CSCI 255: Lab #7 AVL Tree

Darwin Jacob Groskleg

Printed: Tuesday, March 10th, 2020

Contents

Questions

AVL Tree Insertions

Given the AVL Tree above, insert nodes in sequence to the tree. In each step,

- show the node(s) that becomes imbalanced if there is any,
- show the rotations that will fix the imbalance.

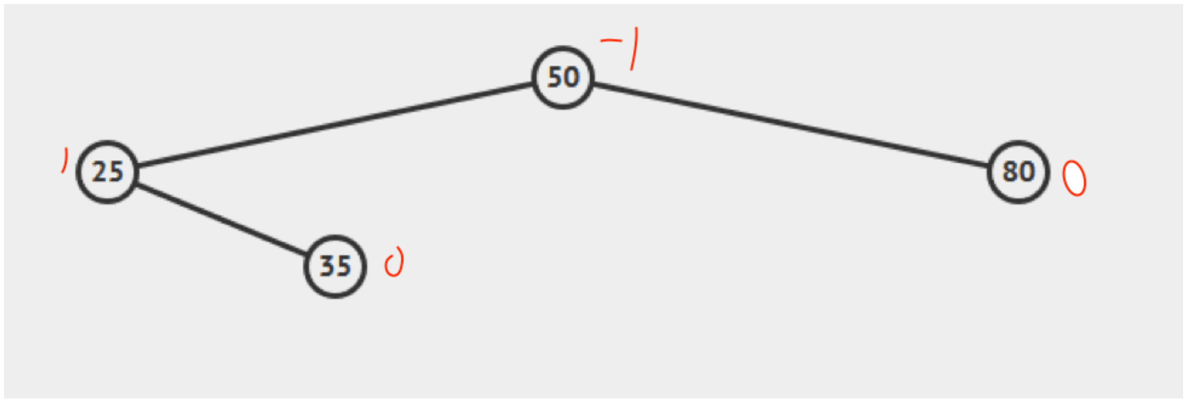

Figure 1: Initial state of AVL Tree.

a) Insert 40

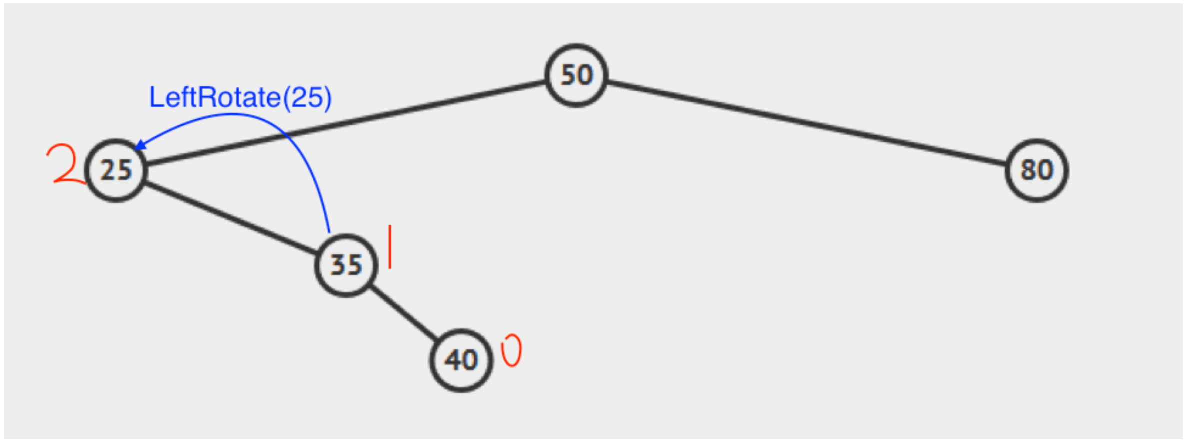

Figure 2: Unbalanced after inserting 40.

