This is what I get for not testing thoroughly.

Here's the problem. The color of a node in a red-black tree is structural. That is, it belongs to the node, not the value stored in the node. The problem arises during the swap-to-leaf method. The bst swaps values and the color is stored in the augmented value sent to the bst. So the color travels with the value to be deleted, when it really should stay with the node.

You can stop working on red-black deletion until I come up with a work-around.

---

Well shoot I can't think off the top of my head what fixes that situation. I'm afraid of the solution too because I have a complete project, and doing something like giving bst nodes a color attribute would nuke my work.

---

I guess the easier solution is implementing a swap to leaf function just for rbt which would be a carbon copy of the one in bst except add three lines of code to swap back the colors. Sort of goes against the project mantra but is way easier for this one problem.

---

Actually, I think the proper solution is to provide a value-swapping function to the swap-to-leaf method. But I am loathe to change the specs at this point, so deletion for red-black trees will not be tested. You only need to implement insertion for red-black trees. Deletion for vanilla trees will still be required.