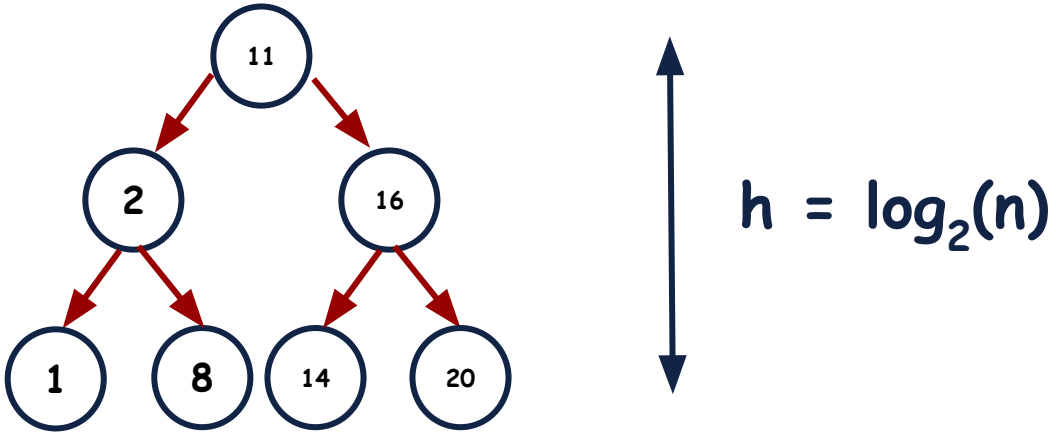
AVL Trees Deletion

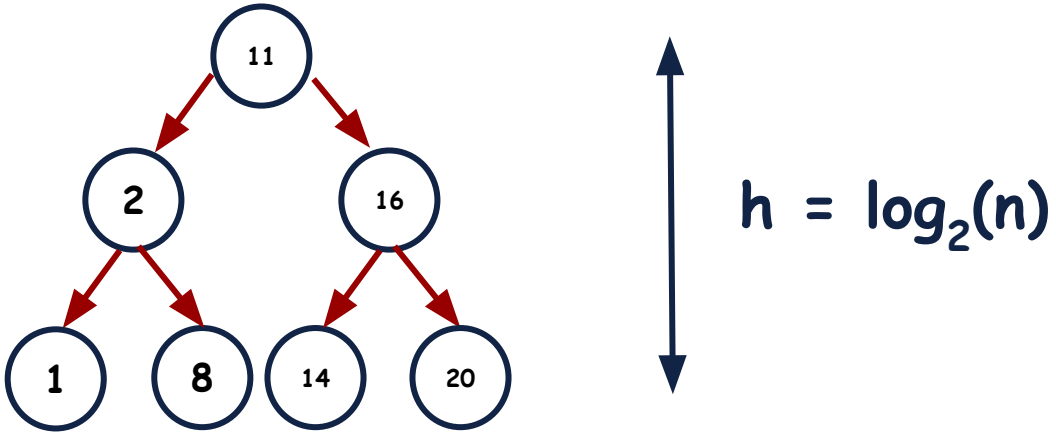
AVL Trees

Such Self Balancing BST in which the heights of the two child subtrees of any node differ by at most one are known as AVL trees (named after inventors Adelson-Velsky and Landis)



AVL Trees

AVL Trees are Self-Balancing Binary Search Trees because they automatically keeps height as small as possible when insertion and deletion operations are performed on the tree



AVL Trees (Insertion)

Previously, we had seen the 4 cases of rotation that are performed for the balancing of the binary search tree.

- Case 1 (LL Case)
- Case 2 (RR Case)
- Case 3 (LR Case)
- Case 4 (RL Case)

AVL Trees (Insertion)

Previously, we had seen the 4 cases of rotation that are performed for the balancing of the binary search trees.

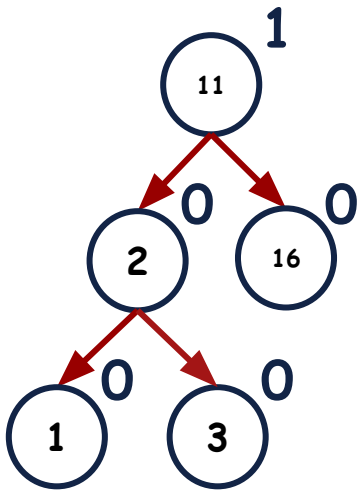
- Case 1 (LL Case)
- Case 2 (RR Case)
- Case 3 (LR Case)
- Case 4 (RL Case)

AVL Trees (Deletion)

Now, let's see how we have to balance the AVL tree when a node is deleted from the AVL Tree and the balance factor becomes greater than 1 or less than -1.

