Computer Science 263 June 7, 2007
St. George Campus University of Toronto

Homework Assignment #2

Due: June 19, 2007, by 6:10 pm

1. *Please complete and attach (with a staple) an assignment cover page to the front of your assignment. You may work alone or with one other student. If you work in a group, write both your names on the cover sheet and submit only one copy of your homework.*
2. *If you do not know the answer to a question, and you write "I (We) 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 the "AVL tree notes" handout posted on the course web site.

a. (5 marks) Insert into an initially empty AVL tree each of the following keys, in the order in which they appear in the sequence: 7, 14, 8, 5, 4, 10, 15, 9, 13, 12, 2, 11.

Show the resulting AVL tree T, including the key and balance factor of each node. (Only the final tree should be shown; any intermediate trees shown will be disregarded, and not given partial credit.)

b. (5 marks) From AVL tree T you built above, delete node 5, and show the resulting tree. Show the key and balance factor of each node.

Question 2. (30 marks) Suppose you have been contracted to design a line-based command line editor with the following user interface.

Assuming there are n lines in the file, the lines are numbered 1 through n. While editing a file, there is a concept of the "current" line. Here are the available commands while using this editor:

- r ⟨filename⟩: Read the lines of file ⟨filename⟩ into memory. All existing lines previously in memory are lost. The current line is set to line one. Generate an appropriate error message if an I/O error is encountered.
- g k: Make line k the current line and display it (i.e. go to line k). Generate an appropriate error message if k is out of range.
- d: Delete the current line and make the next line the current line. If the last line is deleted, then the new last line becomes the current line. Generate an appropriate error message if the file is empty.
- i ⟨line⟩: Insert ⟨line⟩ just before the current line and make the newly inserted line the current line.
- a ⟨line⟩: Append ⟨line⟩ just after the current line and make the newly inserted line the current line.
- n: Display the line number of the current line.

All commands except 'r' and 'n' must work in $O(\log n)$ time.

a. (5 marks) Design an ADT to support this editor. The objects represented by your ADT must be a sequence of lines. Your ADT should be minimal, in the sense that it should not have multiple operations performing similar tasks.

b. (5 marks) Give a precise and full description of a data structure that implements your ADT. Your data structure must be based on an AVL tree where each item in the tree represents one line in the file. Illustrate your data structure by giving an example of it on some collection of operations of your choice.

c. (20 marks) For each operation on your data structure, describe how it is implemented (in pseudo-code) and explain how this implementation permits the above editor commands to be implemented in $O(\log n)$ time. 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).

Question 3. (15 marks) Consider the problem of determining, given an integer d and a set D of n positive integers, whether it is possible to walk east for some distance $x \in D$ and then walk north for some (possibly different) distance $y \in D$ and end up distance d from where we started. (Recall that the distance from (0,0) to (x, y) is $\sqrt{x^2 + y^2}$.)

You must design an algorithm that solves this problem in **expected time** $O(n)$, under some assumptions. Do **not** assume that the set D contains integers that are small or lie in a small range.

a. (10 marks) Describe your algorithm clearly in English and give the corresponding pseudo-code. Briefly justify why your algorithm is correct.

b. (5 marks) Explain why your algorithm has the required time complexity. Explicitly state every assumption you need for your analysis.

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.