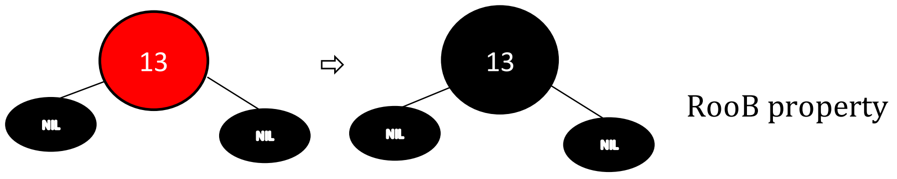
Algorithms and Data Structures: Homework 10

Due on April 24th, 2023, 23:00

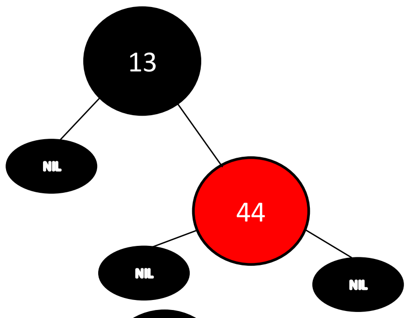
Problem 10.1: Understanding Red Black Trees

a) Draw (or describe by using preorder traversal) the red-black trees that result after successively inserting the values step by step in the following order [13, 44, 37, 7, 22, 16] into an empty red-black tree. You are required to draw (or describe by using preorder traversal) the tree after each insertion, as well as any additional recoloring and balancing.

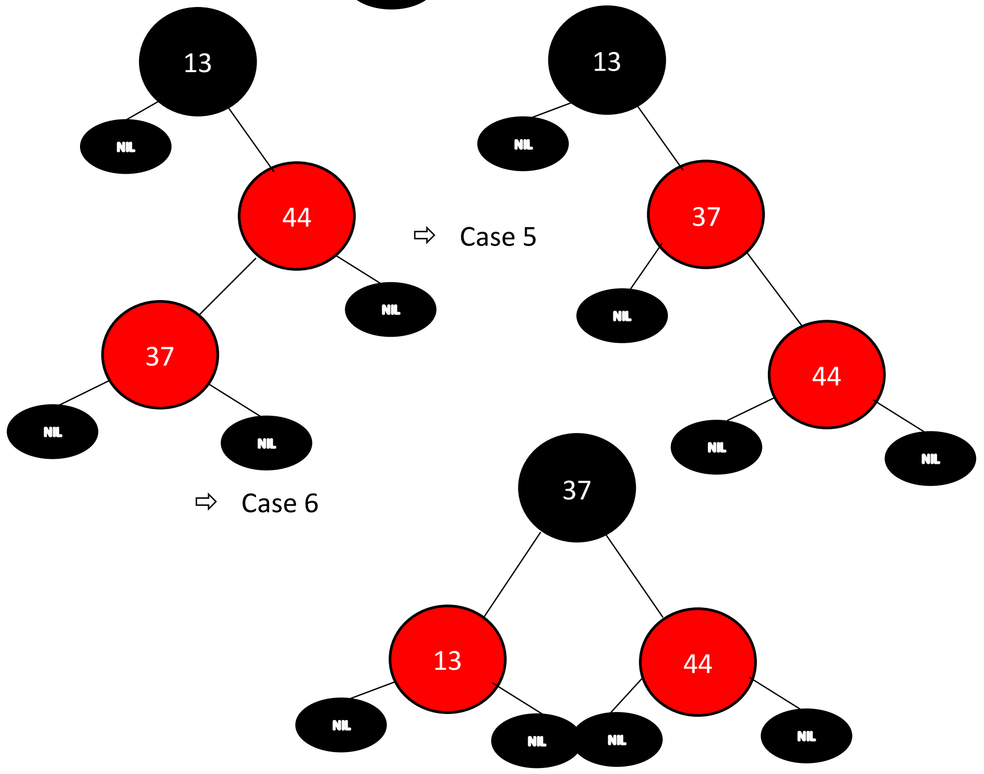
1. Insert 13:



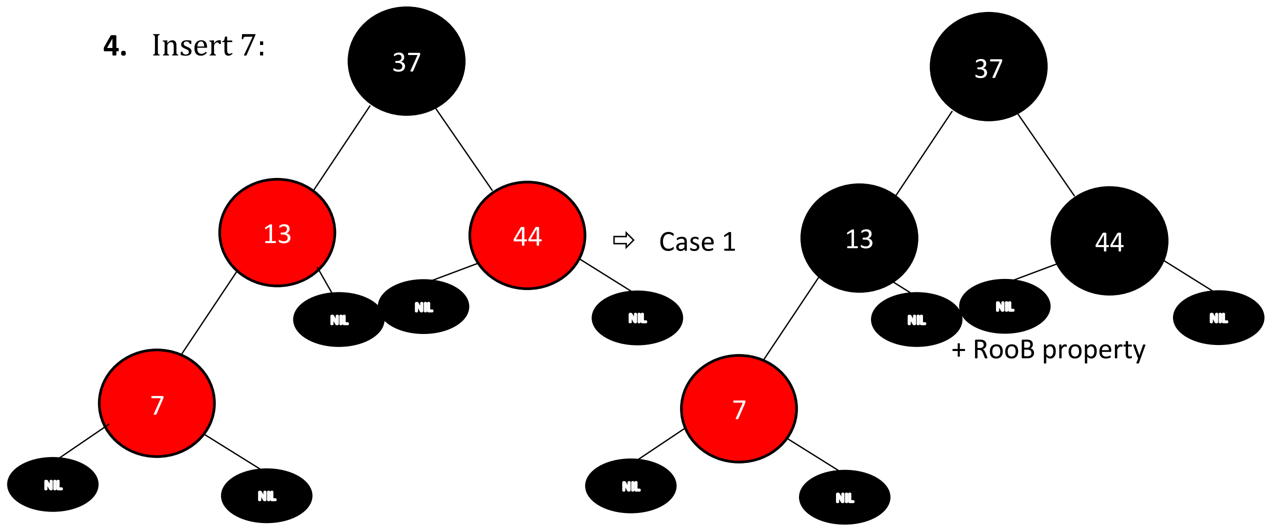
2. Insert 44:



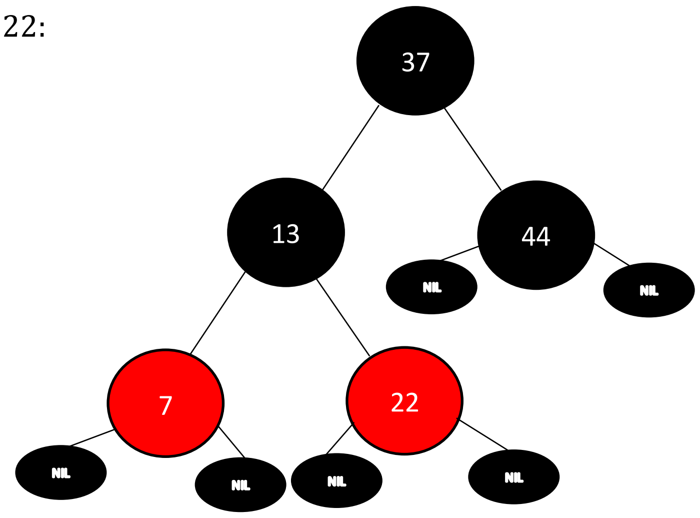
3. Insert 37:



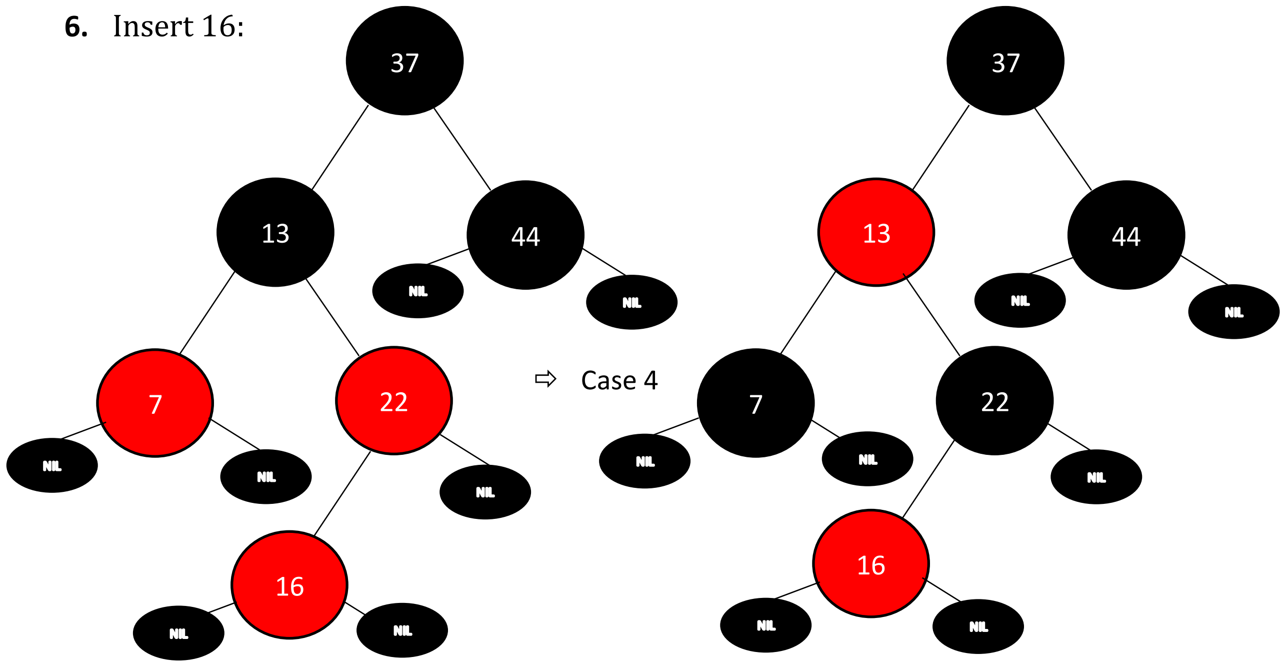
4. Insert 7:



