Computer Science 263 May 24, 2007
St. George Campus University of Toronto

Homework Assignment #1
Due: June 5, 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 (including the programming 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. *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. (7 marks) In class we studied *binary* heaps, i.e., heaps that store the elements in nearly-complete *binary* trees. This question is about *ternary* heaps, i.e., heaps that store the elements in nearly-complete *ternary* trees (where each node has at most *three* children, every level is full except for the bottom level, and all the nodes at the bottom level are as far to the left as possible). Here we focus on MAX heaps, where the priority of each node in the ternary tree is greater or equal to the priority of its children (if any).

a. (3 marks) Explain how to implement a ternary heap as an array A with an associated $Heapsize$ variable. Specifically, explain how to map each element of the tree into the array, and how to go from a node to its parent and to each of its children (if any).

b. (4 marks) Suppose that the ternary heap contains n elements.
(i) Precisely which elements of array A represent internal nodes of the tree?
(ii) What is the exact height of the tree? Do not use asymptotic notation.

Question 2. (20 marks) Suppose you are given k arrays of integers, each list sorted in nondecreasing order. We want to merge these k lists into a single sorted list.

Describe an algorithm that takes as input k and the k sorted lists, and constructs a new list containing all the elements of the the k input lists in nondecreasing order. Justify the correctness of your algorithm and prove that its worst-case running time is in $O(n \log k)$, where n is the total number of elements over all k lists.

Question 3. (13 marks) Suppose we have a max-binomial heap (*alias* max-binomial queue) containing n elements. Consider the following operation on this binomial heap:

- UPDATE(Q, x, key), where x is a pointer to an element in the binomial heap Q:
  Change the priority of the element pointed to by x to key and restore the heap ordering property. Note that the key's priority may increase or decrease (or even remain the same).

a. (5 marks) Describe an efficient algorithm for performing this operation. Justify its correctness.

b. (8 marks) Prove a tight asymptotic bound on your algorithm's worst-case running time (i.e., prove Θ).

Question 4. (20 marks) This question is a programming assignment. To see its description follow the link given in the "Assignments" section of the course web page.