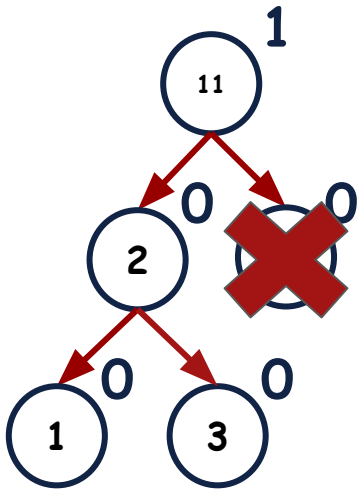
AVL Trees (Deletion)

Suppose we have the following AVL Tree and we want to delete 16 from it.



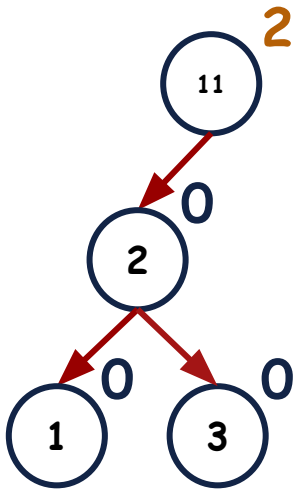
AVL Trees (Deletion)

Suppose we have the following AVL Tree and we want to delete 16 from it.



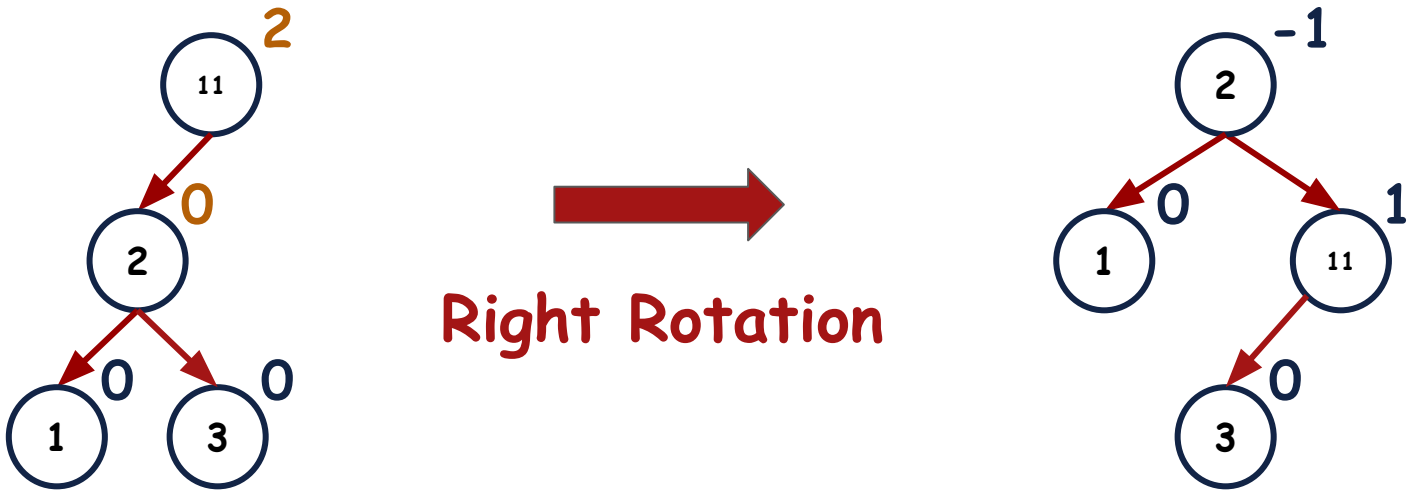
AVL Trees (Deletion)

If the balance factor of the parent node becomes greater than 1 then check the balance factor of parent's left child.



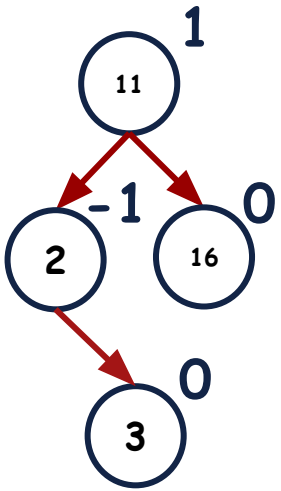
AVL Trees (Deletion)

If the balance factor of parent's left child is greater than or equal to 0 then perform Right rotation.



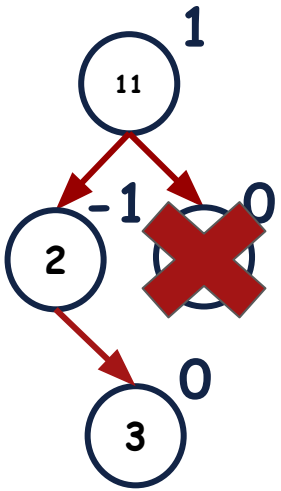
AVL Trees (Deletion)

Suppose we have the following AVL Tree and we want to delete 16 from it.



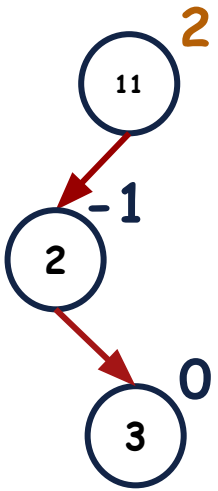
AVL Trees (Deletion)

Suppose we have the following AVL Tree and we want to delete 16 from it.



