

## Homework 4 Documentation

### LeftLeaningRedBlack Modifications and Additions

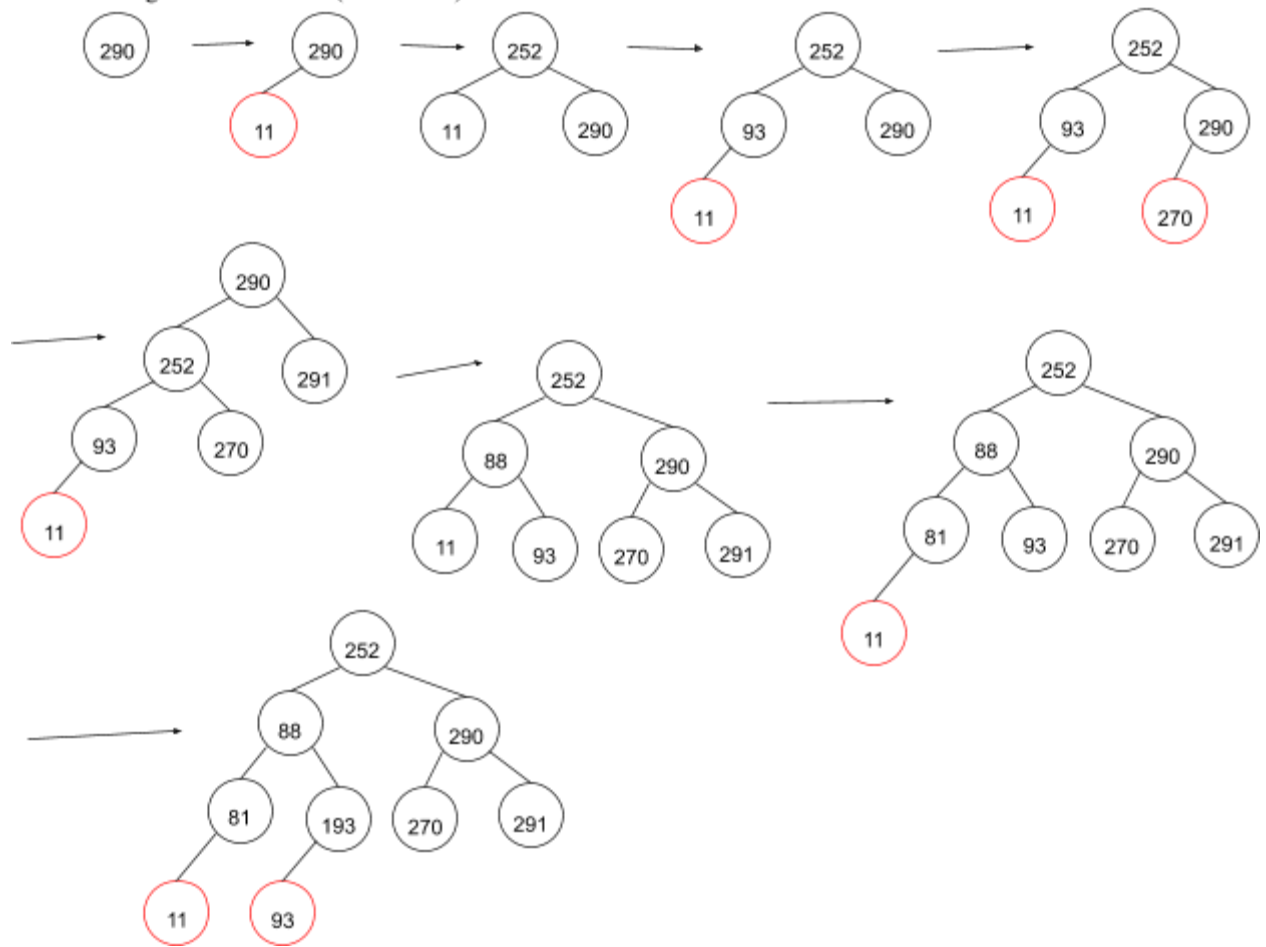
Since we were essentially given the code for left leaning red black trees, I only made modifications to existing code and added some functions to help me achieve what the assignment wanted. The files that I used from the supporting code package were VoidRef.h, LeftLeaningRedBlack.cpp, and LeftLeaningRedBlack.h. I added a Max, Min, ArraySize, SafeDelete, findParent, and findInsert function to the LeftLeaningRedBlack files. ArraySize and SafeDelete were taken from the code given to us in QzCommon.h. The max function finds the larger of two unsigned 32 bit integers while the min function finds the smaller of two unsigned 32 bit integers. FindInsert is exactly the same as the LookUp function that was given to us but returns a LLTB\_t pointer rather than void. FindParent is similar to FindInsert and LookUp but returns the parent of the node with the specified key. It first checks if the root value is equal to the key because the root does not have a parent. If the root is not equal to the key, it traverses the left side if the key is smaller than the root or traverses the right side if the key is larger than the root. I also changed the TraverseRec function to output the value, its color, its parent, and its parent's color.

