

CSCE 221

Problem Set 10

Jacob Purcell, Texas A&M, Student

I.

The following diagrams show the steps taken to balance the binary tree, all arrows indicate pointers, blue shows formed connections, red shows broken connections, and orange shows cursor pointers that are used for keeping track of nodes while the rotations occur. Eq. 1 will be used to calculate the balance of the binary search tree. The null pointer has a height of -1.

$$Balance = h_L - h_R \quad (1)$$

and the condition must hold that

$$|Balance| \leq 1 \quad (2)$$

Figure 1 unbalance starts at the left child of the root node, which has a balance of 3 since the left child has a height of 2 and the right child is the null pointer, so a right rotation is performed.

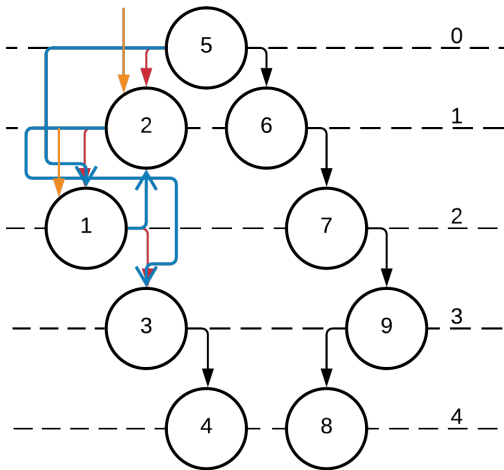


Fig. 1. The initial tree plus its first rotation.

Now the right child of the first unbalanced node (*Left subtree*, *Depth* = 1) has a height of 2 with the left child being the null pointer (balance -3) so a left rotation is performed.

This completes the first rebalancing operation, Left-Right Rotation. Scanning the rest of the root's left subtree, the following values correspond to the the balances

2 : -1

1 : 0

3 : -1

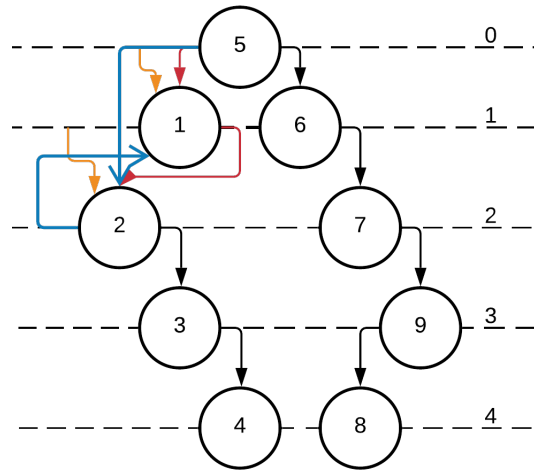


Fig. 2. Result of the preceeding rotation and a left rotation at the same position.

4 : 0

so the root's left subtree is balanced. The next imbalance occurs at the first node on the root's right subtree, with a balance of -3, so a left rotation must be performed.

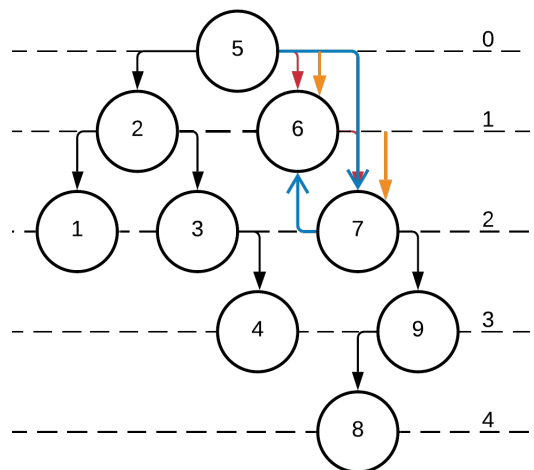


Fig. 3. Results of preceeding rotation and shows the next left rotation.

This completes the left rotation operation. The balance of the root's left subtree is now

7 : -1

6 : 0

9 : 1

8 : 0

with the root having a balance of 0. The tree is now fully balanced, however this process only required 2(*RL*, *LL*) of the 4 operations(*RL*, *LR*, *RR*, *LL*).

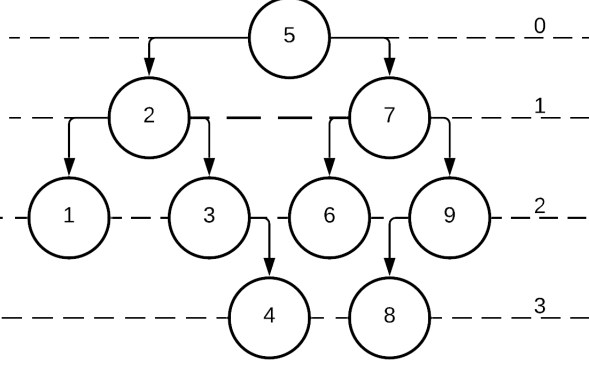


Fig. 4. AVL tree.