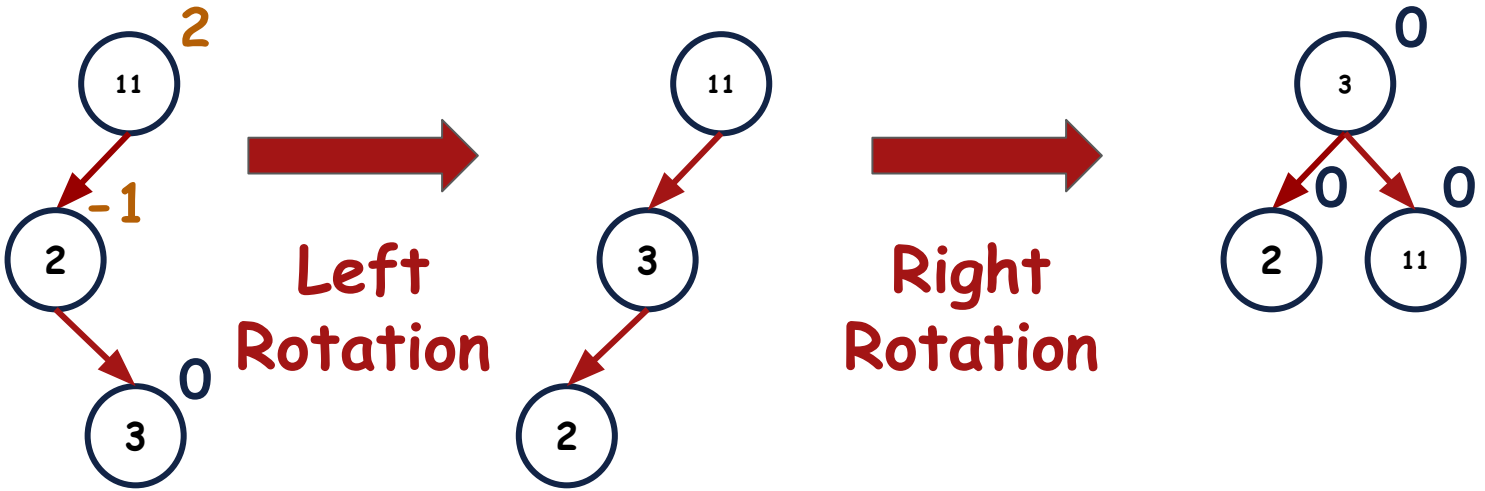
AVL Trees (Deletion)

If the balance factor of the parent node becomes greater than 1 then check the balance factor of parent's left child.



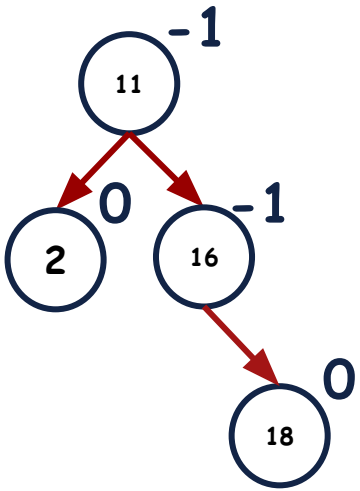
AVL Trees (Deletion)

If the balance factor of parent's left child is less than 0 then perform Left and then Right rotation.



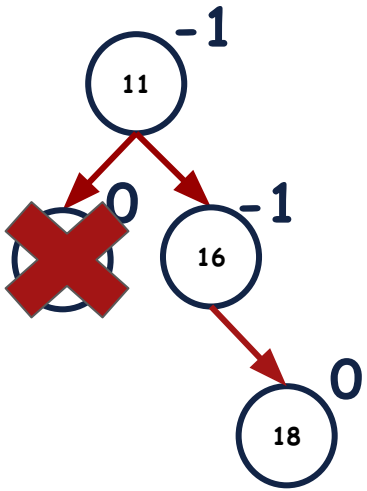
AVL Trees (Deletion)

Suppose we have the following AVL Tree and we want to delete 2 from it.



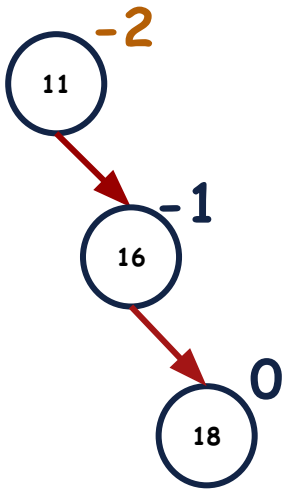
AVL Trees (Deletion)

Suppose we have the following AVL Tree and we want to delete 2 from it.