5. Insert 22:

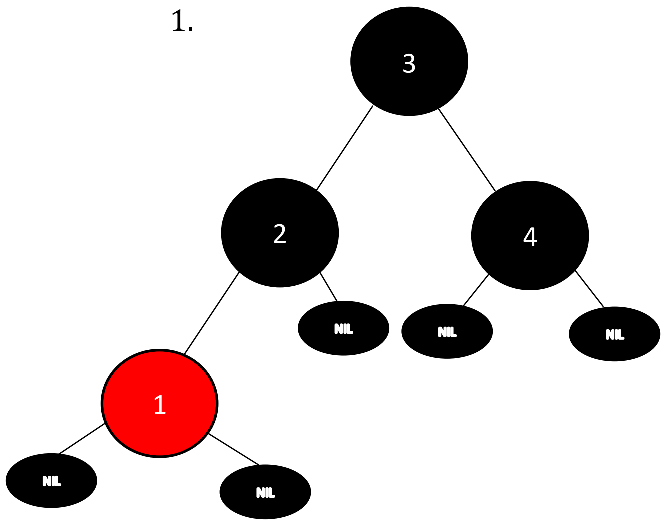


6. Insert 16:

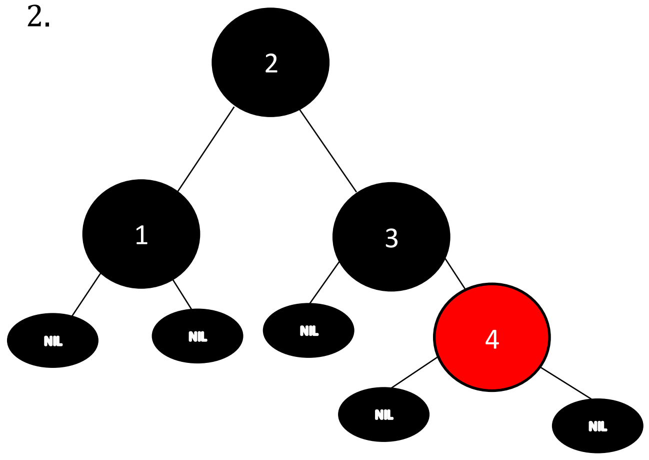


b) Draw (or describe by using preorder traversal) all valid red-black trees that store the values {1, 2, 3, 4}.

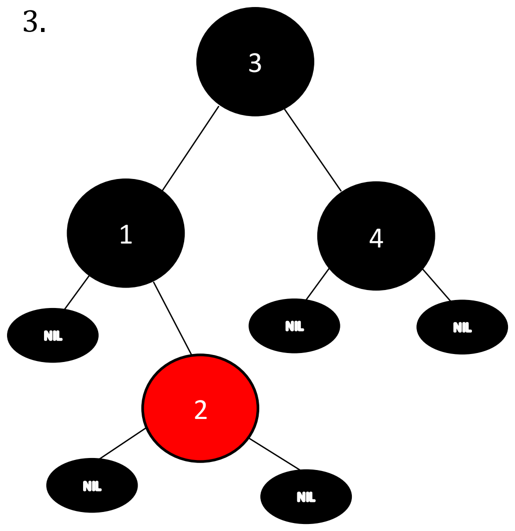
1.



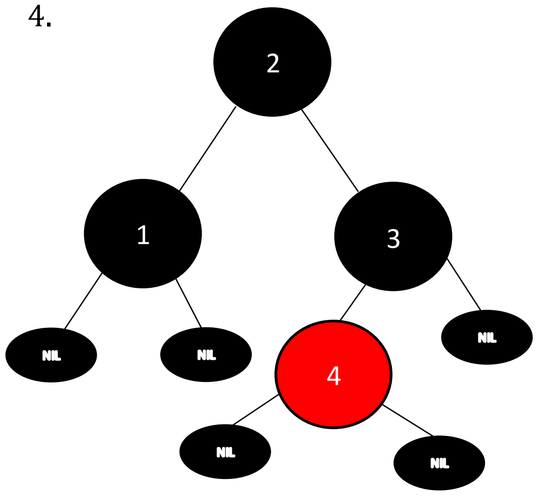
2.



3.



4.



Problem 10.2: Implementing Red Black Trees

Implement a red black tree (with integer nodes), closely following the specifications and algorithms from the lectures. Make sure you handle errors appropriately by printing messages or throwing exceptions. Your implementation has to be along the interface below with the following or equivalent components:

```
enum Color {RED, BLACK};
struct Node
{
        int data;
        Color color;
        Node ∗left, ∗right, ∗parent;
};
class RedBlackTree
{
        private:
                Node ∗root;
        protected:
                void rotateLeft(Node ∗&);
                void rotateRight(Node ∗&);
                public:
                RedBlackTree();
                void insertRB(int);
                void deleteRB(Node ∗&);
                Node ∗ predecessor(const Node ∗&);
                Node ∗ successor(const Node ∗&);
                Node ∗ getMinimum();
                Node ∗ getMaximum();
                Node ∗ search(int);
};
```

The definition of this class is inside the C++ file "RedBlackTree.cpp". A test function, based on Problem 9.1.a is inside the main function of the file "testRedBlackTree.cpp".