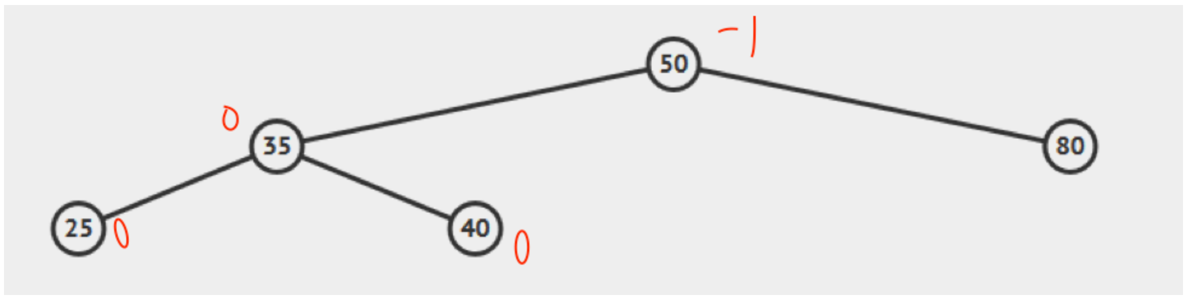

Figure 3: Balanced after a Left Rotate at 25.

b) Insert 36

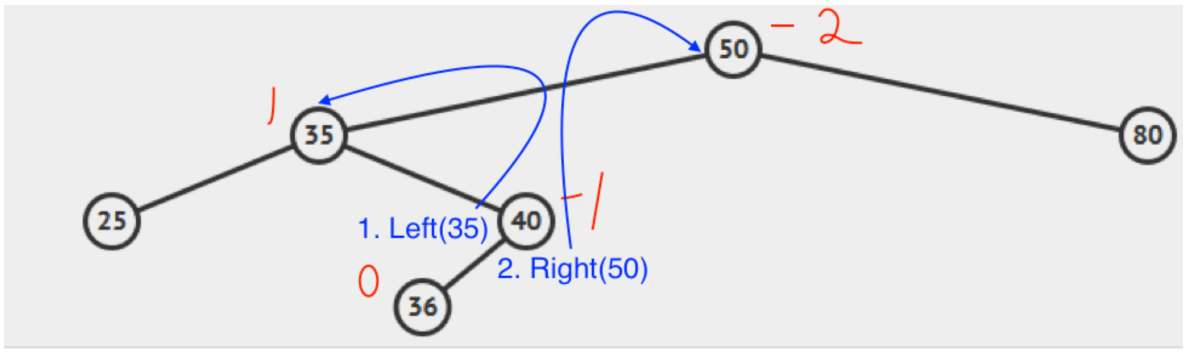

Figure 4: Unbalanced after inserting 36.

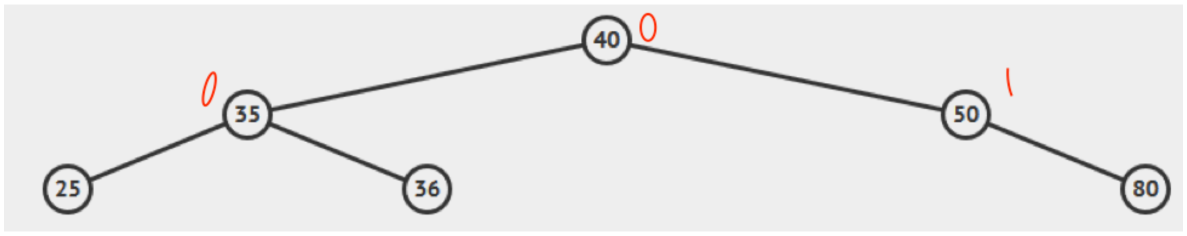

Figure 5: Balanced after a Left Rotate at 35, then a Right Rotate at 50.

