

Luke Erlewein

Dr. Denton Bobeldyk

CIS 263

7/5/2025

Assignment 4

Output showcase

In my program, I used the binary trees printing using ASCII in C from [ximik777 on GitHub](#). I altered the code slightly to get what I wanted out of the printing and to adhere closer to what the assignment wants, but it mostly comes from there. The following images will showcase the different cases that can happen with the program.

```
Enter command: i 10
Inserted 10
10B

Enter command: i 5
Inserted 5
 10B
 /
5R

Enter command: i 1
rbInsertFixup Case 3: Right Rotation and Recolor
Inserted 1
  5B
 / \
/   \
1R  10R
```

Figure 1: Rotation from the addition of 1.

```

Enter command: i 15
rbInsertFixup Case 1: Recoloring
Inserted 15
    5B
   / \
  /   \
 1B   10B
      \
      15R

Enter command: i 12
rbInsertFixup Case 2: Right Rotation
rbInsertFixup Case 3: Left Rotation and Recolor
Inserted 12
    5B
   / \
  /   \
 1B   12B
      / \
     /   \
    10R  15R

Enter command: i 13
rbInsertFixup Case 1: Recoloring
Inserted 13
    5B
   / \
  /   \
 1B   12R
      / \
     /   \
    10B  15B
        /
       13R

```

Figure 2: Recoloring and rotation.

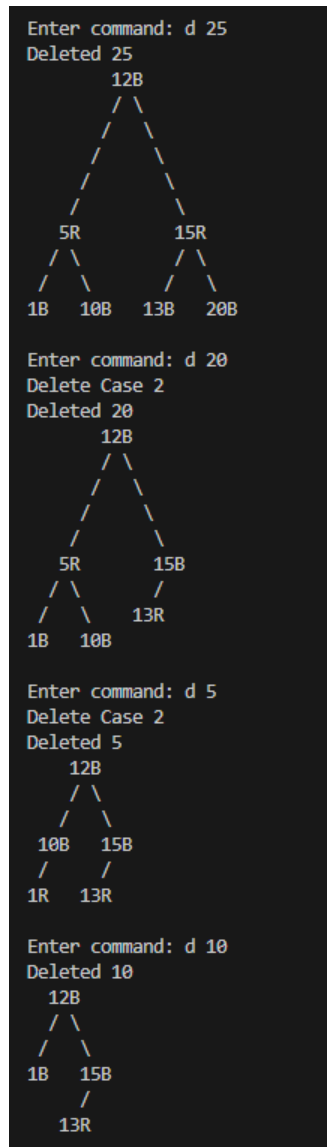


Figure 3: Transplantation and color adjusting from deletion.

Analysis question answer

When inserting the values found in the tree shown in the assignment, I do get the exact same tree. It is dependent on the order in which you insert the values, however. First, you start with the root, and then you go down to the next level. You then add the values in that level in a left to right pattern until you reach the end, where you go down to the next level, and so on. After inserting all the values, this is the following tree that I get.

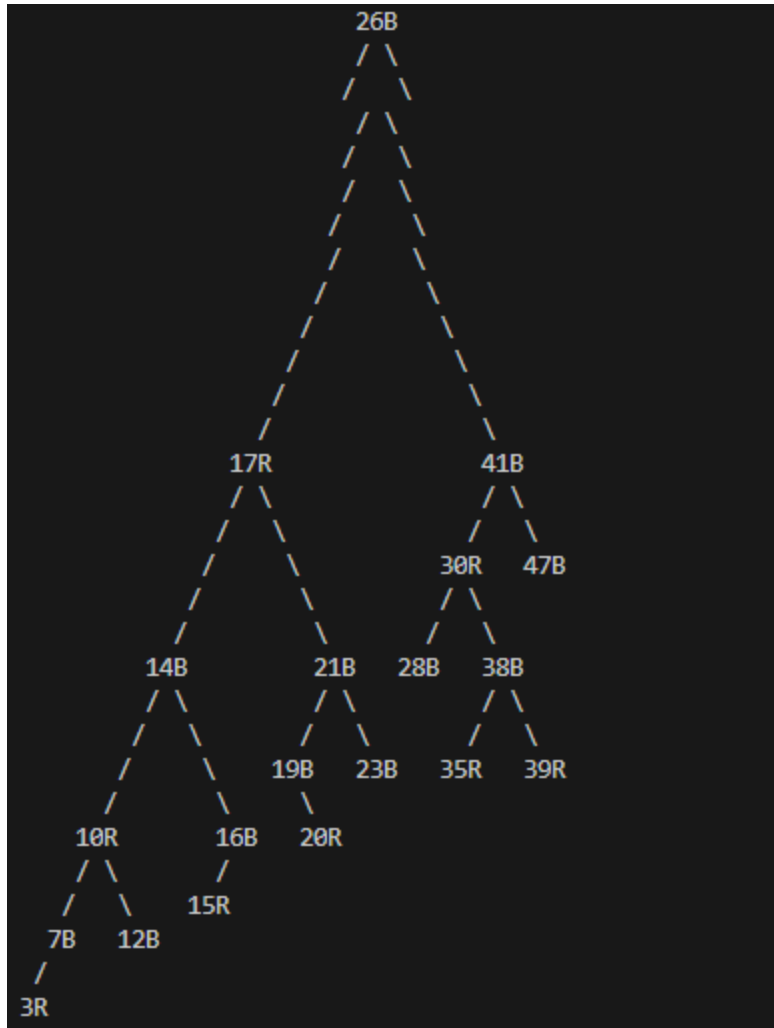


Figure 4: Red black tree from the assignment recreated in the program.

