Weekly Assignment 8

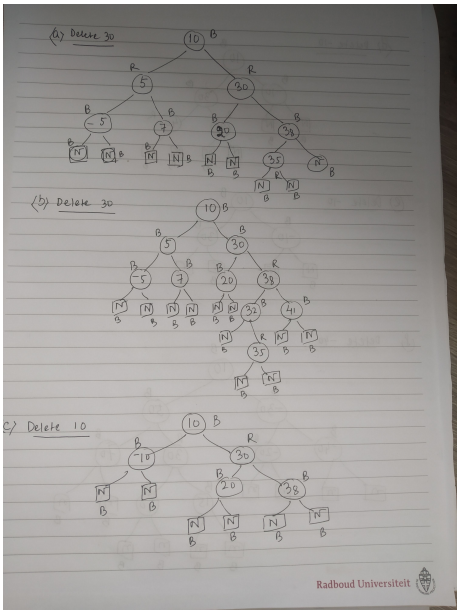
November 6, 2019

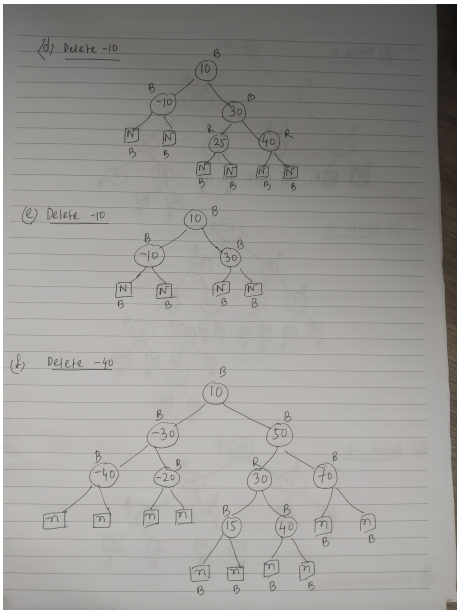
Exercise 1. *Weight: 25%*

Construct a Red-Black tree by taking the empty tree and then inserting nodes with the following keys in the given order: 10, 20, −10, 15, 17, 40, 50. Display the intermediate trees that arise during the construction.

Exercise 2. *Weight: 30%*

Construct the Red-Black trees that are obtained by deleting the specified nodes in the following Red-Black trees. Display the intermediate trees that arise during the construction.





Exercise 3. *Weight: 10%*

Let us define a *relaxed red-black tree* as a binary search tree that satisfies all the properties of a red-black tree except Property 2 ("The root is black"). Consider a relaxed red-black tree T whose root is red. If we color the root of T black but make no other changes, is the resulting tree a red-black tree? Either prove that this is always the case, or give a counterexample.

Exercise 4. *Weight: 10%*

Draw the complete binary search tree of height 3 on the keys $\{1, 2, ..., 15\}$. Add the NIL leaves and color the nodes in three different ways such that the black-heights of the resulting red-black trees are $2, 3$, and 4.

Exercise 5. *Weight: 25%*

In this exercise, we focus on the insertion of a new node N in a RBT. After running procedure *RB-Insert*(N), N is colored red.

1. What do we need to do when the RBT contained no element before the insertion?
2. What do we need to do when the parent P of node N is black?
3. Suppose that the parent P of N is red. Show that N has a grand-parent G who is black, and has an uncle U.
4. In case U is black, show that you can reestablish the properties of a RBT by at most 2 rotations and 2 nodes re-coloring.
5. What do we need to do when U is red and G is the root of the RBT?