c) Insert 37

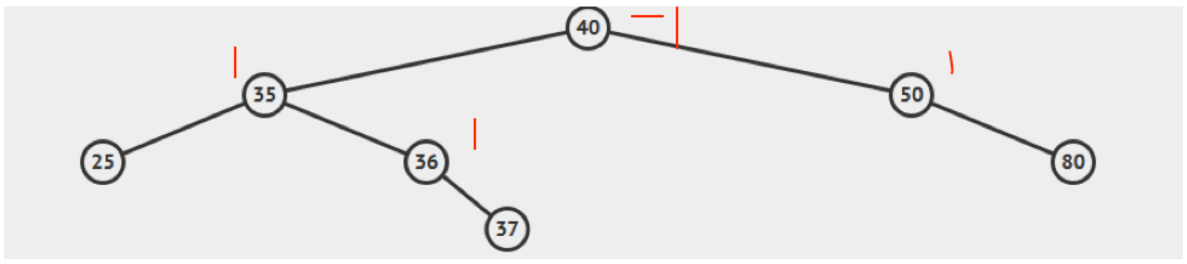

Figure 6: Balanced after inserting 37.

d) Insert 38

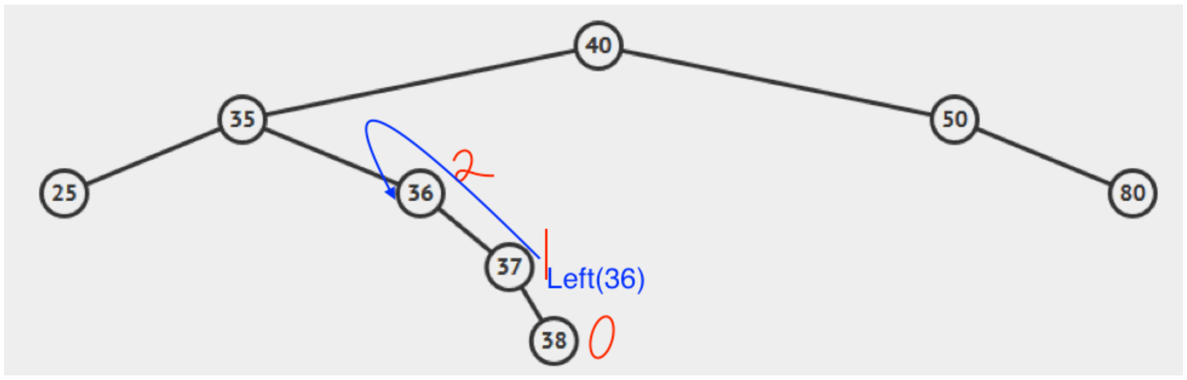

Figure 7: Unbalanced after inserting 38.

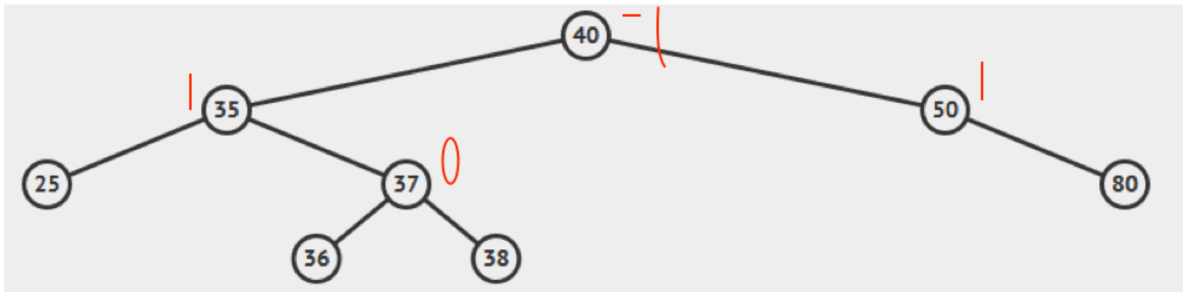

Figure 8: Balanced after a Left Rotate at 36.