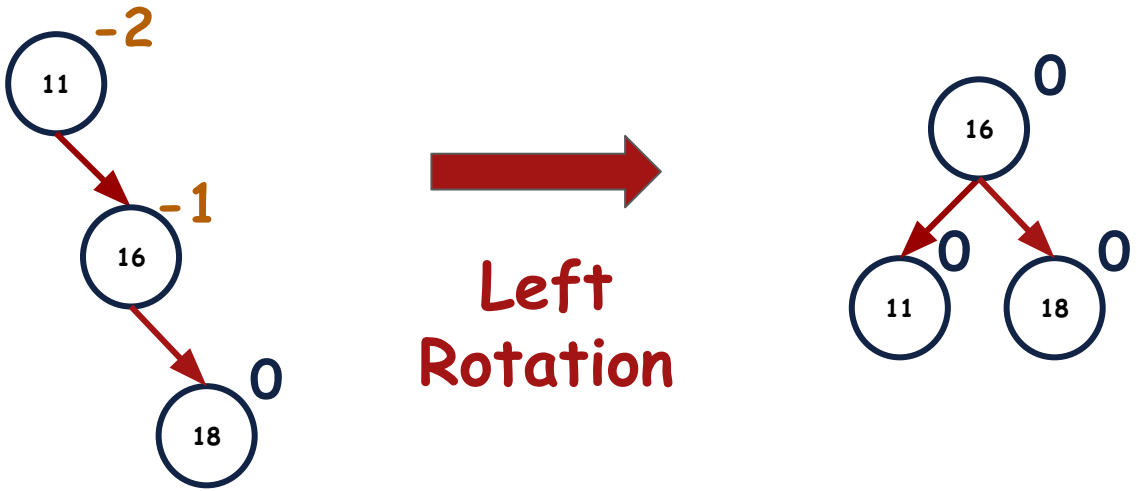
AVL Trees (Deletion)

If the balance factor of the parent node becomes less than -1 then check the balance factor of parent's right child.



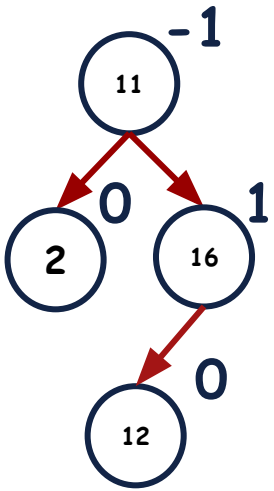
AVL Trees (Deletion)

If the balance factor of parent's right child is less than or equal to 0 then perform Left rotation.



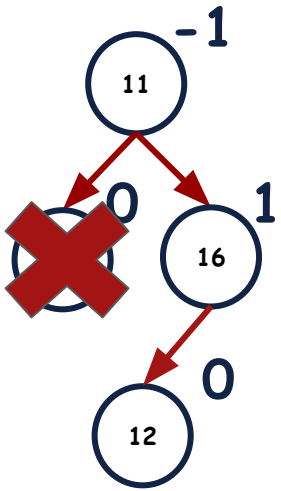
AVL Trees (Deletion)

Suppose we have the following AVL Tree and we want to delete 2 from it.



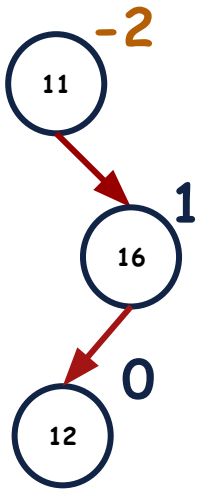
AVL Trees (Deletion)

Suppose we have the following AVL Tree and we want to delete 2 from it.



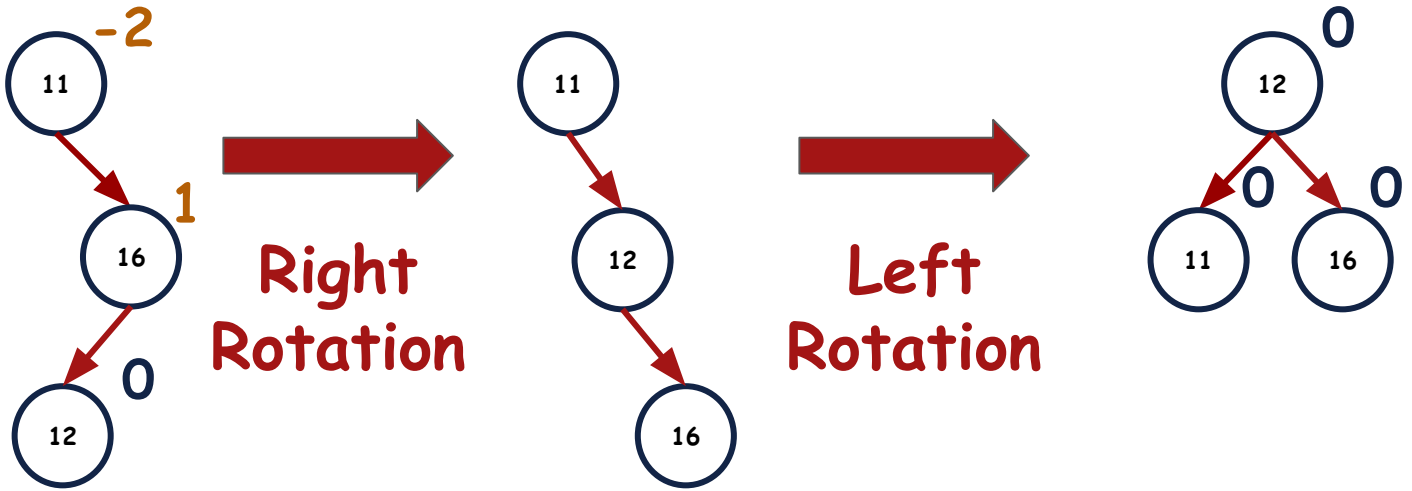
AVL Trees (Deletion)

