CS608 Programming Assignment 10

Complete Binary Trees and Complete Binary Search Trees

This assignment has two parts: Part 10A and Part 10B. If you successfully complete both, you will receive 15 points. If you successfully complete only one (either one), you will receive 10 points.

Programming Assignment 10A: Complete binary tree

Write a Java program to read a file, **inputData10.txt**, and create a complete binary tree from the data in the file. Assume that the data file contains 1000 integers. I will save this file in the same folder as your Java file when I run your program. In your program, just indicate the file name, "**inputData10.txt**", DO NOT include c:\.....

Output to contain:

1. The height of the tree
2. The number of nodes in the tree
3. Print the largest element in the tree
4. Ask me for an integer for searching. After I input an integer, your program prints if the number was found in the tree or not. Repeat this until I input 0.

Programming Assignment 10B: Binary Search Tree, AVL Tree, and CBS Tree

Write a Java program to read a file, **inputData10.txt**, and create (1) a binary search tree (2) an AVL tree and (3) a CBS tree from the data in the file. Assume that the data file contains 1000 integers. I will save this file in the same folder as your Java file when I run your program. In your program, just indicate the file name, "**inputData10.txt**", DO NOT include c:\.....

After you create the trees, you will search for input keys (elements) in each tree. Don't ask me for search keys. Instead, create a small array searchKeys = { 250, 2504, 2078, 2158,3502,7138,6230,9661,1330, 6136}, and search for each key in each of the three trees.

Remember CBS tree is really an array. You must implement CBS tree as an array as discussed in my notes.

Output to contain:

1. Height of each tree.
2. Calculate the time taken (in nanoseconds) to construct each tree. Don't include time taken to read the data file. Maybe you should read the file once and keep the data in an array.
3. Search each tree for the keys in the searchKeys[] array. Calculate the time taken by each tree for searching all the 10 keys. Don't calculate the time taken for searching each key. Get the time taken for searching ALL 10 keys.

Output outline:

Height of Binary Search Tree:	xxx
Height of AVL Tree:	xxx
Height of CBS Tree:	xxx
Time taken to construct Binary Search Tree:	xxx nanoseconds
Time taken to construct AVL Tree:	xxx nanoseconds
Time taken to construct CBS Tree	xxx nanoseconds
Time taken by Binary Search Tree to search for 10 keys in searchKeys array:	xxx nanoseconds
Keys (from searchKeys[] array) found on Binary Search Tree:	
Time taken by AVL Tree to search for 10 keys in searchKeys array:	xxx nanoseconds
Keys (from searchKeys[] array) found on AVL Tree:	
Time taken by CBS Tree to search for 10 keys in searchKeys array:	xxx nanoseconds
Keys (from searchKeys[] array) found on Binary Search Tree:	

Important note: I have made a text file, inputDataSample10.txt, with 1000 integers available to you in this week's folder on BB. You can use it to test your programs (both 10A and107B).

General instructions:

- If your program has several classes, include all of them in the same file and name your Java file CS60810Axxxxx.java (Assignment 10A) and CS60810Bxxxxx.java (assignment 10B), where xxxxx is your last name. **Example**: If your name is John Smith, name the file CS60810Asmith.java and CS60810Bsmith.java. **DO NOT SEND ZIP files.**

- Output must include **Your name, course number, and date (use Date class).** If any of the above items are missing, you will not receive full credit.

- Send your Java file as email attachment to CS608Assignment@gmail.com. Include your name and assignment number in the email subject.

Note: I will run your programs and grade them. If your programs do not compile (that is, show syntax errors, you will receive 0 for the programming assignment).