e) Insert 39

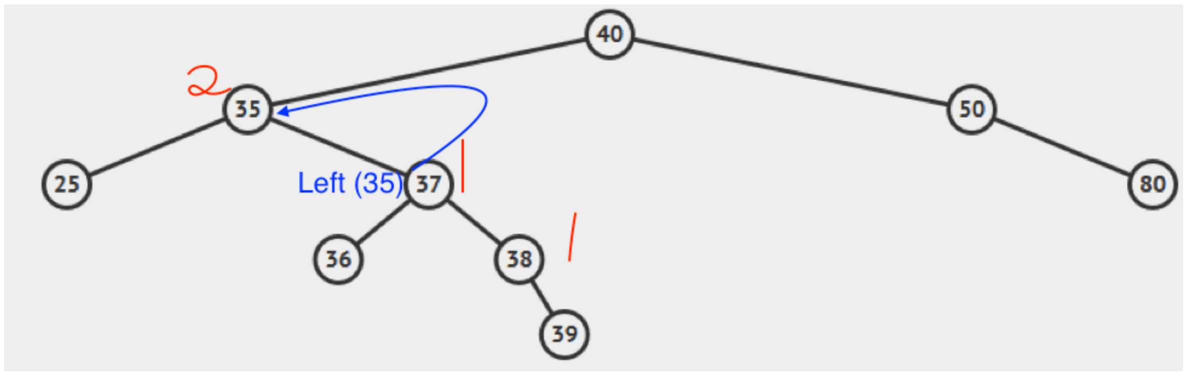

Figure 9: Unbalanced after inserting 39.

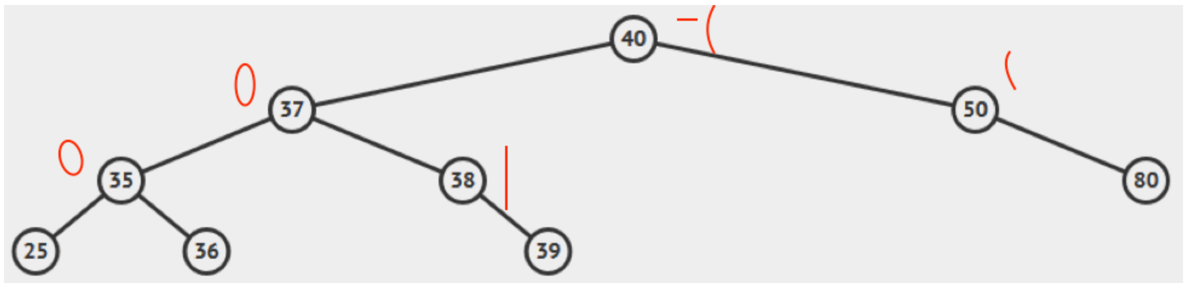

Figure 10: Balanced after a Left Rotate at 35.

This does match the expected state of the tree as seen on AVL animation website.

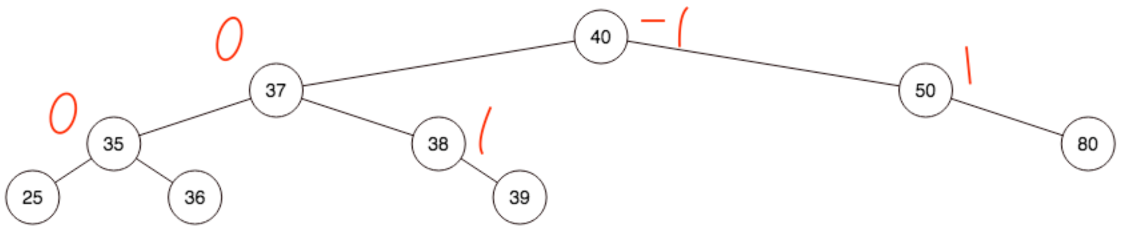

Figure 11: Expected tree after all insertions.

AVL Tree Deletions

Delete the following nodes in sequence from the resulted tree in Q1 (use delete by copy). In each step,

- show the resulted tree after delete by copy and,
- show how to balance the tree if needed.