The reason why this is the only method that works is because when you add the values in a different order, it will still set it up in a valid red black tree, but not in the correct placement of nodes. For example the following is if you insert all the numbers from left to right, not depending on the vertical level of the node.

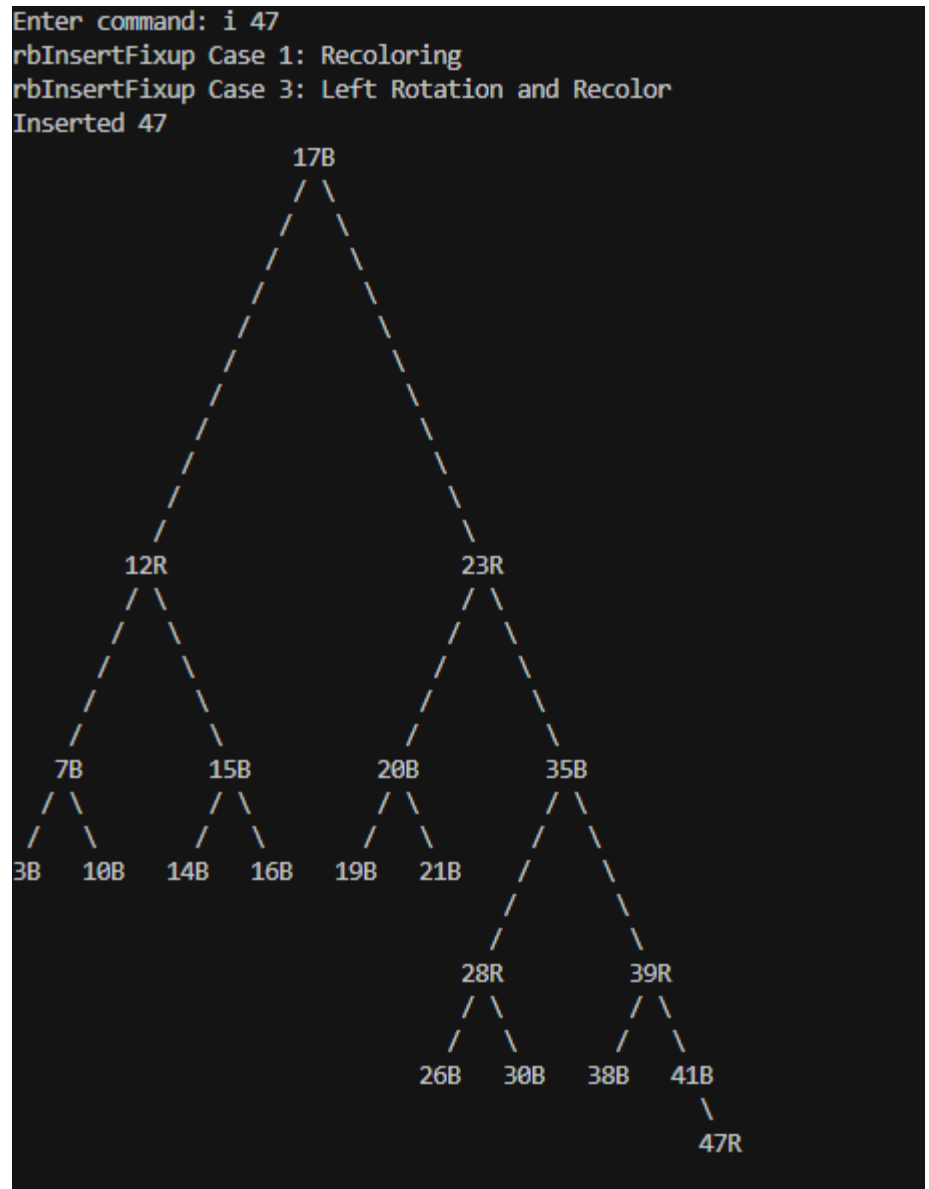


Figure 5: Red black tree in incorrect formatting due to improper input order.

As you can see, it is still a valid red black tree, but there was some adjustment that happened in order to fulfill all of the rules of red black trees. With the first number input being 3, there had to have been some rotation in order to get to 17 being the root node. This is because of the rule that there must be the same number of black nodes before any empty node. You couldn't just have a very long string of nodes coming off of the right side of the root being 3, because then it would fail that rule.