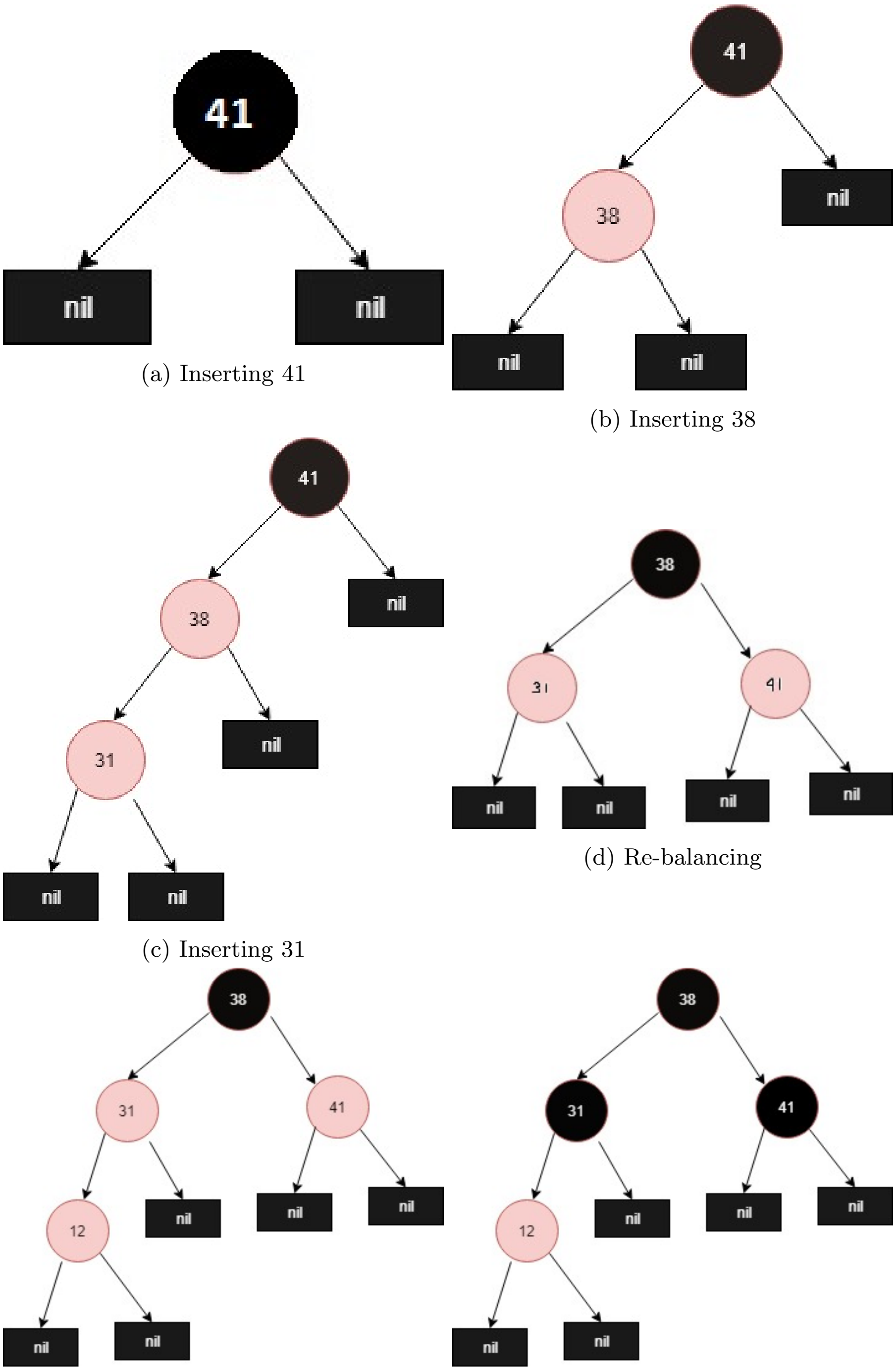
Algorithms Homework 4

Liam Dillingham

October 25, 2018

1 Question 13.3-2

Show the red-black trees that result after successively inserting the keys 41, 38, 31, 12, 19, 8 into an initially empty red-black tree.



