AVL TREES

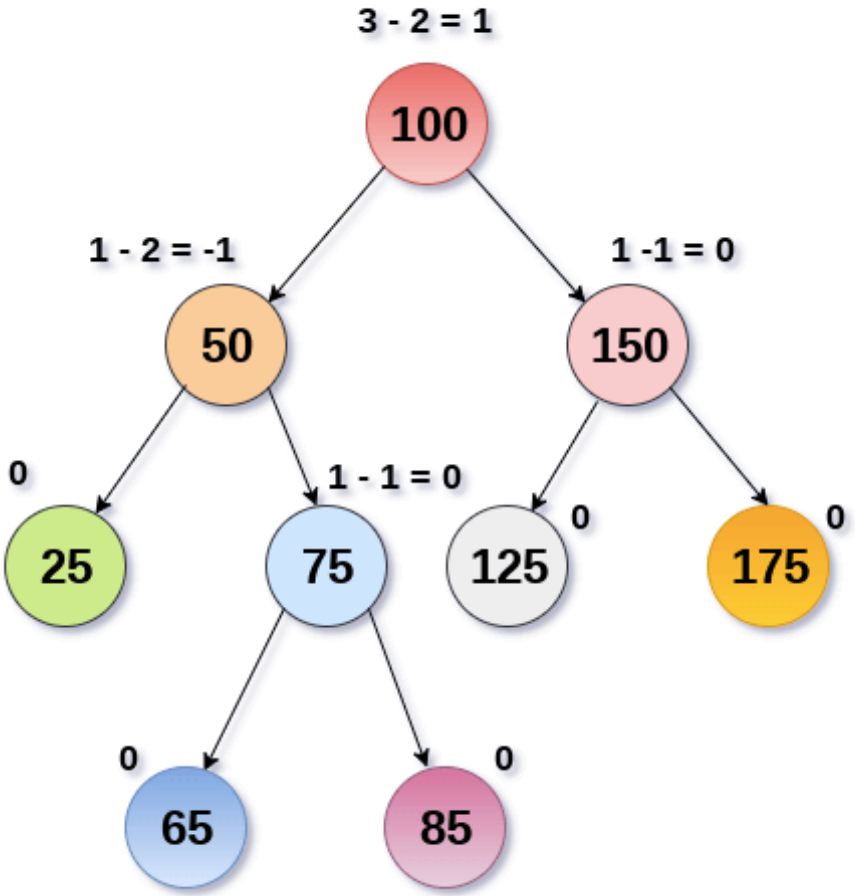
Balance Factor (k) = height (left(k)) - height (right(k))

If balance factor of any node is 1, it means that the left sub-tree is one level higher than the right sub-tree.

If balance factor of any node is 0, it means that the left sub-tree and right sub-tree contain equal height.

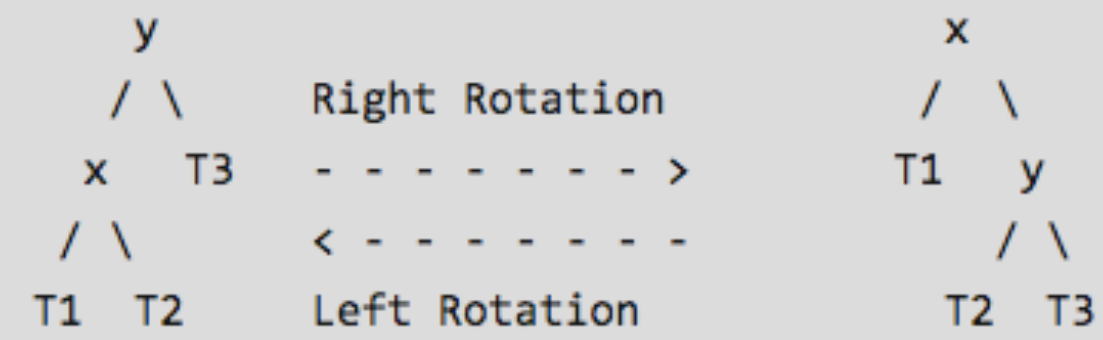
If balance factor of any node is -1, it means that the left sub-tree is one level lower than the right sub-tree.