If the balance factor of the parent node becomes less than -1 then check the balance factor of parent's right child.



AVL Trees (Deletion)

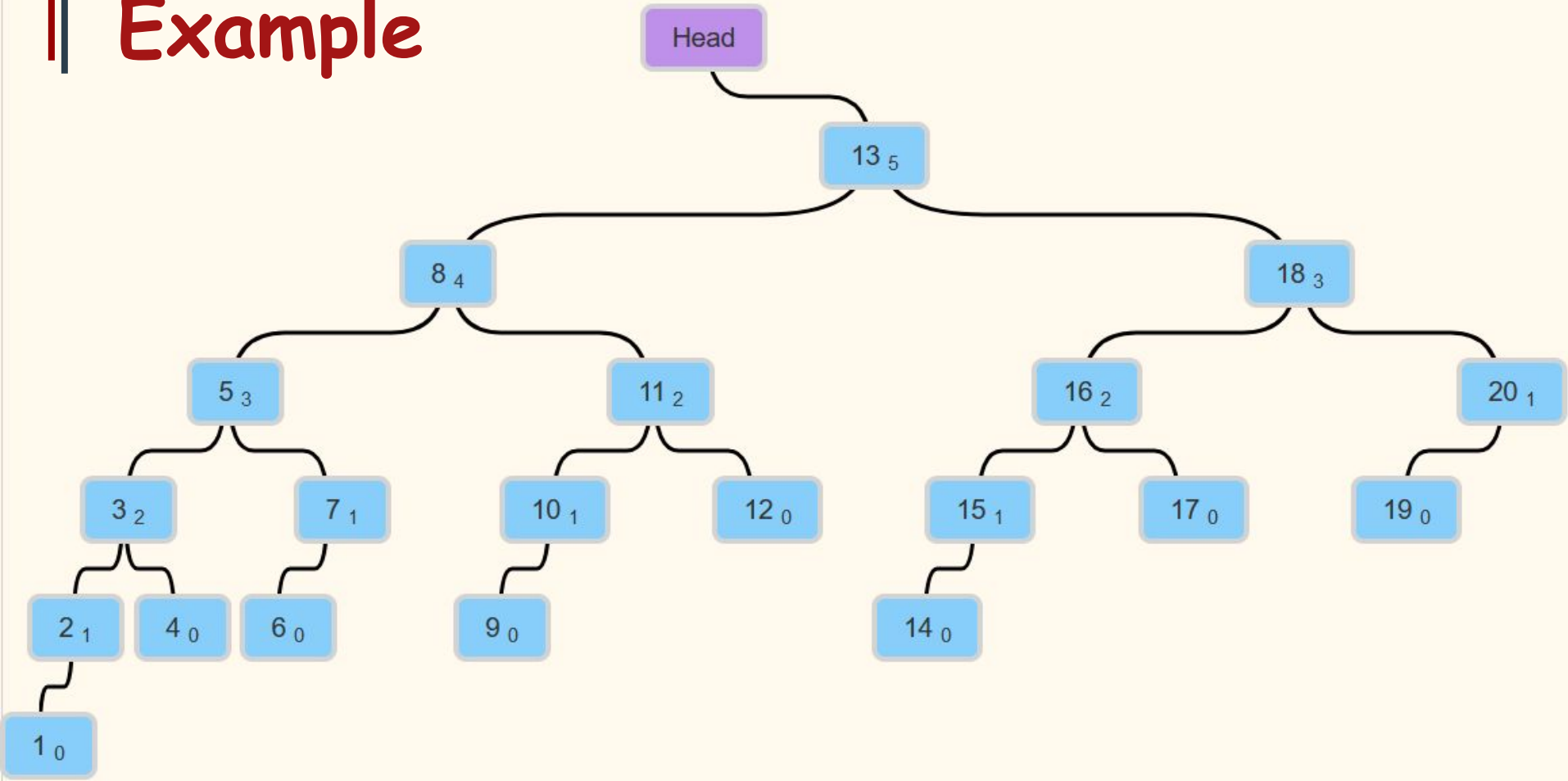
If the balance factor of parent's right child is greater than 0 then perform Right then Left Rotation.