a) Delete 36

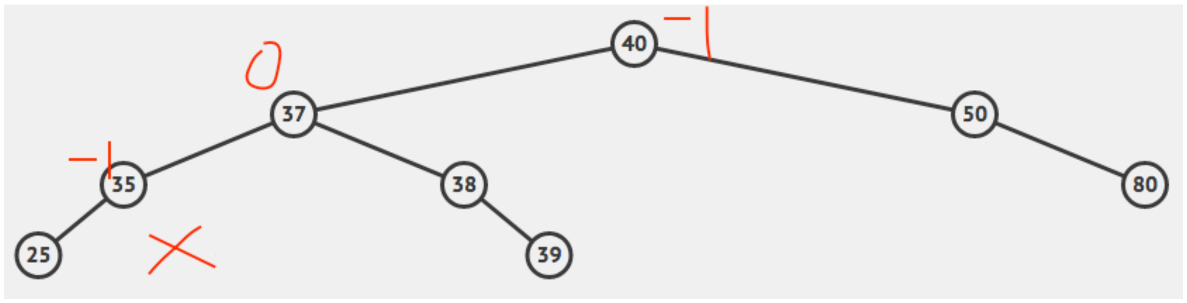

Figure 12: Balanced tree after deleting 36.

b) Delete 35

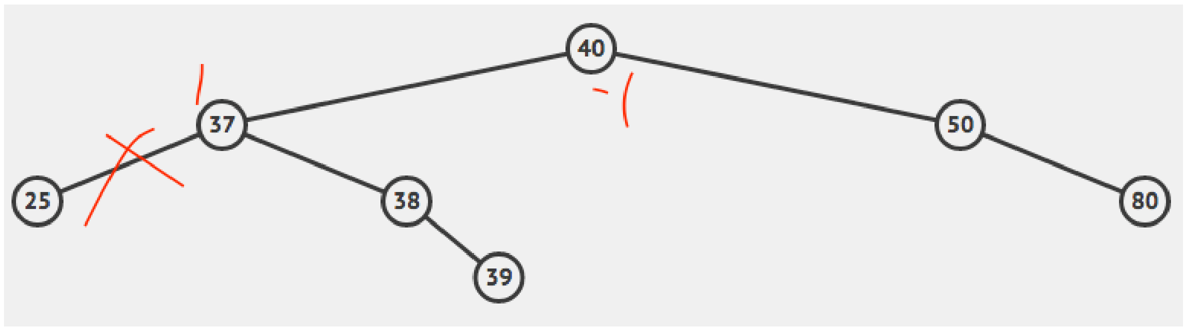

Figure 13: Balanced tree after deleting 35.

c) Delete 40

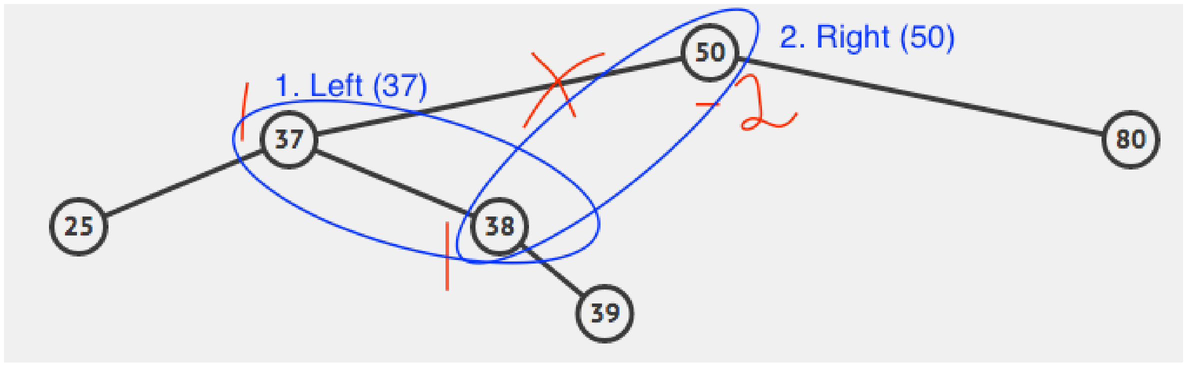

Figure 14: Unbalanced after deleting 40, the apex node.