An AVL tree is given in the following figure. We can see that, balance factor associated with each node is in between -1 and +1. therefore, it is an example of AVL tree.



Rotation

T1, T2 and T3 are subtrees of the tree rooted with y (on the left side) or x (on the right side)



Keys in both of the above trees follow the following order

keys(T1) < key(x) < keys(T2) < key(y) < keys(T3)

So BST property is not violated anywhere.

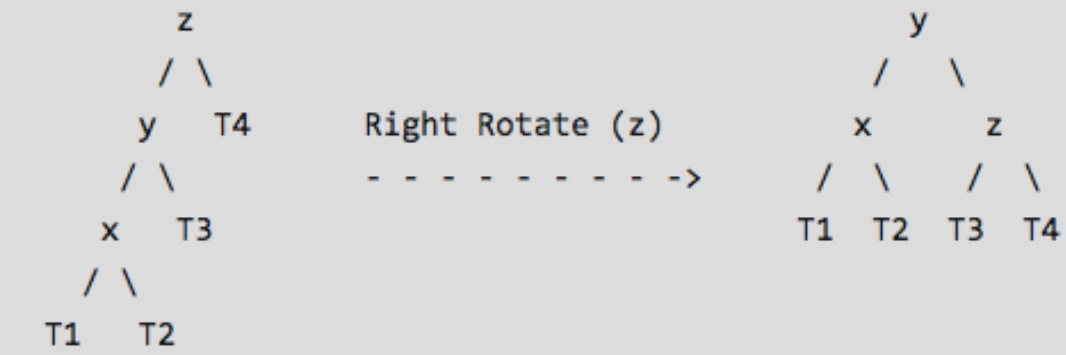
Balance your Binary Search Tree

Left Left Case

y is left child of z and x is left child of y

a) Left Left Case

T1, T2, T3 and T4 are subtrees.