### Main CPP file:

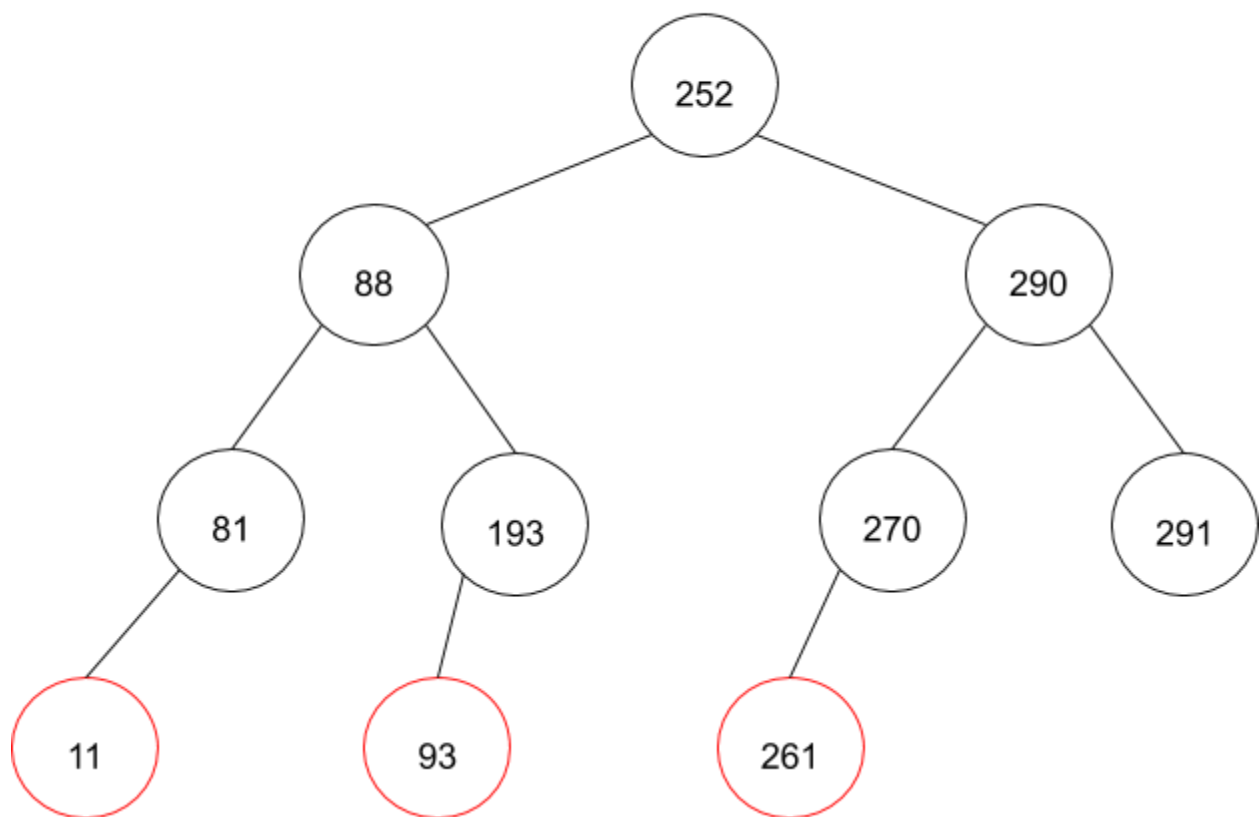
There is the main function and a generateRandom function in the main cpp file. The generateRandom function generates a value from one to three hundred. The main function loops ten times to insert ten random values into the tree. It calls the generateRandom function to create a random value and then inserts it into the tree. After the insertion is complete, it outputs the value, the color, the immediate parent, and the parent's color to the terminal. This is done ten times and then it outputs the inorder traversal of the tree. This traversal outputs the value, color, parent, and parent color to the terminal. Then the fourth inserted value is removed from the tree and the tree is adjusted. Then it outputs the inorder traversal with the nine values in the tree.

I compiled the rbTreeMain executable using cmake. In the folder with the CMakeLists.txt file, I type cmake ./ into the terminal. Then I entered make all to compile my code and create the executable rbTreeMain. The terminal outputs are in my log.txt file.

Left Leaning Red Black Tree (as inserted)



Left Leaning Red Black Tree (10 values inserted)



Removing 4th Inserted Value (93)

