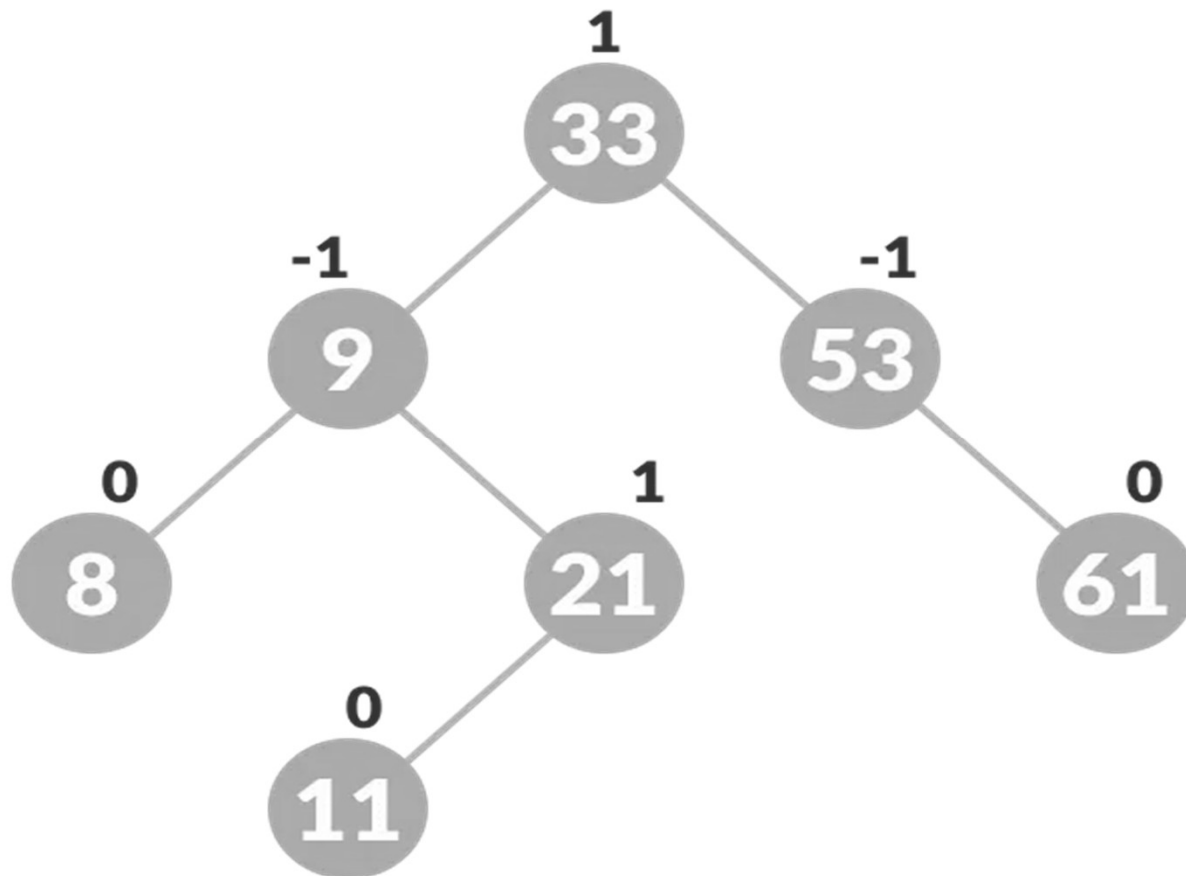
AVL Tree



Balance Factor

- Balance factor of a node in an AVL tree is the difference between the height of the left subtree and that of the right subtree of that node.
- Balance Factor = (Height of Left Subtree - Height of Right Subtree) or (Height of Right Subtree - Height of Left Subtree)
- The self balancing property of an avl tree is maintained by the balance factor. The value of balance factor should always be -1, 0 or +1.