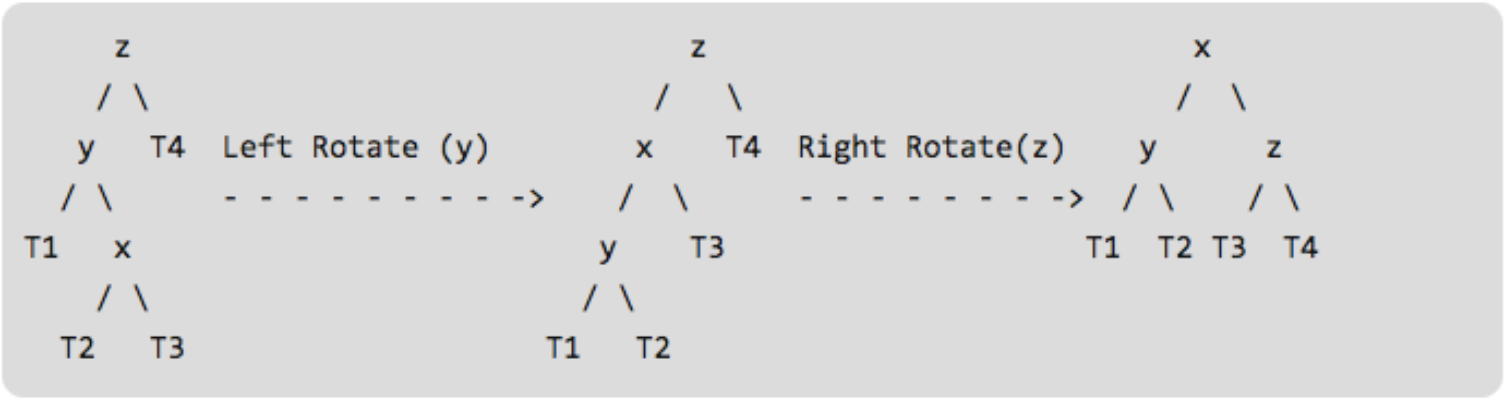


Balance your Binary Search Tree

Left Right Case

y is left child of z and x is right child of y

b) Left Right Case

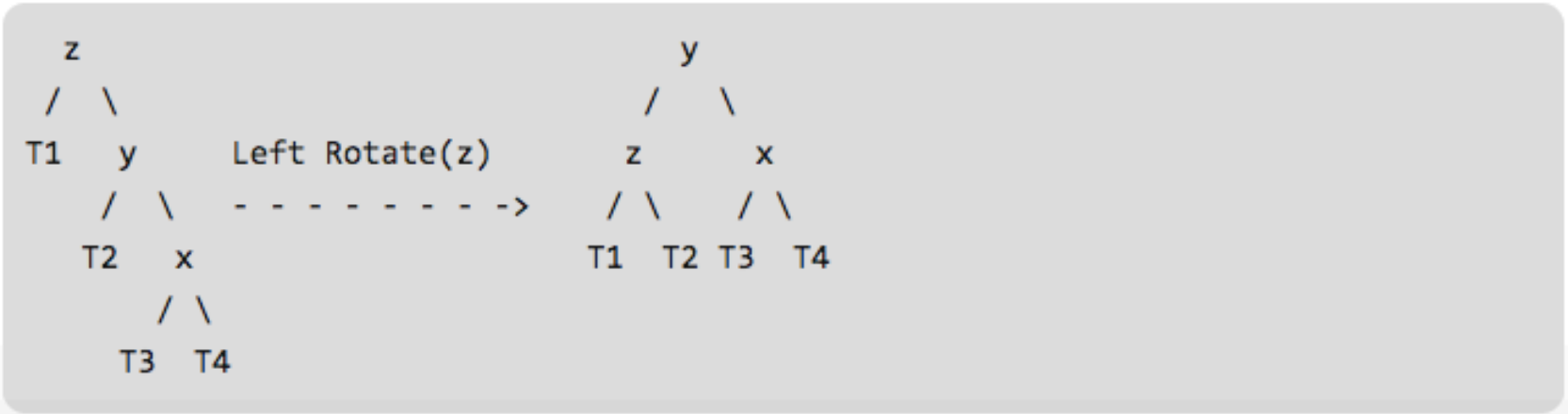


Balance your Binary Search Tree