AVL Trees (Deletion): 4 Cases

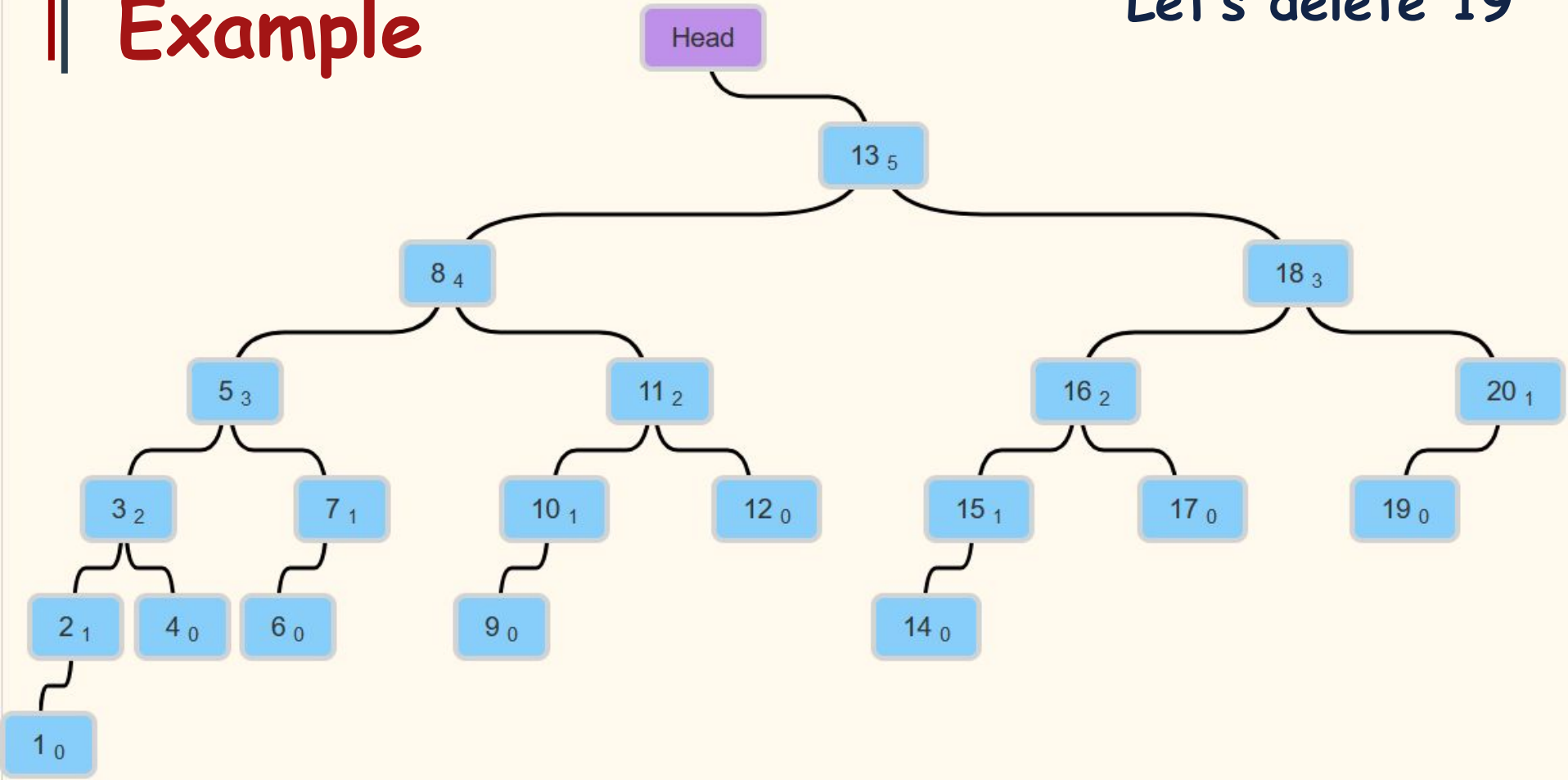
1. If BF(parent) > 1
   a. If (BF(parent->left) >= 0)
      (Right Rotation)
   b. If (BF(parent->left) < 0)
      (Left Rotation then Right Rotation)
2. If BF(parent) < -1
   a. If (BF(parent->right) <= 0)
      (Left Rotation)
   b. If (BF(parent->right) > 0)
      (Right Rotation then Left Rotation)

