Objectives: You will gain experience with AVL put implementation

To start the lab: Download and unzip the file: http://www.cs.uni.edu/~fienup/cs1520s20/labs/lab10.zip

Part A: Starting with an empty AVL tree, what would be the shape of the AVL tree be after put’s for keys: 90, 60, 50, 55, 40, and 53? (Show all necessary rotation(s) needed for each put.)

Part B: In lecture 23 we discussed the AVL tree rotateLeft method. For Part B, you need to implement the rotateRight method. Start by copying the rotateLeft method code, and paste it as the starting point for rotateRight. Now, modify the pasted rotateRight code is two steps:

1) updating the “pointers” to the nodes to do the right-rotation
   - HINT: Since rotateRight is a mirror image of rotateLeft, change all the left’s to right’s, and all the right’s to left’s

2) updating the balanceFactors for the rotRoot and newRoot nodes. You will need to use math similar to lecture 23 where were calculated values for the rotateLeft method. Use the next two pages to calculate needed balanceFactors for the rotateRight method. Remember the follow rules of algebra:

Algebra Review:
- a - (b - c) when removing the paretheses you get: a - (b - c) = a - b + c
- max(x, y) + c = max(x + c, y + c) should be clear from the following diagram:

Consider max(x, y) + c:

\[
\begin{align*}
\text{Consider max}(x, y) + c: & \quad \text{max}(x, y) = y \\
& \quad x \quad y \\
& \quad 0 \quad x + c \\
& \quad y + c \\
\end{align*}
\]

\[
\begin{align*}
\text{Consider max}(x+c, y+c): & \quad \text{max}(x+c, y+c) = y + c \\
& \quad 0 \quad x \quad y \\
& \quad x + c \\
& \quad y + c \\
\end{align*}
\]
- min(x, y) + c = min(x + c, y + c) similarly
- -max(x, y) = +min(-x, -y) should be clear from the following diagram:

Clearly, \( \text{min}(x, y) = x \)

\[
\begin{align*}
\text{max}(-x, -y) = -x \text{ and negating both sides gives:} & \quad \text{max}(-x, -y) = -x \text{ if } \text{min}(x, y)
\end{align*}
\]

- \( -\text{min}(x, y) = +\text{max}(-x, -y) \) similarly
Calculate the needed balance factors for the rotateRight method below:

Before right rotation:

rotRoot

newRoot

D

B

T_{A}\text{ height } h_{A}

T_{C}\text{ height } h_{C}

After right rotation at pivot:

newRoot

D

B

T_{A}\text{ height } h_{A}

T_{C}\text{ height } h_{C}

T_{E}\text{ height } h_{E}

Consider the balance factor formulas for rotateRight. We know from the above diagram:

oldBal(B) = h_{A} - h_{C}

newBal(B) = h_{A} - (1 + \max(h_{C}, h_{E}))

oldBal(D) = (1 + \max(h_{A}, h_{C})) - h_{E}

newBal(D) = h_{C} - h_{E}

To determine newBal(D), consider:

newBal(D) - oldBal(D) =

(See back for newBal(B) calculation)
Consider the balance factor formulas for rotateRight. We know from the above diagram:

\[
\text{oldBal}(B) = h_A - h_C \quad \text{and} \quad \text{newBal}(B) = h_A - \left(1 + \max(h_C, h_E)\right)
\]
\[
\text{oldBal}(D) = (1+\max(h_A, h_C)) - h_E \quad \text{and} \quad \text{newBal}(D) = h_C - h_E
\]

To determine newBal(B), consider:

\[
\text{newBal}(B) - \text{oldBal}(B) =
\]