Right Right Case

y is right child of z and x is right child of y

c) Right Right Case

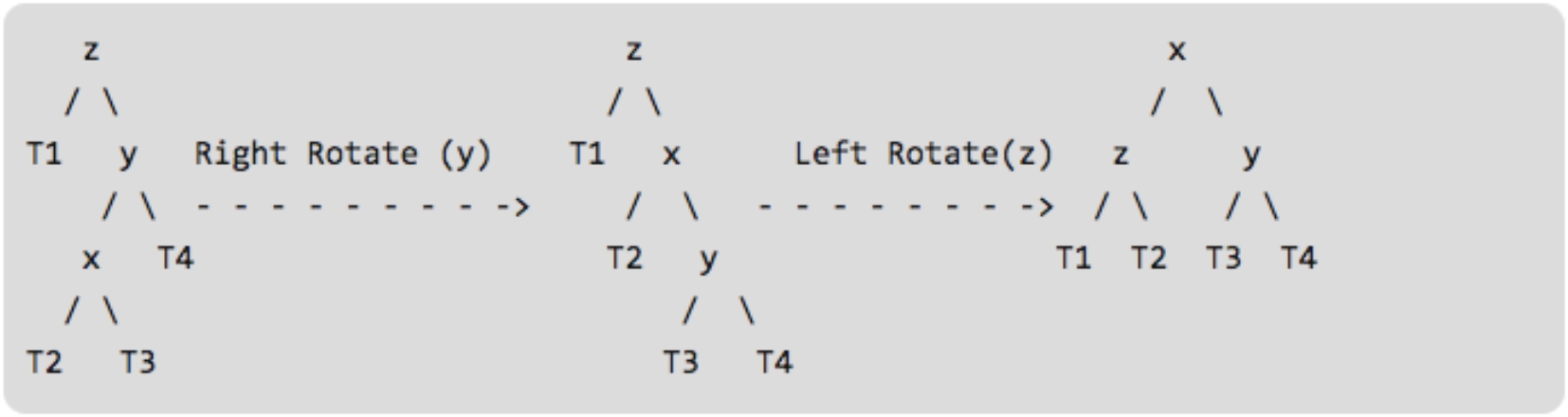


Balance your Binary Search Tree

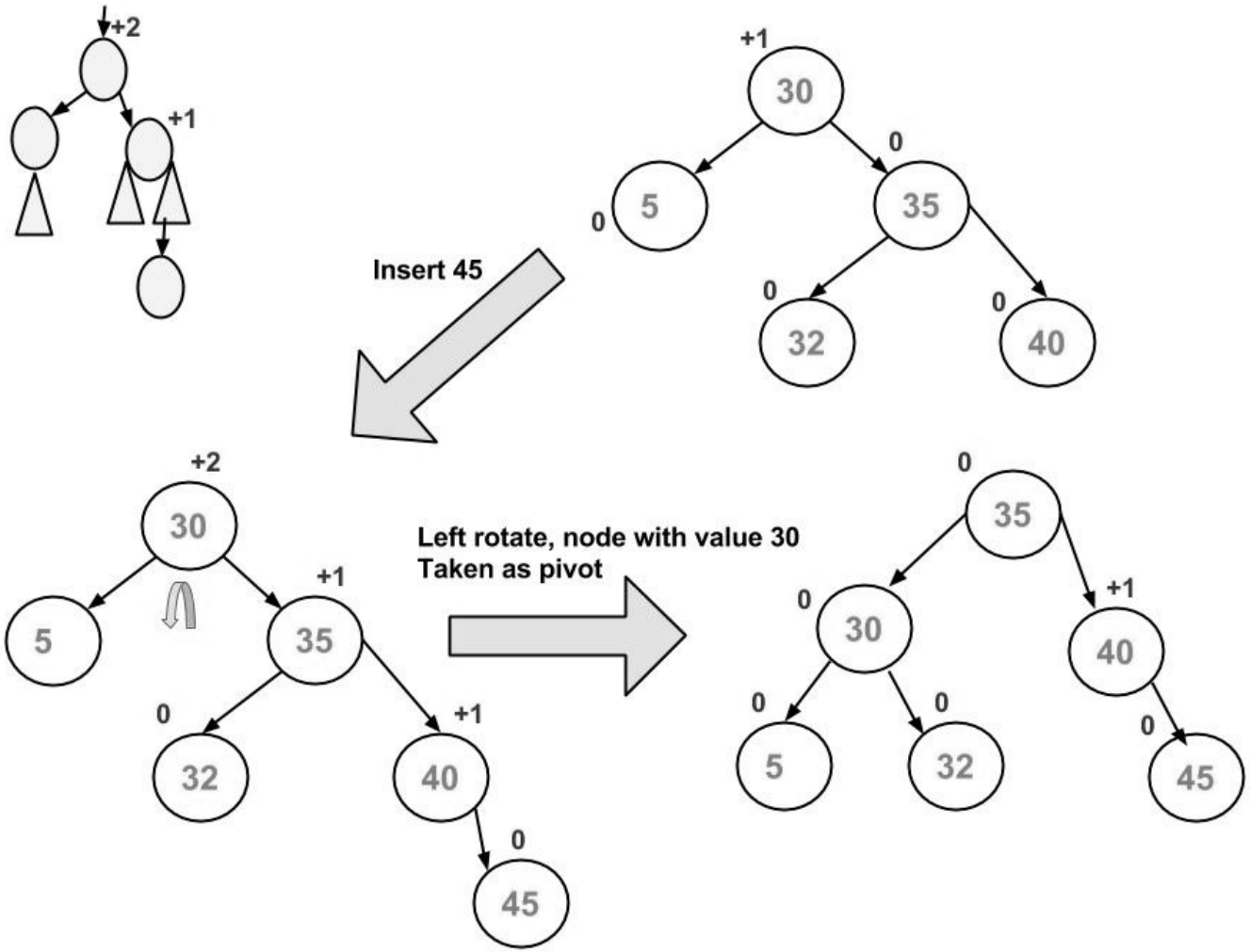
Right Left Case