Example

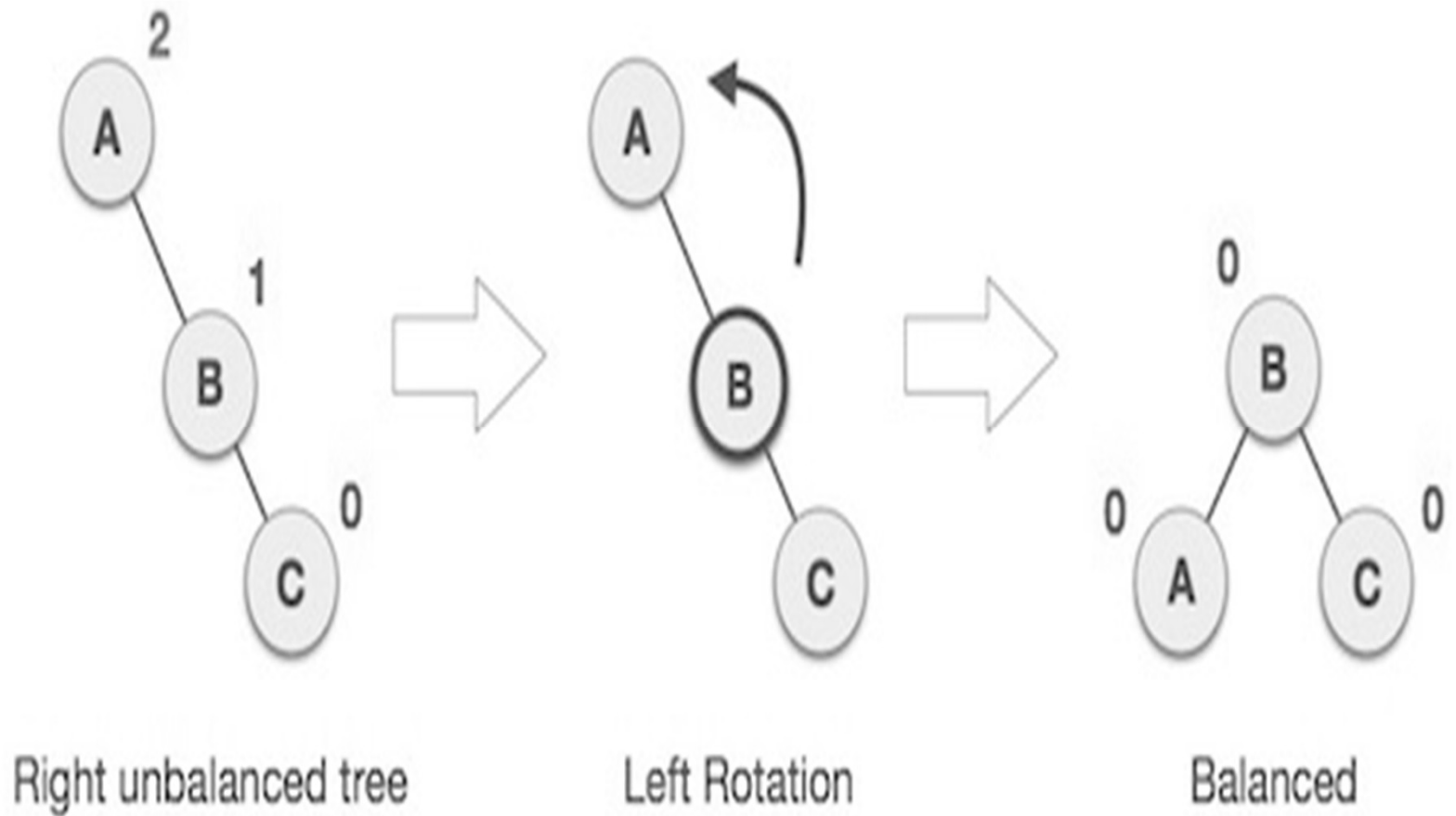


Insertion Operation

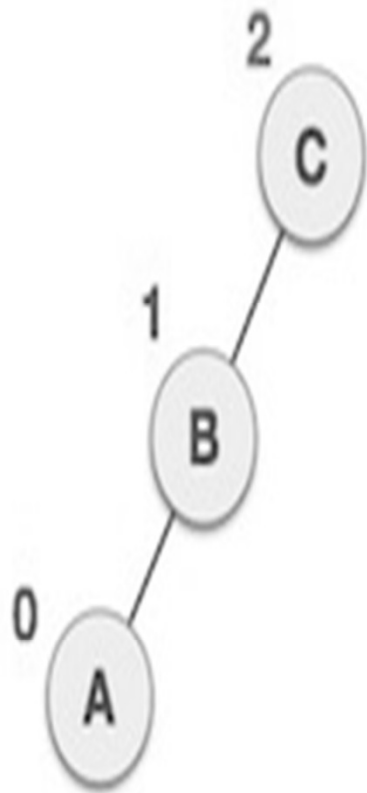
Insertion in an AVL tree is similar to insertion in a binary search tree. But after inserting an element, you need to fix the AVL properties using left or right rotations:

- If there is an imbalance in the left child's right sub-tree, perform a left-right rotation
- If there is an imbalance in the left child's left sub-tree, perform a right rotation
- If there is an imbalance in the right child's right sub-tree, perform a left rotation
- If there is an imbalance in the right child's left sub-tree, perform a right-left rotation

