Bowen Kruse

CSCI 232

5/20/2019

Homework 3 - AVL/Red-Black Trees

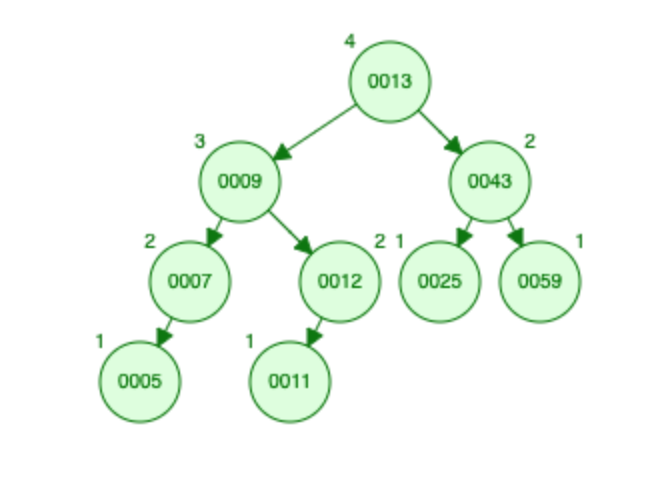
**Q1: Draw the resulting AVL tree from the following commands:**

*Insert: 5,7,8,9,12,13*

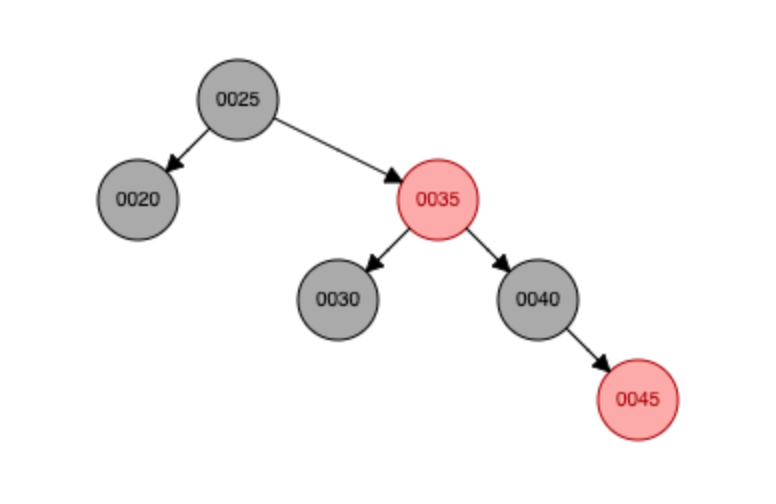
*Delete: 8*

*Insert 20, 25, 43, 59, 11*

*Delete 20*



**Q2: Provide an example of a legal red-black tree configuration that is NOT a legal AVL tree configuration.**



**Q3: Write pseudocode for deleting the node with the minimum value in an AVL tree.**

* Go left until reaching null left link(minimum)
* Replace the link to the node with its right link
* The recursive statement then sets the appropriate link for parents and updates the counts for all nodes in the path to the root.

**Q4: Refer to the binarytree.pdf file that was provided on the Google Drive. Draw the array representation of the resulting tree you drew in Q1. From within the array itself, how could you find the parent, right child, and left child of that node by only working with the array representation?**

[13, 9, 43, 7, 12, 25, 59, 5, 11]

0 1 2 3 4 5 6 7 8

If a nodes array index number is i, its left child is 2\*i+1, the right child is 2\*i+2, and the parent is (i-1)/2

Programming: (20 pts each)

Review documentation on using threads:<https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>

Threading Part 1:

Write a program that makes use of threading. You will have one thread count how many times a second has passed and the second thread will count how many times 3 seconds have passed. This will occur in an infinite loop and the counts will be printed to the screen. Indent the count of one of the threads for easier viewing.

**See Git “CSCI 232 Homework 3”**

Threading Part 2:

Write a program that has four threads. These four threads will share the use of an array. Let each thread take turns adding a random number to the array. Each thread should wait its turn and not be doing an insert when another thread is currently doing the insert. Print to the screen each action of each thread. (ex. “Thread <thread\_id> inserted <number>”)

**See Git Rep “CSCI 232 Homework 3”**

Working with files Part 3:

Write a program that reads in two different files and compares each line of the file. If the lines are different, output that they are different to an output file called “diff.txt”. Do not print any output to the screen, only to the output file. zipped project for each program or a GitHub link.

**See Git “CSCI 232 Homework 3”**