COMS S3134 Data Structures in Java

Midterm Exam, Summer 2016

NAME:_____

UNI:_____

There are **8** questions on this exam totaling 100 points. The exam is closed book and closed notes. No calculators or computers are allowed. You will have 90 minutes to complete this exam. Do not open this exam packet until instructed.

Print your name and UNI on the exam. Read and sign the academic honesty statement below. Next, write your name and UNI on the front cover of your blue book.

Place all answers in your blue book. Answers written on the exam sheet itself will not be graded. If you need more than one blue book, number your blue books and put your name and UNI on all of them.

At the end of the exam submit **BOTH** your blue book **AND** your signed exam sheet. Your exam will not be graded if we do not receive both pieces.

Academic Honesty Statement:

I certify that I have neither given nor received unauthorized help on this exam and that I did not use any notes, electronic devices, or other aids not specifically permitted. I will not discuss the content of this exam with anyone who is not taking the midterm at this time. I understand that any violation of this policy can result in an exam grade of zero and will be reported.

Signature:_____

1. (10 points total) Using induction, prove that the number of NULL child pointers in a binary tree of N nodes is N+1.

2. (8 points total) Using algorithms or data structure operations we have studied in class, name an example of an algorithm that satisfies each of the conditions below (2 points each):

    a. A constant time algorithm

    b. A log N time algorithm

    c. A linear time algorithm

    d. An exponential (2^N) time algorithm.

3. (10 points total) Assume you have an initially empty Stack. A program performs 10 push operations of the integers 0 through 9 in increasing order. That is, you would perform push(0), push(1), push(2), …, push(9). After each push, it performs any number of pop operations, such that the total number of pops is 10. For example, you might have a sequence like this:

    push(0), pop(), push(1), push(2), push(3), pop(), pop(), push(4), pop(), push(5), push(6), push(7), pop(), pop(), push(8), pop(), push(9), pop(), pop(), pop()

    It would result in the numbers being popped off in this order:

    0, 3, 2, 4, 7, 6, 8, 9, 5, 1

    As in the example, consider the sequence of values returned by the pops. For each of the following sequences, provide the sequence of stack operations that would generate it. If the sequence cannot occur, explain why.

    a. 5, 7, 6, 4, 3, 8, 2, 9, 1, 0

    b. 5, 8, 7, 6, 9, 4, 2, 3, 1, 0

4. (12 points total) You are given both the post-order traversal and the in-order traversal for a unique binary tree. Draw the corresponding tree and write down its pre-order traversal.

    Post-order traversal: B A F D C G

    In-order traversal: B F A G C D

5. (14 points total) A degenerate binary tree is a binary tree in which no node has more than 1 child. Write a Java method, *boolean isDegenerate(TreeNode root)* that, given a reference to the root node of a binary tree, returns true if the binary tree is degenerate. Assume a standard Binary TreeNode implementation with left child and right child references. (Assume that an empty tree is degenerate.)

6. (14 points total) Binary Search Trees T1 and T2 represent two sets. Write pseudocode for an algorithm to create a single Binary Search Tree T3 that contains the nodes from T2 which **do not** appear in T1. You can use any of the methods in the standard BinarySearchTree ADT. Assume that the root of the tree a public field, so the root of T2 can be accessed directly via T2.root.

7. (16 points total) Starting with an empty AVL tree, show the tree after inserting each of the following values sequentially: 20, 50, 70, 40, 30, 45. Be sure to also show all rotations performed (double rotations should show both single rotations involved).

8. (16 points total) You are given an application that matches words from song lyrics to song titles. The current system uses a hash table implementation. Each word is hashed to an entry in the table and each entry would then point to a linked list which contained references to every song that had that word in its lyrics. Assume that the database has N distinct words and M songs for which lyrics have been hashed.

    a. (4 points) Assuming no collisions in the hash table, what is the worst-case complexity for a method that will print **ALL** the titles of songs that contain a particular word. Give the answer in big-O notation in terms of N and/or M.

    b. (4 points) Assuming no collisions in the hash table, what is the worst-case complexity of a method that returns true if a particular word is in a particular song. Give the answer in big-O notation in terms of N and/or M.

    c. (4 points) How does the answer about the worst case cost in part b change if the song titles are stored using a binary search tree rather than a linked list?

    d. (4 points) Now assume that there are collisions in the hash table. Briefly describe how you would modify this particular hash table to handle collisions using the separate chaining method.