**Worksheet: AVL Trees Name:**

An Active Learning Approach to Data Structures using C

Worksheet: AVL Tree Practice

In Preparation: Read Chapters 8 and 10 on Bags and Trees, respectively. In you have not done so ready, do Worksheets 29 and 30 on Binary Search Trees.

Insert the following values, in the order that they are given into an AVL Tree.

1) 30, 20, 50, 40, 60, 70

2) 50, 22, 80, 70, 75, 73

50

22 80 C

70 B

NO A 75 D

73

Then I decided that maybe 50 is the unbalanced top node.

1. the right child is tallest by more than 1
2. The right child is heavy on the left side

We need to do 2 rotations

ROTATION 1: Rotate the right child to the right

1. B becomes the new root
2. B takes ownership of C, as its right child

50 A

22 70 B

NO D 80 C

75

73

Does not look promising, but I am moving on to the next rotation anyways:

Rotation 2: Rotate unbalanced top node to the left

To do the LEFT rotations, we follow the following steps:

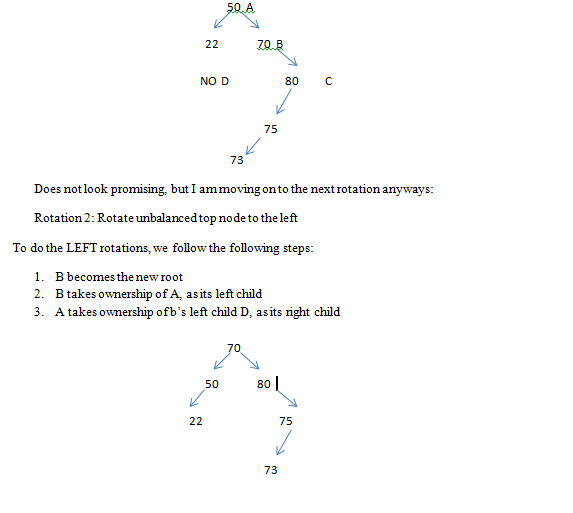
1. B becomes the new root
2. B takes ownership of A, as its left child
3. A takes ownership of b’s left child D, as its right child

70

50 80

22 75

73



50

22 80

70 A

75 B

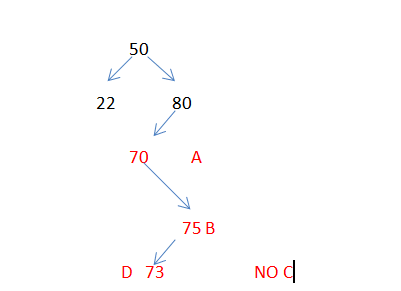
D 73 NO C

The tree is unbalanced:

1. the left child is tallest by more than 1
2. The left child is heavy on the right side

This indicates that we need to do 2 rotations.

FIRST ROTATION: we need to rotate the left child to the left

To do the LEFT rotations, we follow the following steps:

A

B

D C

1. B becomes the new root
2. B takes ownership of A, as its left child
3. A takes ownership of b’s left child D, as its right child

50

22 80

75

70

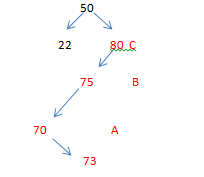
73

SECOND: Rotate unbalanced node to the right

The rule to do the Right rotations, we follow the following steps:

C

B

 A D

1. B becomes the new root
2. B takes ownership of C, as its right child
3. C takes ownership of b’s right child D, as its left child

Following this rules we end up with something like this:

1. A takes ownership of b’s left child D, as its right child

So we’ll get something like this:

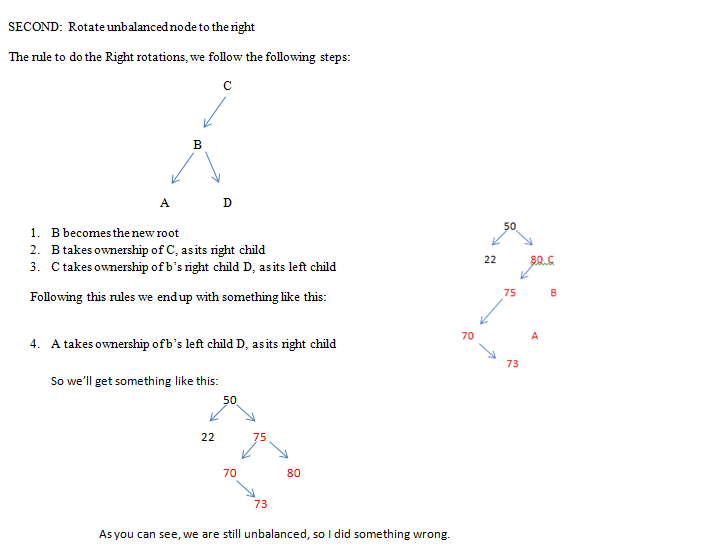
50

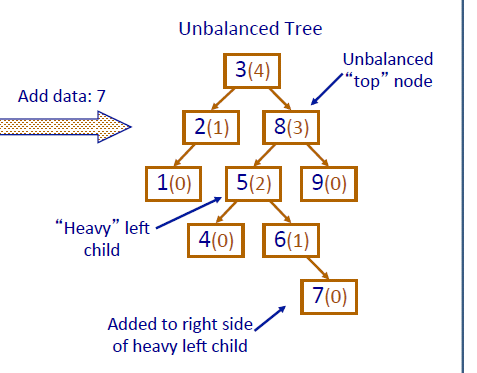
22 75

70 80

73

As you can see, we are still unbalanced, so I did something wrong.





The tree is STILL unbalanced:

1. the right child is tallest by more than 1
2. The right child is heavy on the left side

SECOND ROTATION: Rotate unbalanced node to the left

To do the LEFT rotations, we follow the following steps:

A

B

D C

1. B becomes the new root
2. A takes ownership of b’s left child D, as its right child
3. B takes ownership of A, as its left child

70

50 80

22 73 75

http://pages.cs.wisc.edu/~paton/readings/liblitVersion/AVL-Tree-Rotations.pdf

3) 75, 70, 100, 60, 80, 105, 77, 120

4) Take the tree from #3 and remove 60

5) Take the tree from #3. Remove 120. Remove 60.