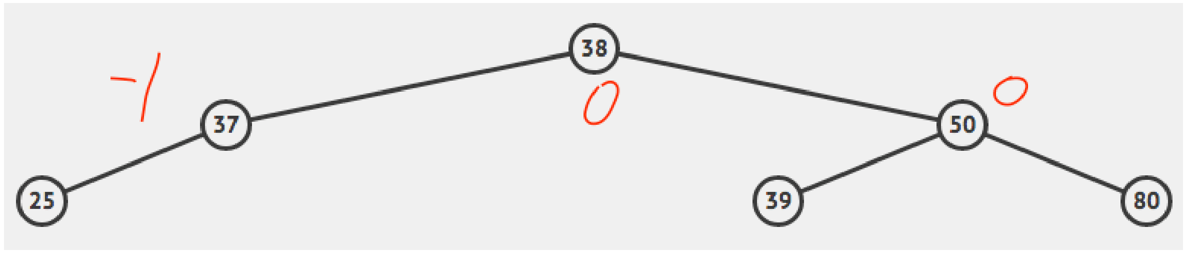

Figure 15: Balanced after a Left Rotate at 37, then a Right Rotate at 50.

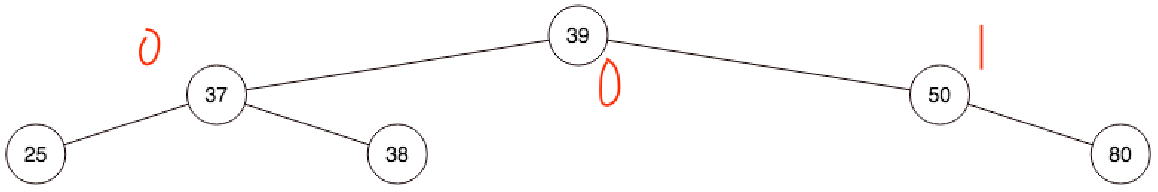

Figure 16: Notice that this would be the result if your ‘deleteByCopy’ method did swap with successor instead of predecessor.

d) Delete 37 (e originally)

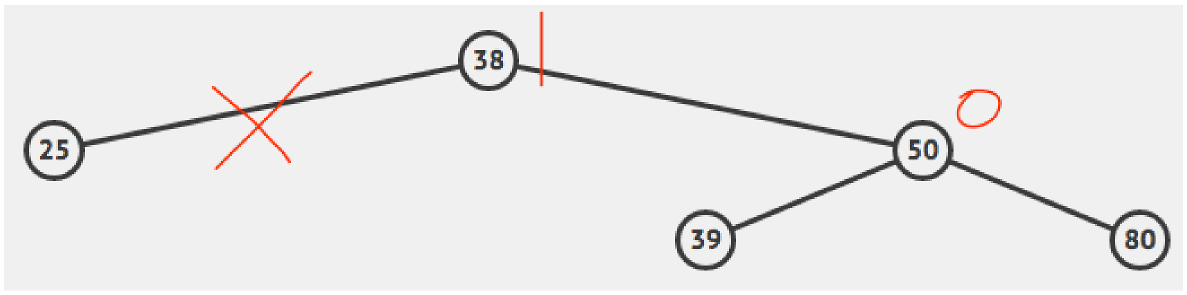

Figure 17: Balanced tree after deleting 25.

This does not match the final tree made on AVL animation website but it is nonetheless correct. The implementation detail of how ‘deleteByCopy’ picks a node to swap with, either its successor or predecessor, results in a different layout.

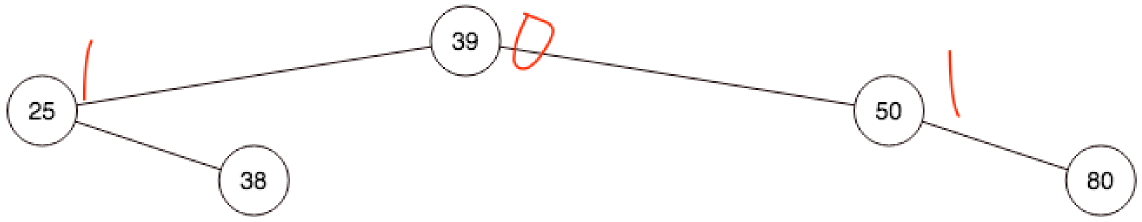

Figure 18: Expected final tree after all deletions from AVL website.