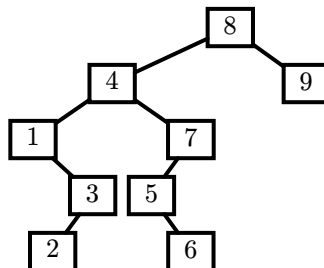


1 BST Insertion, Deletion, and Rotation (Lectures 17, 19)

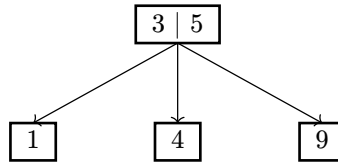
- (a) Suppose we have the BST shown below. Insert 0.



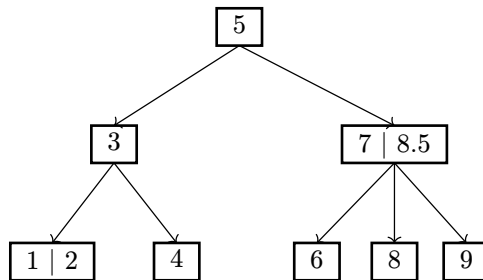
- (b) Now, starting from the original tree (before the insertion of 0), delete 4 using Hibbard deletion. Draw the new tree below. Check your answer with your TA's answer key.
- (c) Now, starting from the original tree (before the insertion of 0), rotate 8 to the right. Draw the new tree below. Check your answer with your TA's answer key.
- (d) To better understand tree rotations, go to https://hugocode.com/61b/bst_rotation and balance the tree. Suggestion: play until you're able to do it without thinking too hard.

2 2-3 Tree Insertions (Lectures 18, 19)

(a) Draw what the following 2-3 tree would look like after inserting 7, 6, 2, 8, and 8.5.

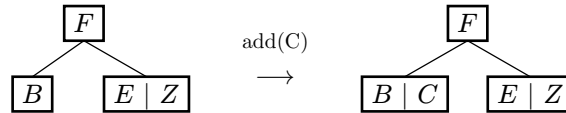


(b) Convert the 2-3 tree below to a left-leaning red-black tree.

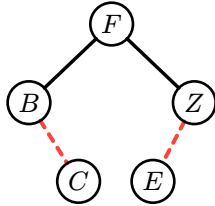


3 Deriving the LLRB Rotation Rules (Lecture 19)

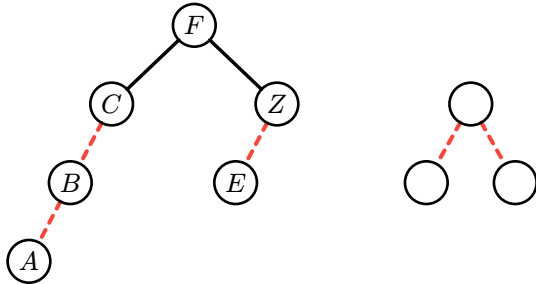
- (a) In 2-3 tree world, if we add C to the tree on the left, we get the tree on the right:



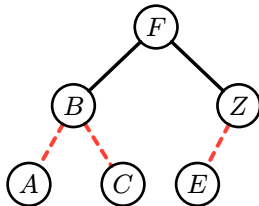
By contrast, in LLRB world, if we add C with a red link but don't perform any rotations, we end up with the tree below. What rotation do we perform to restore the proper 1-1 correspondence? Draw the entire tree after this rotation. Draw red links with a dashed line or label the line with "red".



- (b) If we now insert A into the tree you just rotated, we get the tree on the left. We see that A, B, and C are the wrong shape to be a temporary 3-item node: the correct shape is shown on the right. What rotation do we perform? Draw the entire tree after this rotation.



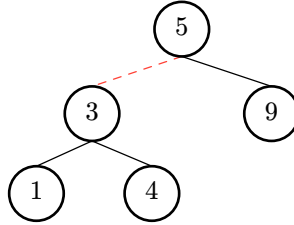
- (c) After the rotation, we have the tree below. If this were a 2-3 tree (i.e. a B-tree where the limit is 2 items), we'd call **split(B)**. On an LLRB, we instead call **colorFlip(B)**. Draw the result after the color flip.



- (d) Lastly, convert this LLRB into its equivalent 2-3 tree.

4 LLRB Insertions (Lecture 19)

Given the LLRB below, perform the following insertions and draw the final state of the LLRB. In addition, for each insertion, write the balancing operations needed in the correct order (**rotateRight**, **rotateLeft**, or **colorFlip**). If no balancing operations are needed, write "Nothing". Assume that the link between 5 and 3 is red and all other links are black at the start.



(a) 1. Insert 7

2. Insert 6

3. Insert 2

4. Insert 8

5. Insert 8.5 (warning: this one is a bit lengthy, you may want another sheet of paper)

6. Final state

(b) Convert the final LLRB to its corresponding 2-3 Tree.

(c) For more practice with LLRB insertion, try the game mode at <https://hugcode.com/61b/llrb/>.