Example

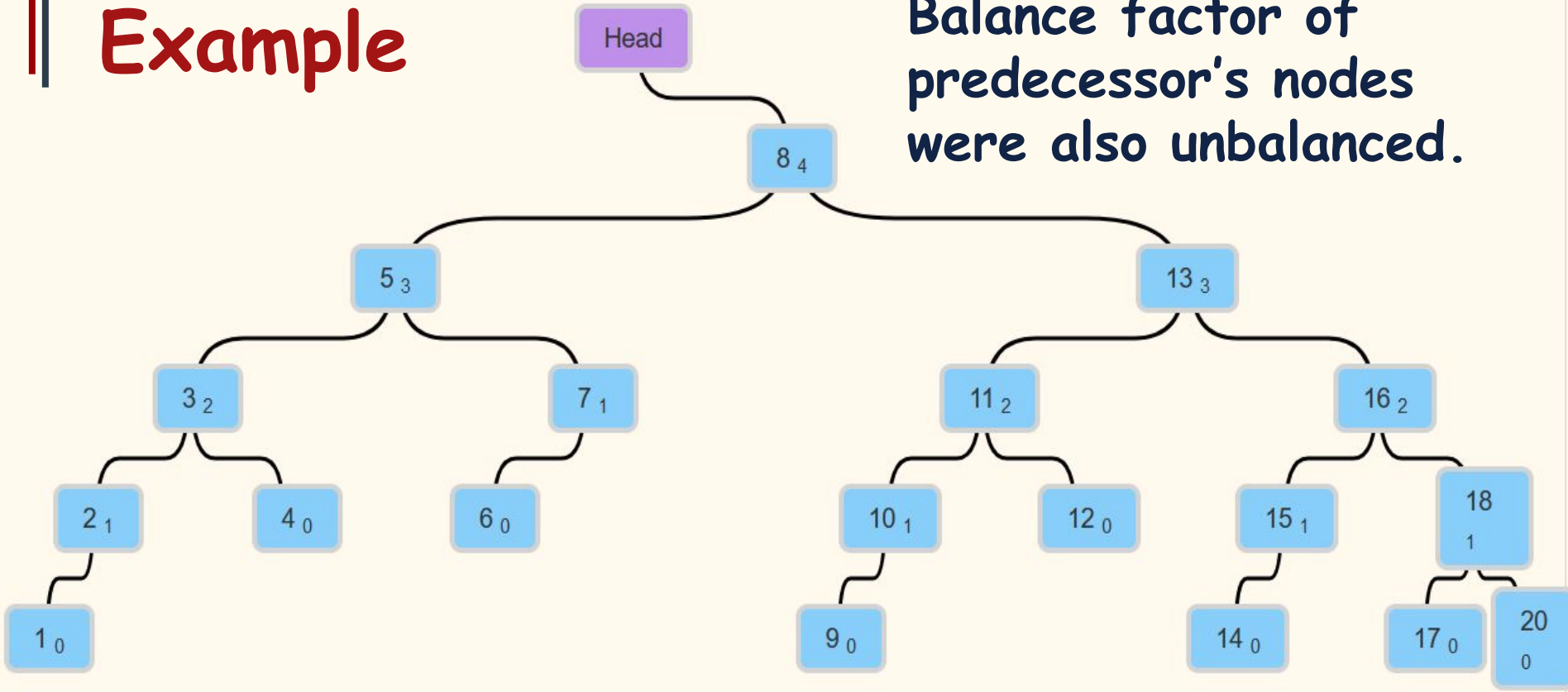

Head
13 5
8 4
18 3
5 3
11 2
16 2
20 1
3 2
7 1
10 1
12 0
15 1
17 0
19 0
2 1
4 0
6 0
9 0
14 0
1 0