(a) Inserting 41

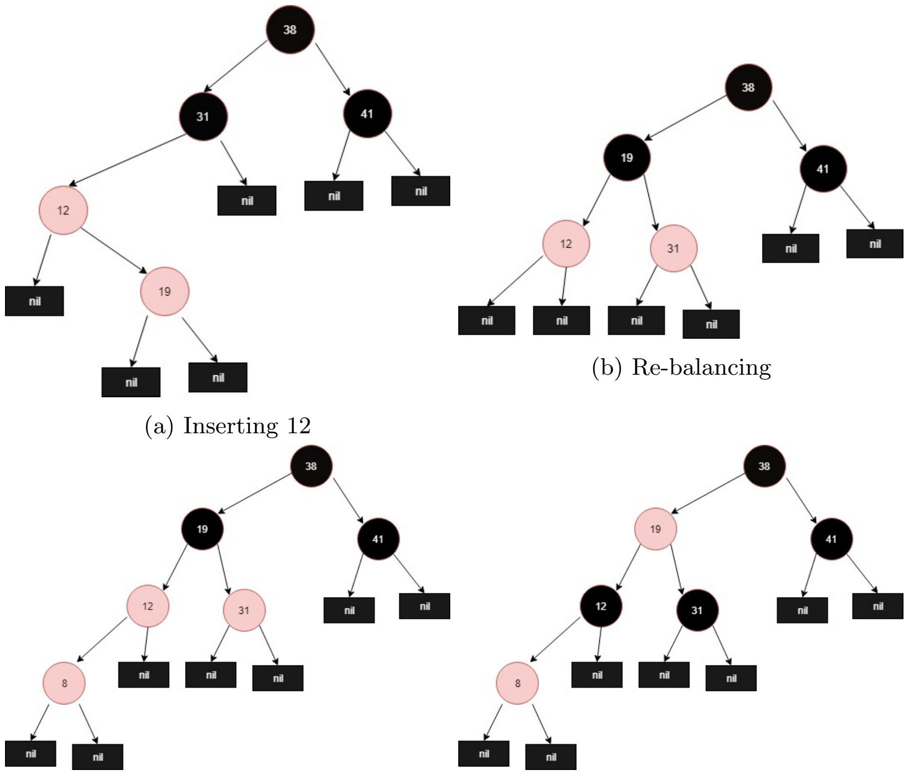
(b) Inserting 38

(c) Inserting 31

(d) Re-balancing

(e) Inserting 12

(f) Re-balancing



(a) Inserting 12

(b) Re-balancing

(c) Inserting 12

(d) Re-balancing

2 Question 13.4-3

In Exercise 13.3-2, you found the red-black tree that results from successively inserting the keys 41, 38, 31, 12, 19, 8 into an initially empty tree. Now show the red-black trees that result from the successive deletion of the keys in the order 8, 12, 19, 31, 38, 41.

---

3 Question 14.1-3

Show how OS-SELECT(T.root, 10) operates on the red-black tree T of Figure 14.1

It may be useful to have the code for OS-SELECT(x, i) below to walk through

```
OS-SELECT(x,i)
   r = x.left.size+ 1
   if i == r
     return x
   elseif i < r
     return OS-SELECT(x.left, i)
   else return OS-SELECT(x.right, i - r)
```

First we calculate the rank of the root. $r = 12 + 1 = 13$. since $i = 10$, and $10 < 13$, we recursively call OS-SELECT(T.root.left, 10).
Now we calculate the rank of the current subtree root. $r = 7 + 1 = 8$. since $8 < 10$, we call OS-SELECT(x.right, 2).
Now we are at the node with a key = 21, and we calculate its rank. $r = 2 + 1 = 3$. since $i < r$, we take the left subtree, and call OS-SELECT(x.left, 2)
Now we are at the node with key = 19, where $i = 2$, and rank $r = 0 + 1 = 1$. since $i > r$, we make the call OS-SELECT(x.right, 1).
Now we are at node key = 20. $r = 0 + 1 = 1$ and $i = 1$ thus, since $i == r$, we have our 10th smallest node, which is node where key = 20.

4 Question 14.1-5

Given an element x in an n-node order-statistic tree and a natural number i, how can we determine the ith successor of x in linear order of the tree in $\mathcal{O}(\lg n)$ time?

---

determining the ith successor of a node x is similar to determining ith smallest element, except that we pretend that x is the smallest node in the tree. So determining the ith successor in x is essentially taking the rank of x and adding it to the ith smallest element greater than x. So for example, if the rank of x is 4, then finding the 3rd successor of x is equivalent to OS-SELECT(T.root, x.rank + 3) = OS-SELECT(T.root, 7). or to word this generally, we can use the pseudocode function from the book OS-RANK(T,x) and plug it into our OS-SELECT function.

```
OS-SELECT(T.root, OS-RANK(T, x) + i)
```

5 Question 14.1-6

Observe that whenever we reference the size attribute of a node in either OS-SELECT or OS-RANK, we use it only to compute a rank. Accordingly, suppose we store in each node its rank in the subree of which it is the root. Show how to maintain this information during insertion and deletion. (Remember that these two operations can cause rotations).

---

6 Question 14.1-7

Show how to use an order-statistic tree to count the number of inversions (see problem 2-4) in an array of size n in time $\mathcal{O}(n \lg n)$

Problem 2-4 states:
Let $A[1..n]$ be an array of n distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair (i, j) is called an inversion of A.

hi