# Algorithms and Data Structures

Drishti Maharjan

April 12, 2019

## Assignment 8

## Problem 8.1

a. Insertion in the order [13; 44; 37; 7; 22; 16]:

*Note: Visualization website was used to get the images of trees and the colors and screenshots of steps have been pasted*


Figure 1: Inserting 13 and 44

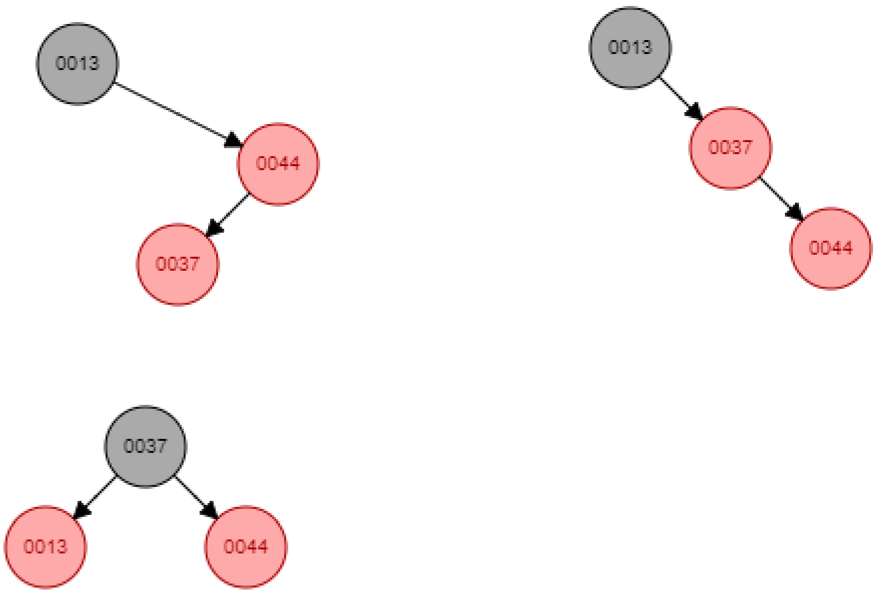



Figure 2: Inserting 37

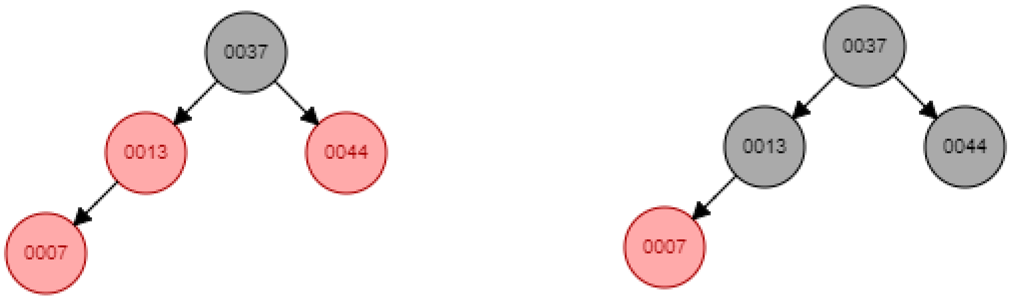



Figure 3: Inserting 7

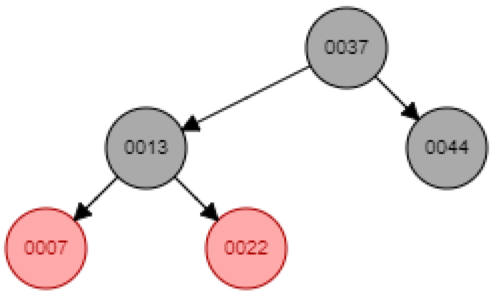



Figure 4: Inserting 22

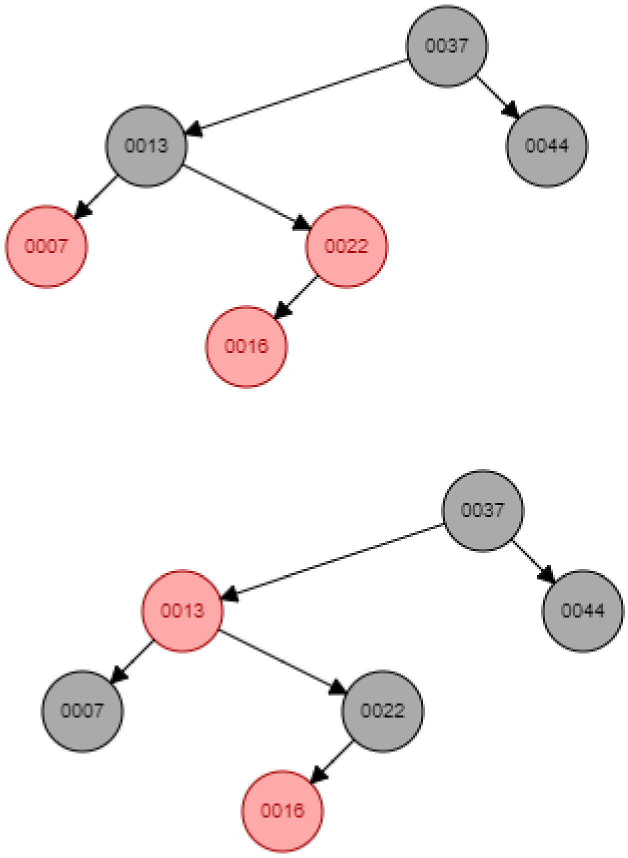
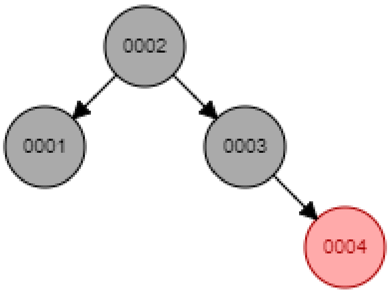
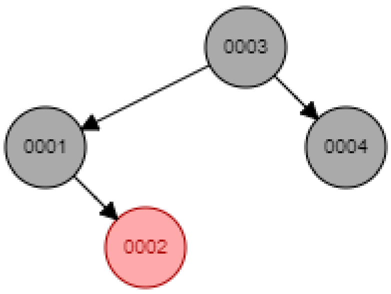
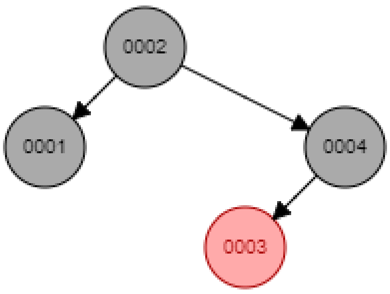


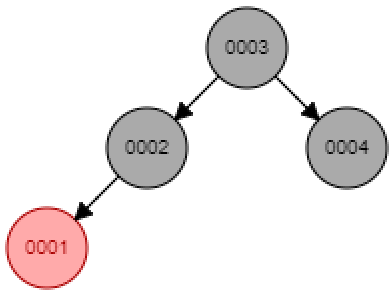

Figure 5: Inserting 16

b. All permutations of [1,2,3,4] were taken and red black tree was implmented with the help of C++ implementation for problem 8.2 and lecture notes. All nil nodes are black leaves, which are not shown in the trees here. And, possible 4 trees are drawn below:


0002
0001
0003
0004


0003
0001
0004
0002


0002
0001
0004
0003


0003
0002
0004
0001

c. **Proof by Induction**

*Base Case:*
N = 2. The node to be inserted is red and root is black by default, and no properties are violated so no fixup is required. No of red node = 1.

*Inductive Hypothesis:* Let's suppose that for a red black tree of n nodes, there exists at least 1 red node for $1 < n <= N$.

*Inductive Step:* We need to prove that for a red black tree of $N + 1$ nodes, there exists at least 1 red node.

While inserting the $N + 1$ element, we will have to consider 2 cases:

Case 1 : The inserted Red Node is a child of a black node. This is also equivalent to the base case, which has been proved already.

Case 2: Red Node $N + 1$ is inserted as a child of a red node. This violates the property of Red Black Tree, so we need to fix it to balance the tree. Let's look at the insertion cases that have X as a child of a red node P. This divides into further cases: [P-parent, G-grandparent, U-uncle]

Insertion Case 1: P,G,U is recolored and X remains red after the recoloring. It stil retains the same color after we move reference X to Grandparent of X. Thus, this remains a red node, giving us at least 1 red node.

Insertion Case 2: X is rotated about P and then G, this retains the red color of parent P. X and G is recolored while color of P remains the same i.e. Red. So,we have at least one red node.

Insertion Case 3: X remains red after the rotation of P about G. X remains red after the recoloring of P and G. Thus we have atleast 1 red node.

Hence, the claim has been proved.

## Problem 8.2

a. Code in testRBT.cpp and redblacktree.h, and Makefile in Makefile.txt