Example

Let's delete 19

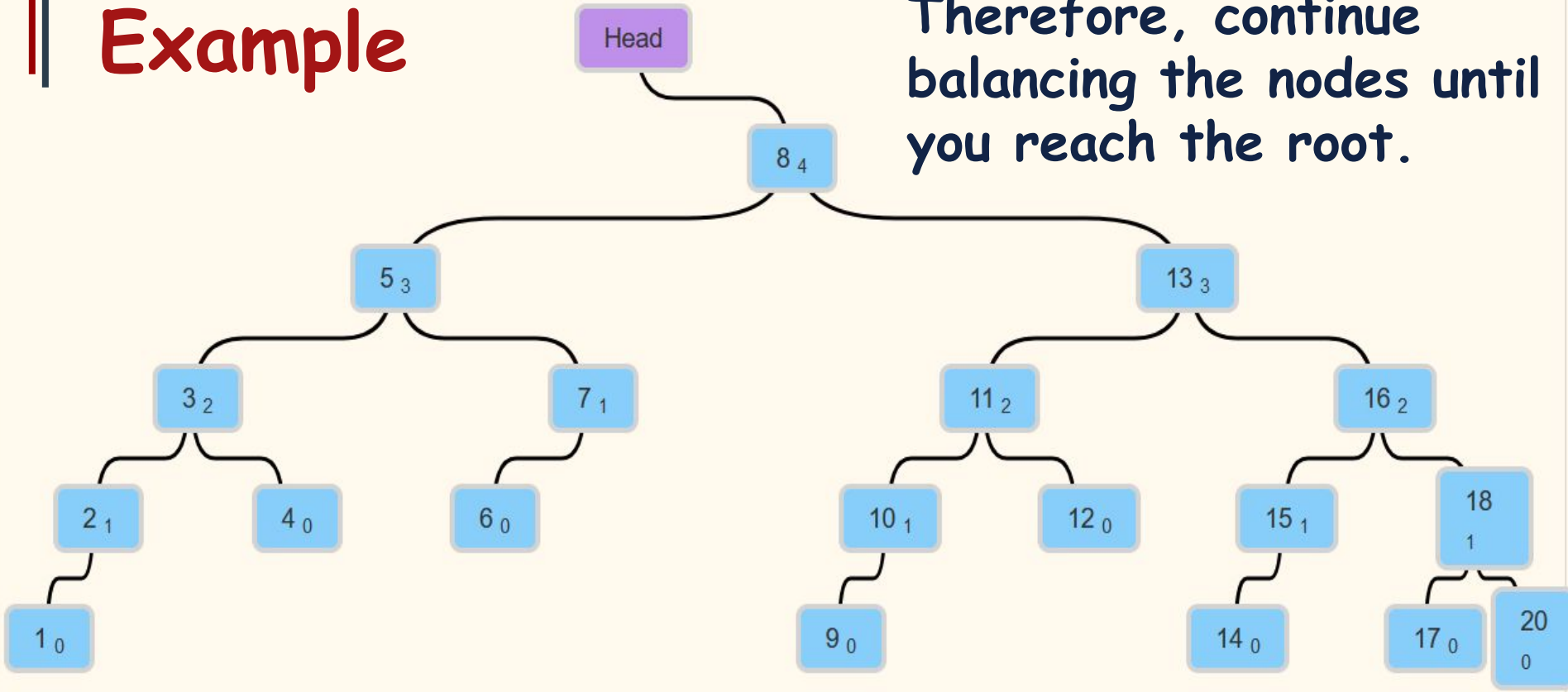


Example



Balance factor of predecessor's nodes were also unbalanced.

Example



Therefore, continue balancing the nodes until you reach the root.

AVL Trees (Deletion): 4 Cases

While stack is not empty

1. If BF(parent) > 1
   a. If (BF(parent->left) >= 0)
      (Right Rotation)
   b. If (BF(parent->left) < 0)
      (Left Rotation then Right Rotation)
2. If BF(parent) < -1
   a. If (BF(parent->right) <= 0)
      (Left Rotation)
   b. If (BF(parent->right) > 0)
      (Right Rotation then Left Rotation)

AVL Trees: Working Example

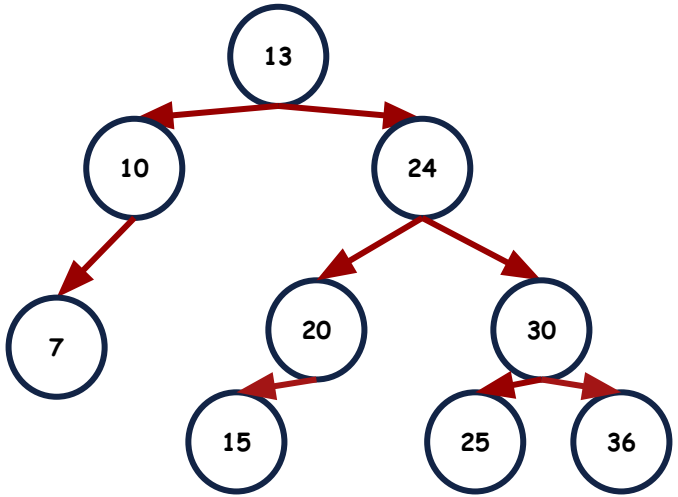
Build an AVL tree with the following values.

Input: 15, 20, 24, 10, 13, 7, 30, 36, 25

AVL Trees: Working Example

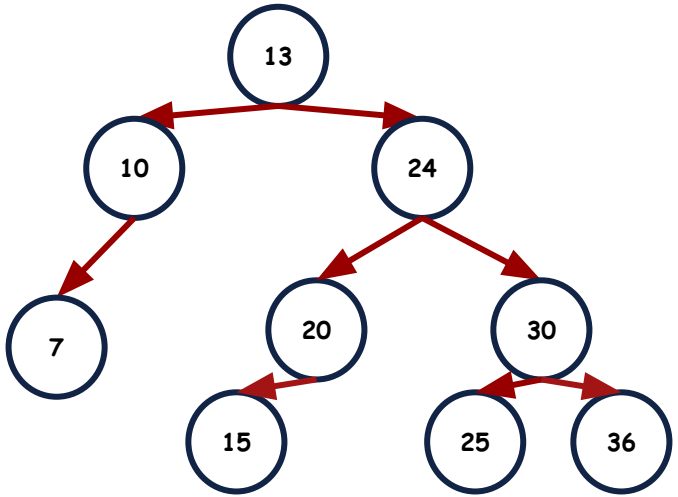
Build an AVL tree with the following values.

Input: 15, 20, 24, 10, 13, 7, 30, 36, 25



AVL Trees: Working Example

Now, delete the nodes in the following sequence.
7, 24, 25, 30, 13



Learning Objective

Students should be able to delete elements in the AVL trees.

Self Assessment

Write the Function in the AVLTree Class to delete the given value from the AVL tree while maintaining height balance of the AVL tree.

Links

Visualize the AVL Tree
https://www.cs.usfca.edu/~galles/visualization/AVLtree.html