Computer Science B63 June 1, 2008
Scarborough Campus University of Toronto

Homework Assignment #2
Due: June 19, 2008, by 1:10 pm

1. *Please complete and attach (with a staple) an assignment cover page to the front of your assignment.*
2. *If you do not know the answer to a question, and you write "I do not know the answer to this question", you will receive 20% of the marks of that question. If you just leave a question blank with no such statement, you get 0 marks for that question.*
3. *Unless we explicitly state otherwise, you should justify your answers. Your paper will be marked based on the correctness and completeness of your answers, and the clarity, precision and conciseness of your presentation.*

Question 1. (10 marks) In this question, you must use the insertion and deletion algorithms as described in lecture for B-trees. Specifically, during an insert, split a node only when

a. (5 marks) Insert into an initially empty B-tree with minimum degree 2 (i.e., $t = 2$, also known as a 2-3-4 tree or a (2,4)-tree) each of the following keys, in the order in which they appear in the sequence: $F, S, Q, K, C, L, H, T, V, W, M, R, N, P, A, B, X, Y, D, Z, E$

Show only the intermediate tree results just before some node must split, and show the resulting B-tree T. Be sure to include the order of keys in each node.

Hint: You should obtain FQT as a root node in one of your intermediate results.

b. (5 marks) From the B-tree T you built above, delete nodes L and N (in this order), and show the resulting tree. Show the order of keys within each node. (Show only the final tree.)

Question 2. (28 marks) A summer student has been asked by a law firm to build a system to digitally store their files recording legal cases and monetary settlements in a computer system. Beyond just inserting and deleting records into the system, they also want to be able to retrieve the largest settlement before or after a given date, and the number of settlements (i.e., records in the system) that were reached within a given time period. Since there are many files to store, and they want the data returned quickly, they require a $O(\log n)$ worst-case running time for each operation, where n is the number of files stored in the system.

Unfortunately, this student hasn't yet taken CSC B63, so the student is asking for your help. Your job is to design a data structure D based on AVL trees to support the following five operations in $O(\log n)$ worst-case running time:

- Insert(D, x): Insert into D a record of a settlement, which includes (among its fields) a date d the settlement was reached and amount of the settlement a. (You may assume that there is no previous record in D for date d.)
- Delete(D, x): Given a reference x to an existing record in D, remove the record from D.
- LargestBefore(D, d): Return the amount of the largest settlement over all records dated on date d or earlier. If there is no record in D dated d or earlier, return 0.
- LargestAfter(D, d): Return the amount of the largest settlement over all records dated on date d or later. If there is no record in D dated d or later, return 0.
- NumBetween(D, d_1, d_2): Return the number of settlements reached between dates d_1 and d_2 inclusive. Assume that $d_1 \leq d_2$.

Note that the dates d may include times (and thus may have a large range represented) and the settlements are integers (which may come from a very large range, say 64-bit integers), thus a direct access table for these fields is not appropriate.

a. (6 marks) Give a precise and full description of a data structure that implements this ADT. Your data structure must be based on an AVL tree where each node in the tree represents one settlement record. Illustrate your data structure by giving an example of it on some collection of operations of your choice.

b. (18 marks) For each of the above operations, describe how to implement it in $O(\log n)$ worst-case running time and explain why, in each case, your algorithm achieves this time complexity. You do not need to describe any operations or repeat any complexity analysis that were given in class or in the textbook: simply refer to these as needed (and concentrate on any modifications you require).

c. (4 marks) Describe how your would change your data structure if multiple records with the same settlement date were allowed. You don't need to make these changes to your data structure, but it should be clear from your description how one would suitably modify your data structure. Could you maintain the $O(\log n)$ worst-case running time performance of these operations?

Question 3. (7 marks) A large university want to determine the median GPA of its students. The GPA is recorded as a number between 0.00 and 4.00 (recorded to two decimal places). Using a data structure studied in class, design an algorithm to find the median GPA of n students in $\Theta(n)$ time and constant space (you may assume that the number n can be stored using constant space). Briefly justify your algorithm's correctness and complexity.

(Recall that the median in a list is the middle number if the list was ordered. For example, the median of 5,17,12,5,8,4,8 is 8. If there is an even number of elements, either middle element is acceptable.)

Question 4. (25 marks) This question is a programming assignment. To see its description follow the link given in the "Assignments" section of the course web page.