y is right child of z and x is left child of y

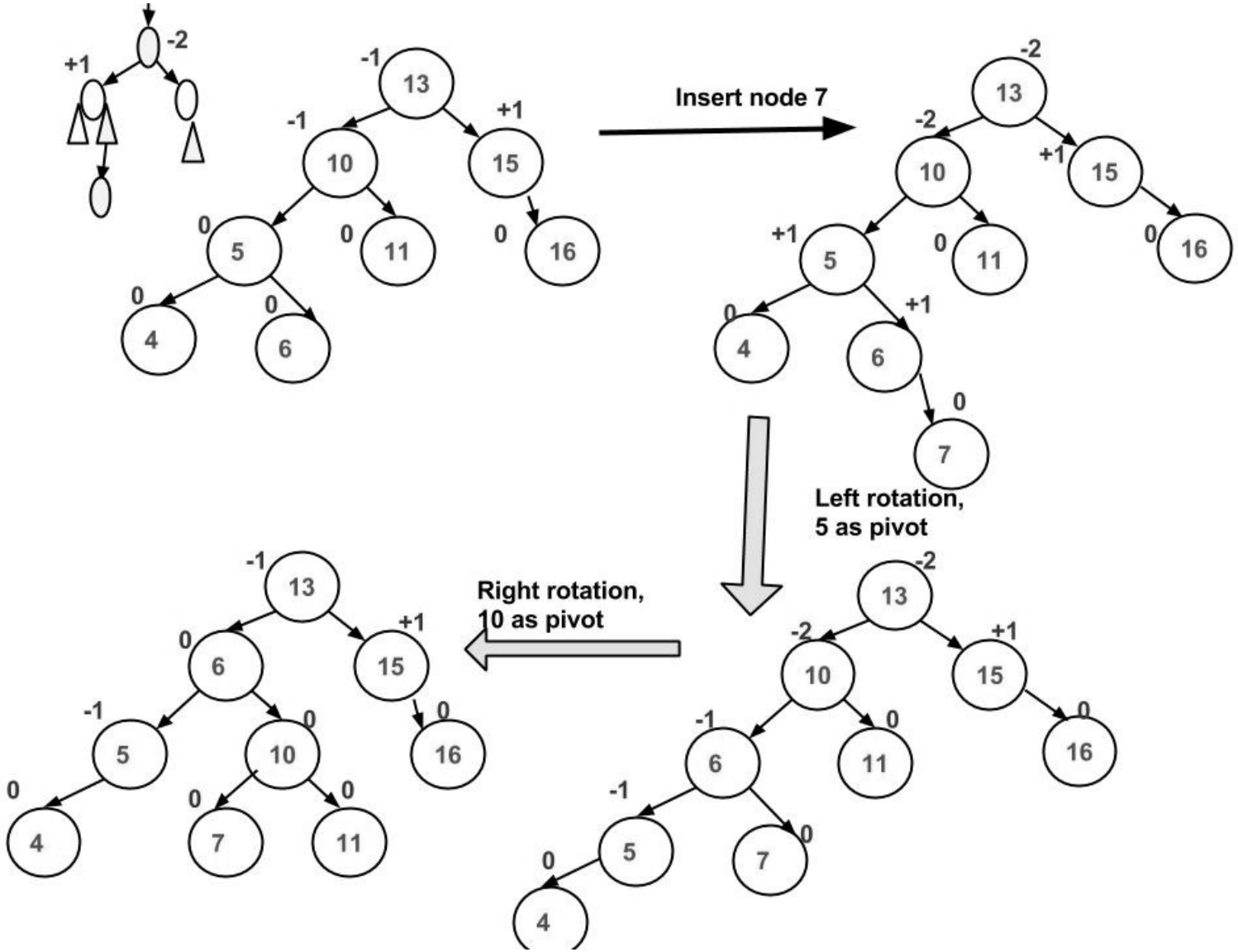
d) Right Left Case



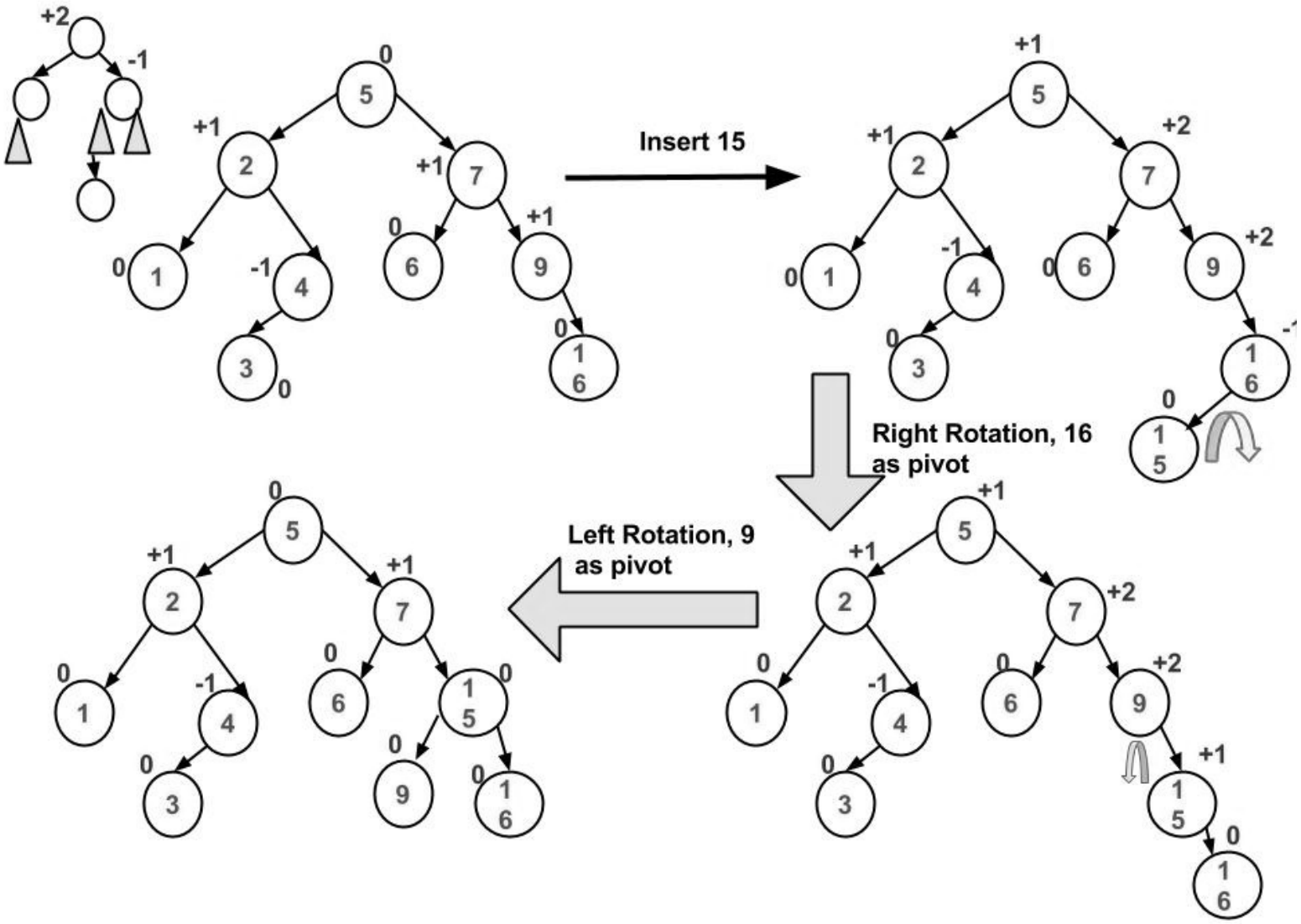
Example


+2
+1
Insert 45
+1
30
0
5
35
0
32
0
40
+2
30
+1
5
35
0
32
+1
40
0
45
Left rotate, node with value 30
Taken as pivot
0
35
0
30
+1
40
0
5
0
32
0
45

Example


Insert node 7
Left rotation,
5 as pivot
Right rotation,
10 as pivot

Example


Insert 15
Right Rotation, 16 as pivot
Left Rotation, 9 as pivot