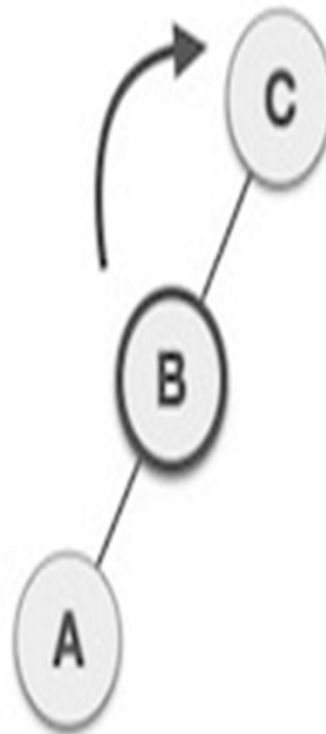
Left Rotation



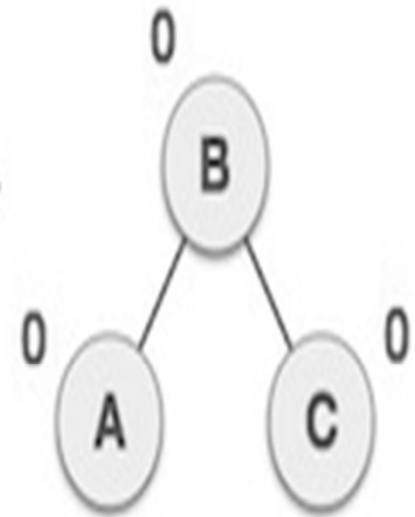
Right Rotation



Left unbalanced Tree

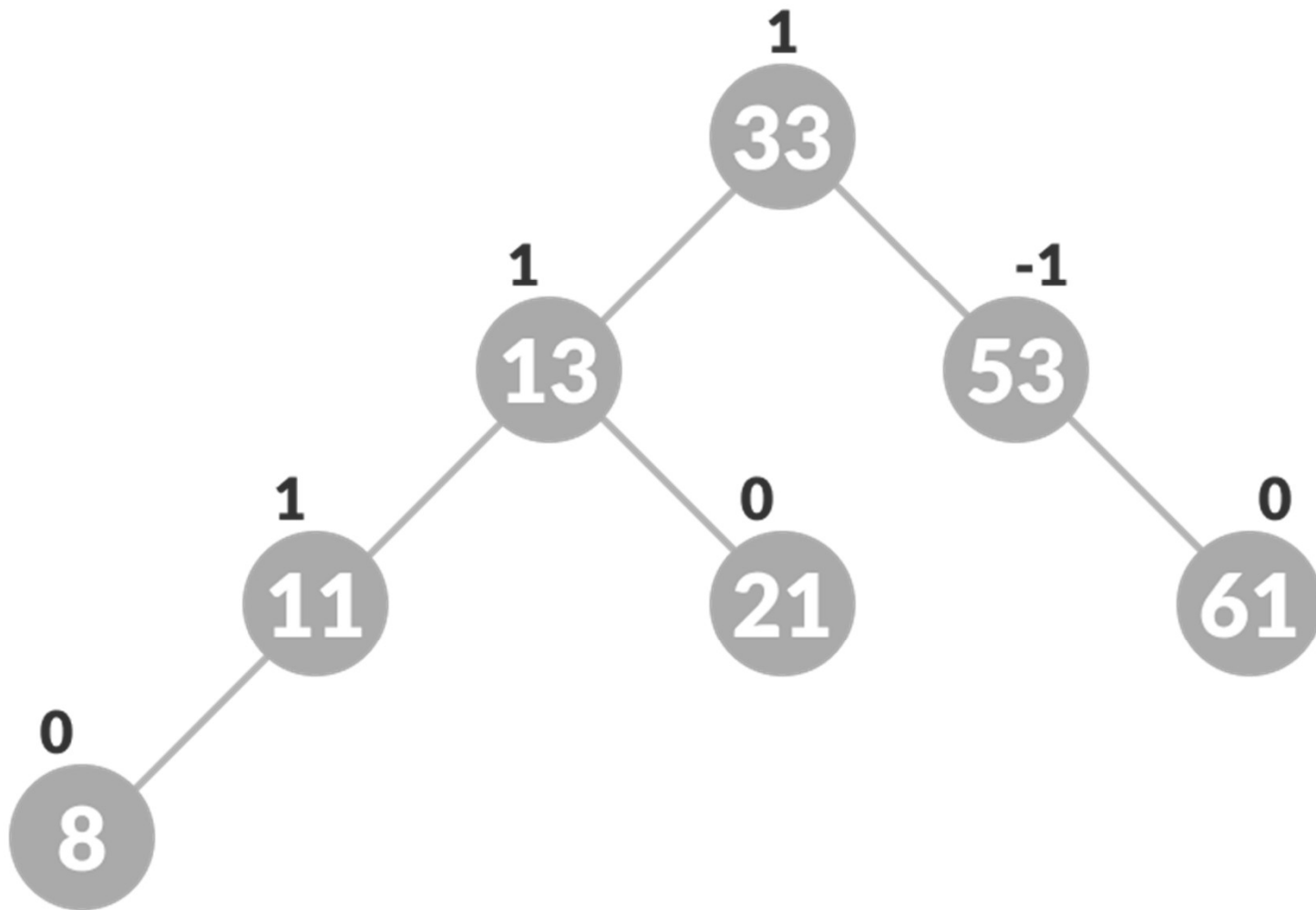


Right Rotation