II.

As shown in Fig. 5, the tree is initially balanced but once 12 is deleted, the node that carries 11 has a balance of 2, so a right rotation must be performed(Fig. 6). Now the root node has a balance of 2 so a right rotation must be performed at the root node(Fig. 7). The tree is now balanced and 12 has been removed(Fig. 8).

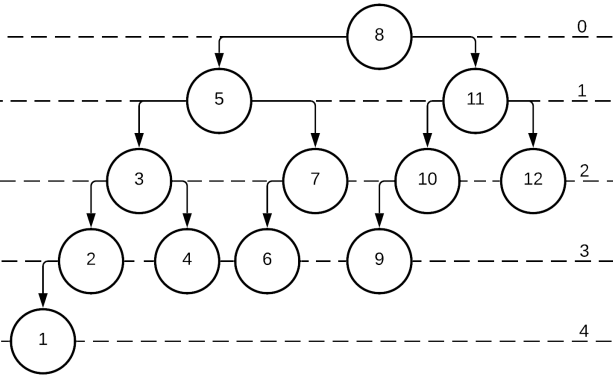


Fig. 5. Initial AVL tree.

III.

As proven in Problem Set 9, the equation relating number of nodes in a tree is

$$Nodes = 2^{height-1} - 1 \quad (3)$$

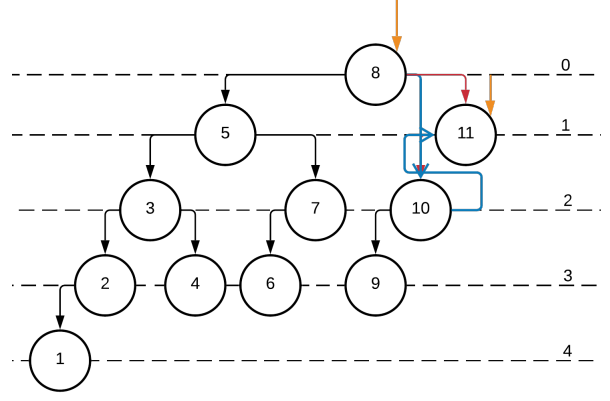


Fig. 6. Deleted value unbalanced the left subtree, therefore a right rotation is performed.

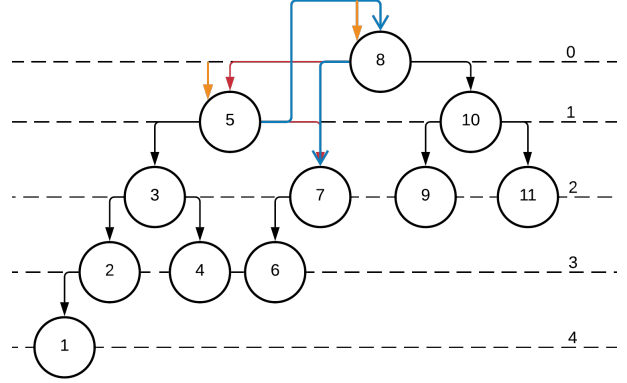


Fig. 7. Another rotation must be performed to maintain tree integrity.

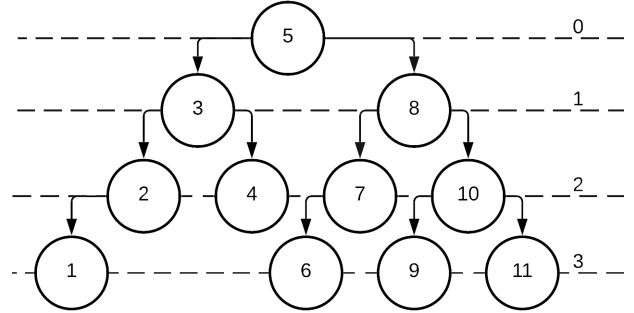


Fig. 8. Final result of removing 12.

To find the minimum number of nodes in a tree with height 13, first we find the full tree at height 12, then add one leaf node. This results in the following equation,

$$Nodes = 2^{12-1} - 1 + 1$$

$$Nodes = 2^{11}$$

$$Nodes = \boxed{2048}$$

IV.

An example of a double rotation is included in "mainps10.cpp," the definition of this rotation is in "tree_functions.h."