University of Toronto
Department of Electrical and Computer Engineering

Midterm Examination

ECE 345 Algorithms and Data Structures
Fall 2008

Print your name and ID number neatly in the space provided below; print your name at the upper right corner of every page.
The exam is ten (10) pages including the cover page; if not, report it to the instructor or TA.

Name:
ID Number:

- This is an *open book* exam. You are permitted to use the textbook for the course but nothing else is permissible. You may reference/use textbook page/section/algorithms if you wish so. Non-native English speakers may use a dictionary.
- Do all four problems in this booklet. Try not to spend too much time on one problem. Use terminology from the textbook. You must define any different terms before you use them.
- Write clearly and only in the space provided. Ask the proctor if you need more paper. **Nothing on the back of the sheets will be graded**.
- You have 110 minutes for this exam. Raise your hand if you have a question. Turn off your mobile phone and put it away.
- Do not give C code! Write pseudocode and analyze time/memory requirements of your algorithms when asked to receive full credit. All logarithms are base 2 unless otherwise noted.

Question	Points	Score	Grader
1	40		
2	20		
3	20		
4	20		
Total	100		

1. **[Asymptotics, Induction, BSTs, Recurrences] Short Answers, 6+6+7+7+7+7 points.**
Answer *only* in the space provided.

(a) State if the following is true/false. You need to justify your answer in the space provided to receive full credit.

$$2^{3^n} = \Theta(2^{3^{n+1}})$$

T **F**

(b) State if the following is true/false. You need to justify your answer in the space provided to receive full credit.

$$n^2 = O(\frac{n^3}{\lg n})$$

T **F**

(c) Use induction to prove that for all positive integers n,

$$1^2 - 2^2 + 3^2 - 4^2 + \ldots + (-1)^{n+1} n^2 = (-1)^{n+1} \left(\frac{n(n+1)}{2}\right)$$

Show clearly the induction base/hypothesis/step.

(d) Draw the *final* binary search tree that results after inserting the keys 65, 22, 37, 5, 2, 77, 4 *in that order*.

(e) Solve the following recurrence by using a *recursion tree*. Justify your answer.

$$T(n) = T(n/5) + T(4n/5) + \Theta(n)$$

(f) Solve the following recurrence.

$$T(n) = 4T(n/2) + n\sqrt{n}\lg n$$

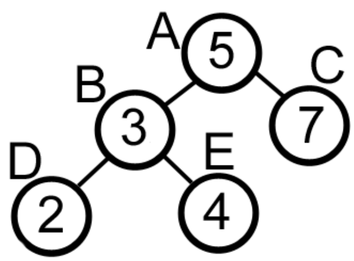
2. **Sorting, 10+10 points.**

   For the following two parts, explain in English how your algorithm works, justify its correctness and its running time. If you use an algorithm from the text, you don't need to rewrite it in its entirety, or reprove its complexity. You can just reference the page number in CLRS and give the function name. Partial credit for less efficient solutions. *Write clearly!*

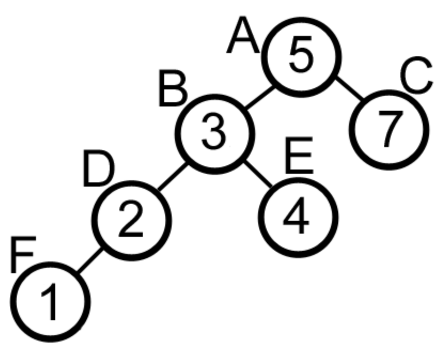
   (a) Given a list of n distinct unsorted numbers, describe an $O(n)$ algorithm that returns the *largest* difference between any two numbers.

   (b) Given a list of n distinct unsorted numbers, describe an $O(n \lg n)$ algorithm that returns the *smallest* difference between any two numbers.

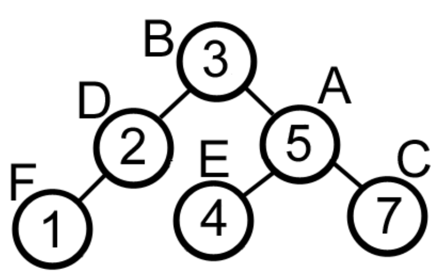
3. **Balanced BSTs, 10+10 points.** An AVL tree is a binary search tree with one additional structural constraint: For any internal node v, the heights of the children of v differ at most by 1. This is called the AVL Tree *balancing* condition. Here, *the height of a node v is defined to be the number of nodes in the longest path from v to a leaf, including v.* For example, consider the following AVL tree. Inside every node we have the key the node contains. The heights of the nodes in the tree below are $h(A) = 3, h(B) = 2, h(C) = 1, h(D) = 1, h(E) = 1$. If a node v does not have a right-child (or left-child), the height of the right-child (or left-child) of v is defined to be 0. In the tree below, the heights of node A's right-child (C) and left-child (B) differ by 1, while the heights of node B's right-child (E) and left-child (D) differ by 0.



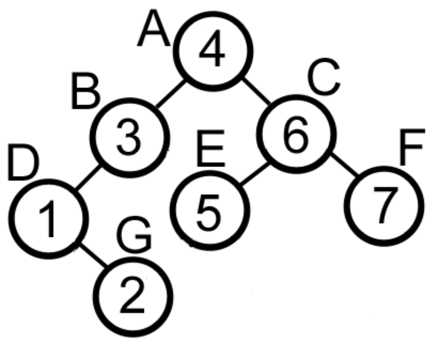
Nevertheless, if you insert 1, in the same way we insert in a Binary Search Tree (BST), the tree becomes unbalanced because node B (node A's left-child) has height 3, while node C (node A's right-child) has height 1.



In this case, we can rebalance the tree by performing a right-rotation at node A (that contains 5) as follows:



(a) You are given the following binary search tree. Why is this BST tree not an AVL tree? Use *at most* 2 rotations to make it an AVL tree. State the nodes you rotate on, the direction of each rotation (left/right) and show the resulting tree after each rotation.



(b) Let $min(h)$ denote the *minimum* number of internal nodes of an AVL tree whose root has height h. Note that we have $min(1) = 1$ and $min(2) = 2$. Argue that, for $n > 2, min(h) = 1 + min(h - 1) + min(h - 2)$. You do not need to use induction or contradiction. Your answer must be less than 8 lines to receive full credit.

4. **Greedy Algorithms, 20 points.** Given a set $\{x_1, x_2, \ldots, x_n\}$ of (not necessarily sorted) n points on the real line, design an efficient algorithm that determines the smallest set of unit-length closed intervals that contains all of the given points. For example, given the set $\{2, \sqrt{2}, 5\}$, one possible choice of unit-length closed intervals containing the given points is: $[1, 2]$ and $[4, 5]$. Prove your algorithm is correct and analyze its running time. Your grade will depend on the time required by your algorithm.

(DO NOT REMOVE – This page left intentionally blank)

(DO NOT REMOVE – This page left intentionally blank)