Balanced Tree

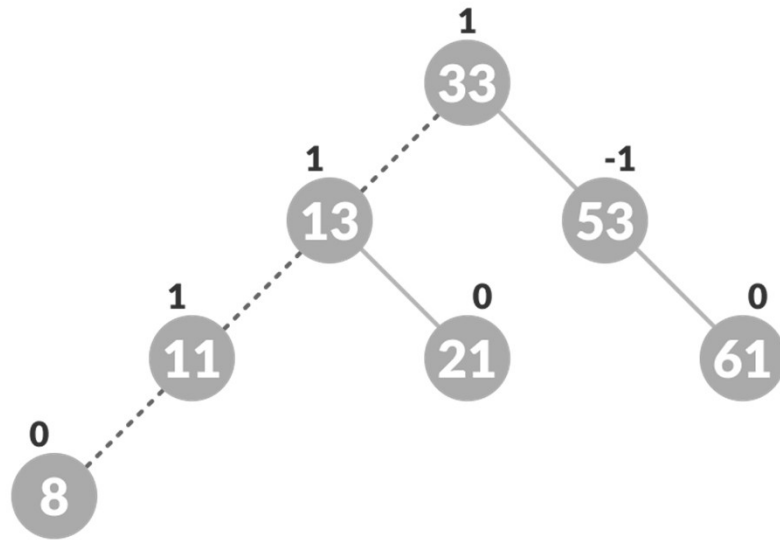
Insertion Operation



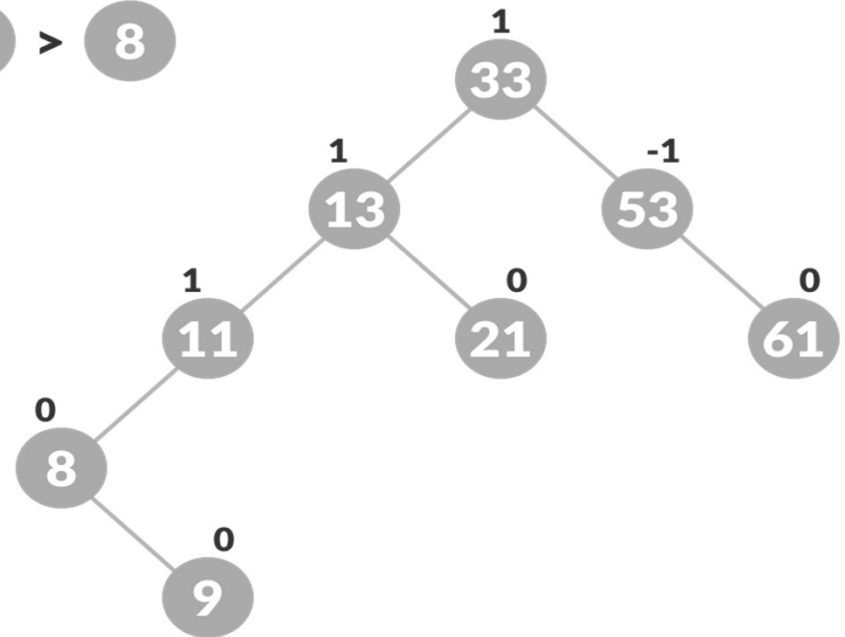
9 < 33

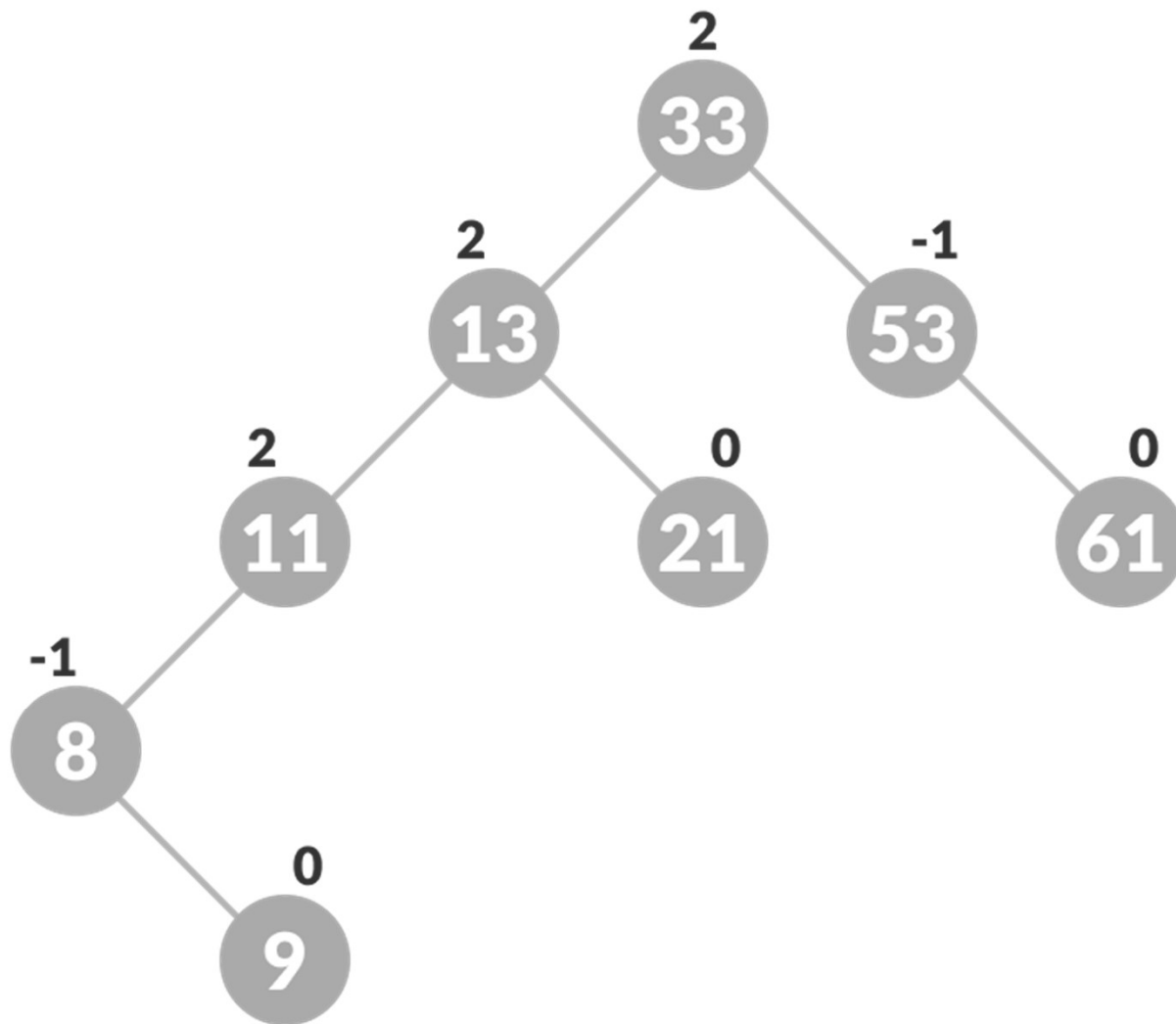
9 < 13

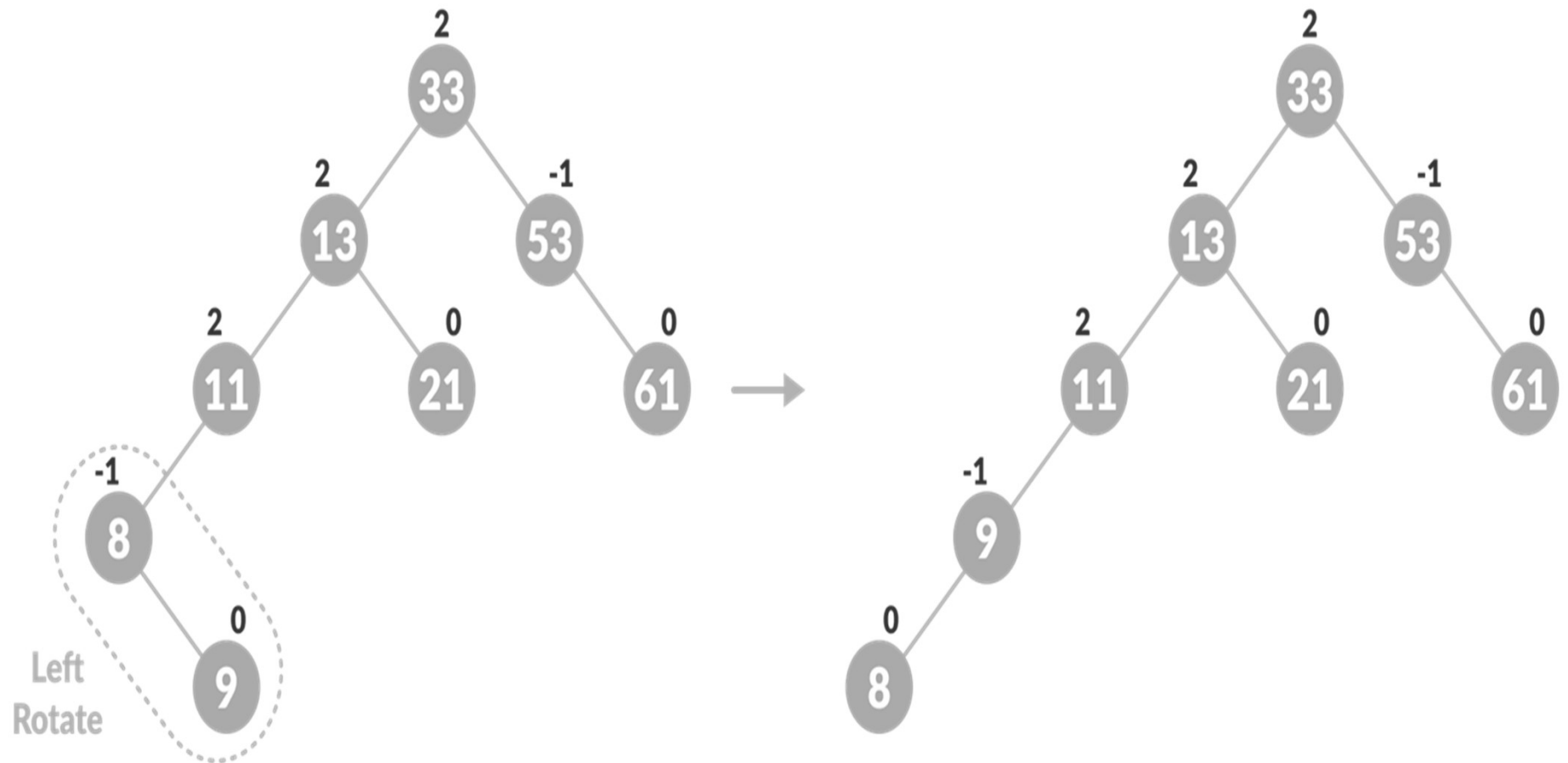
9 < 11

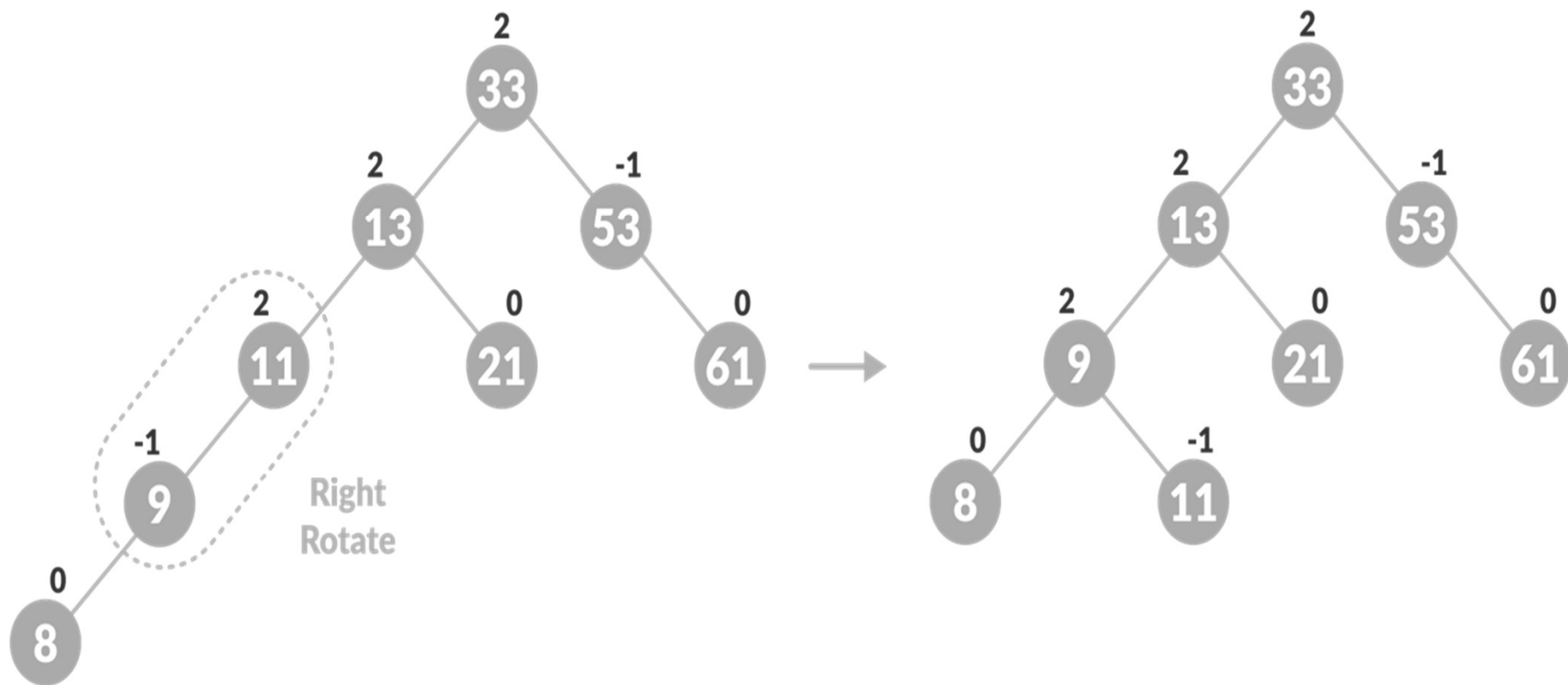


9 > 8

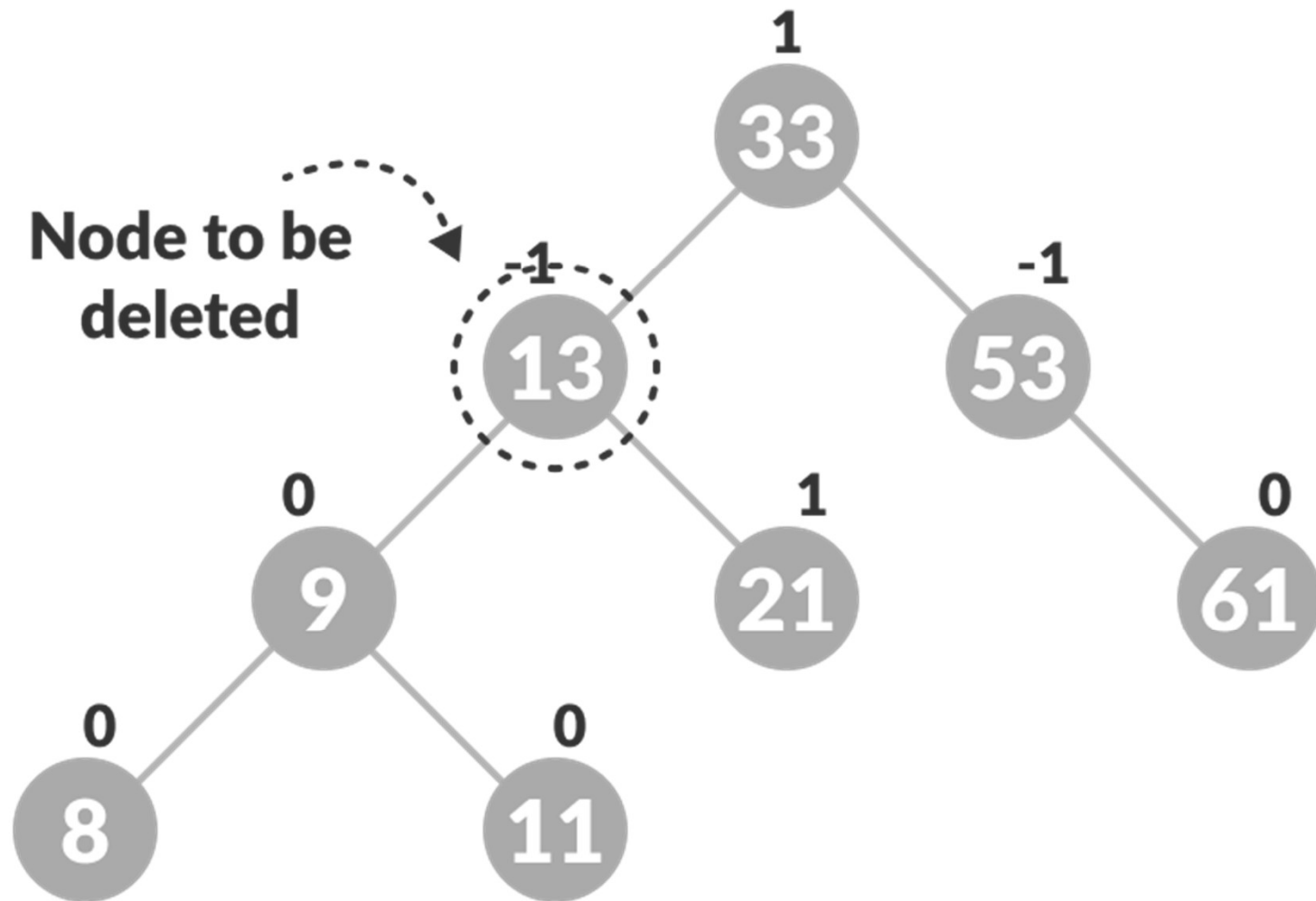








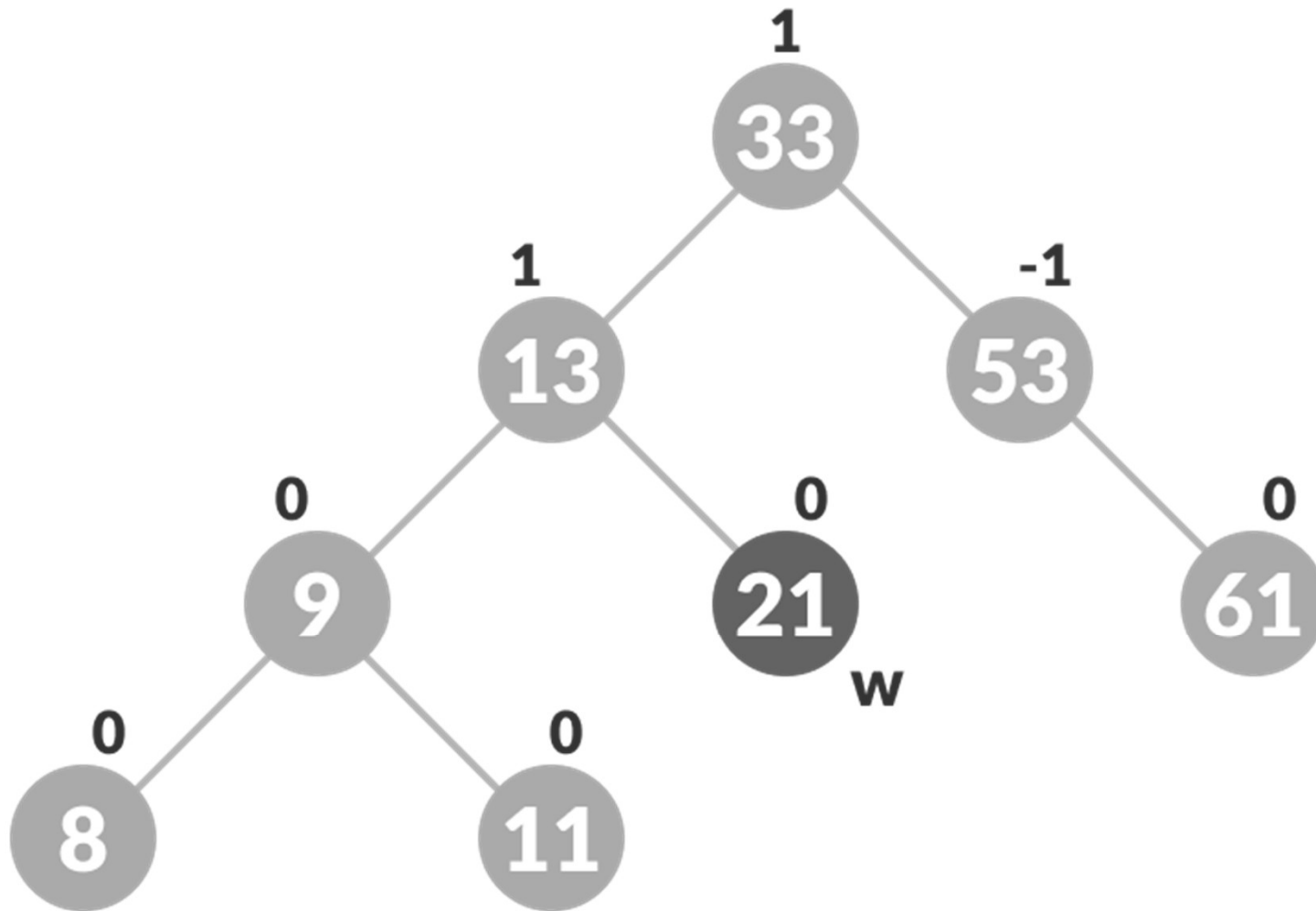
Deletion Operation



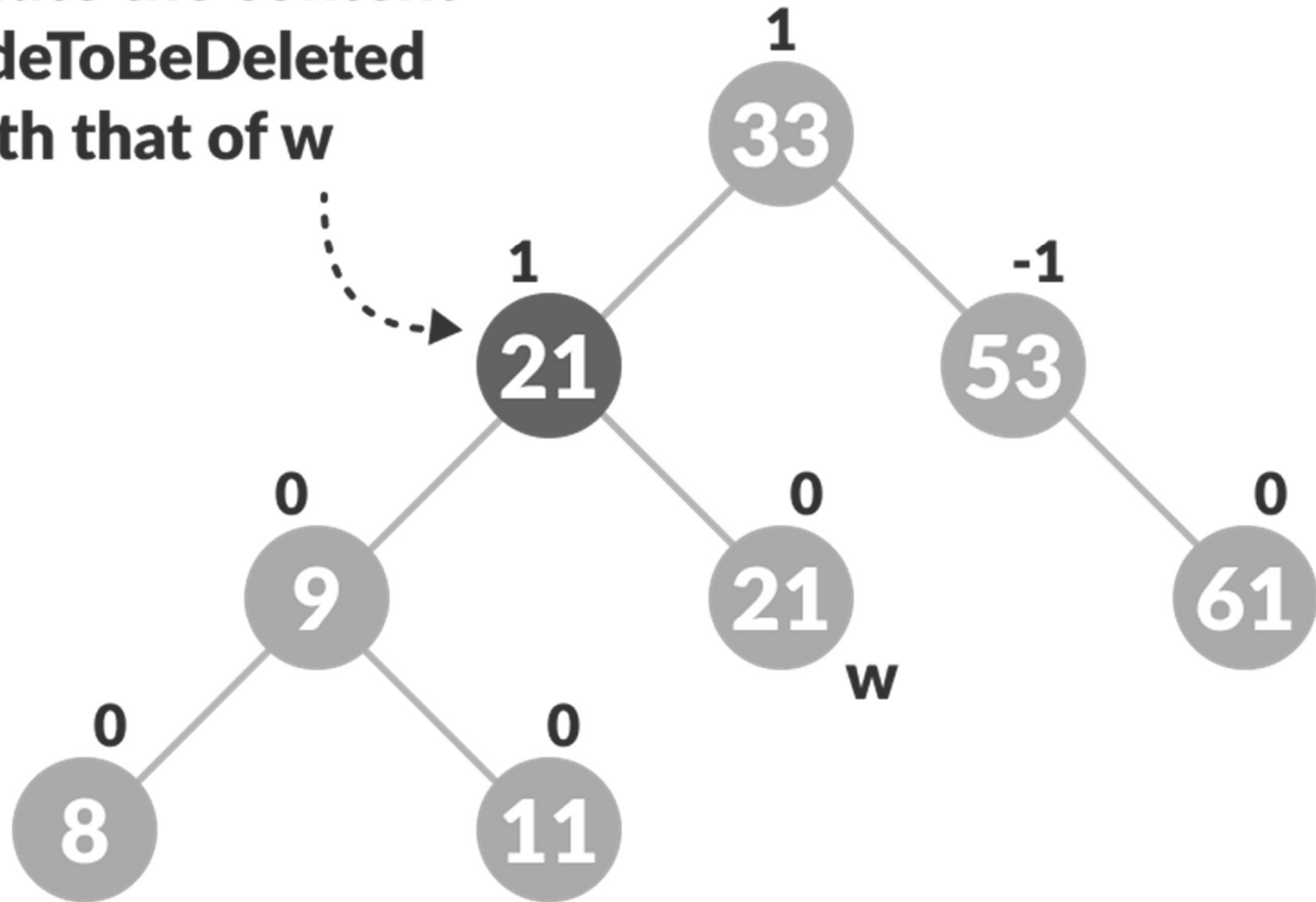
3 Cases of Deletion

- If nodeToBeDeleted is the leaf node (ie. does not have any child), then remove nodeToBeDeleted.
- If nodeToBeDeleted has one child, then substitute the contents of nodeToBeDeleted with that of the child. Remove the child.
- If nodeToBeDeleted has two children, find the inorder successor w of nodeToBeDeleted (ie. node with a minimum value of key in the right subtree).

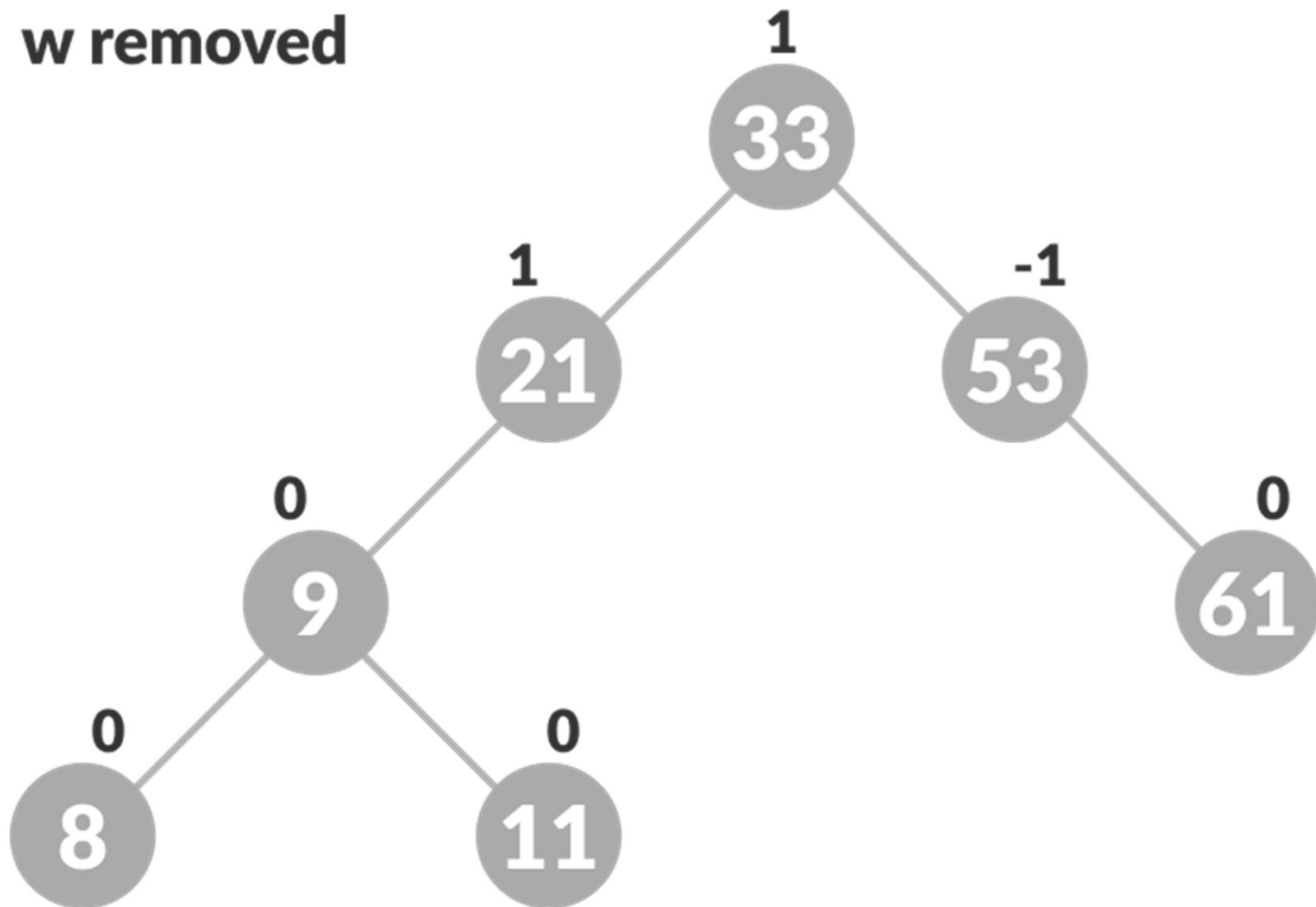
Example



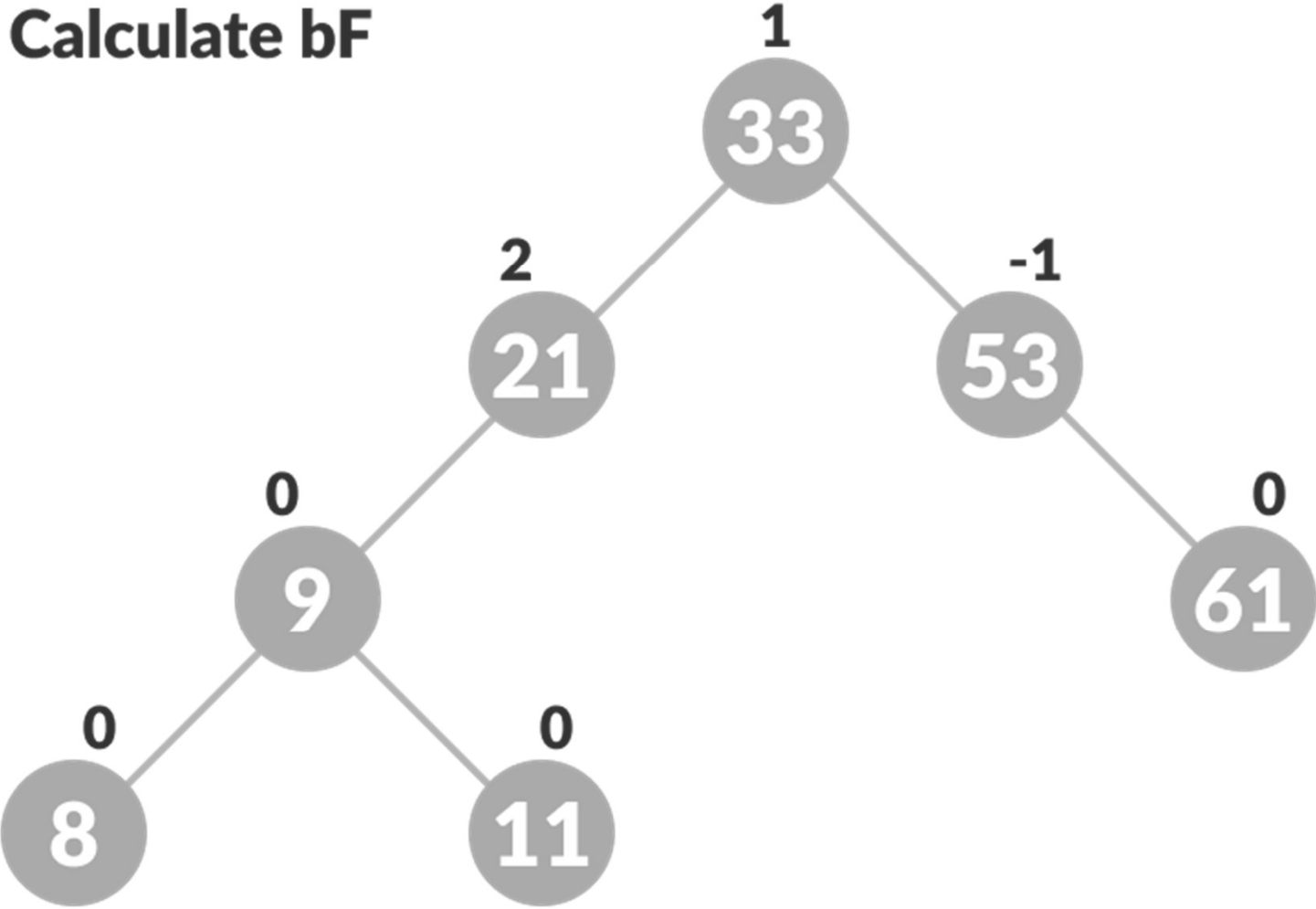
**Substitute the content
of nodeToBeDeleted
with that of w**

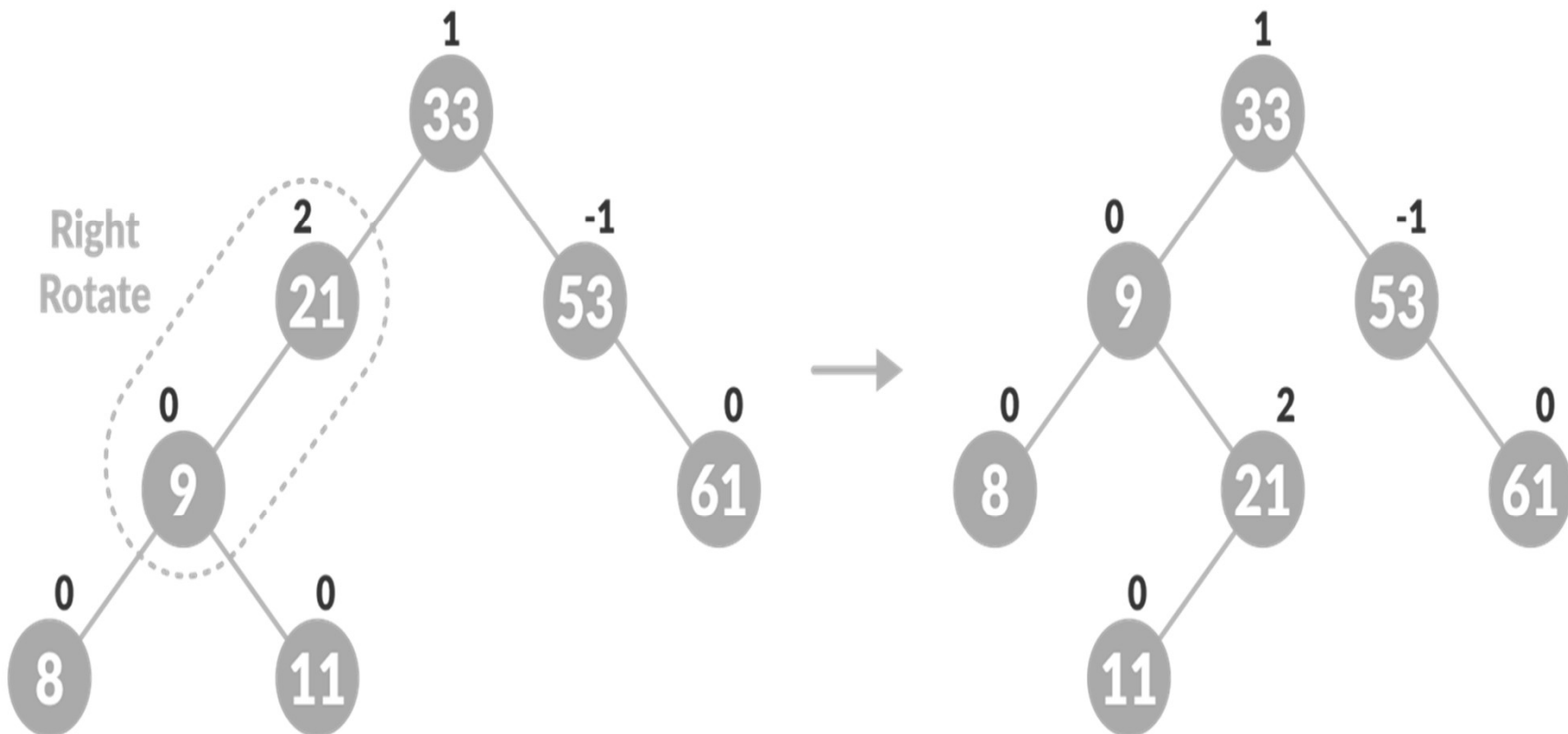


w removed



Calculate bF





Thank You