Homework 3 – Due Wednesday, September 19, 2007

Please refer to the general information handout for the full homework policy and options.

Reminders

- Your solutions are due before the lecture. Late homework will not be accepted.
- Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.
- To facilitate grading, please write down your solution to each problem on a separate sheet of paper. Make sure to include all identifying information and your collaborators on each sheet. Your solutions to different problems will be graded separately, possibly by different people, and returned to you independently of each other.
- *For all problems where you are asked to design an algorithm, do not forget to prove correctness and analyze your algorithms time and space complexity.*

Exercises These should not be handed in, but the material they cover may appear on exams: problems in Chapter 4.

Problems to be handed in

1. (**Asymptotic Notation Properties**) Let $f(n)$ and $g(n)$ be asymptotically positive functions. Prove or disprove (by giving a counterexample) each of the following conjectures.
   (a) $f(n) = O(g(n))$ implies $g(n) = \Omega(f(n))$.
   (b) $f(n) = \Theta(f(n/2))$.
   (c) $f(n) = O(g(n))$ implies $\log(f(n)) = O(\log g(n))$, when $\log(g(n)) \geq 1$ and $f(n) \geq 1$ for all sufficiently large n.
2. (**Analysis of d-ary heaps**) A **d-ary heap** is like a binary heap, described in Chapter 2.5 of Kleinberg Tardos, with the exception that non-leaf nodes have d children instead of 2.
   (a) How would you represent a d-ary heap in an array?
   (b) Implement Parent(i) that, given the index i of a node, returns the index of its parent and Child(i, k) that, given the index i of a node, returns the index of its kth child.
   (c) What are the minimum and the maximum number of elements in a d-ary heap of height h?
   (d) Design an efficient implementation of Heapify-Up in a d-ary min-heap, analogous to the procedure on page 61 of KT. Analyze the running time of your algorithm in terms of d and n.

(e) Design an efficient implementation of HEAPIFY-DOWN in a d-ary min-heap, analogous to the procedure on page 63 of KT. Analyze the running time of your algorithm in terms of d and n.

(f) Suppose we implement a priority queue using a d-ary heap. Give the running times of all operations, described on pages 64–65 of KT, in terms of d and n.

3. (**Greedy Stays Ahead**) Chapter 4, problem 5. Do not forget to prove the correctness of your algorithm, and analyze its time and space complexity.

4. (**Greedy: A Structural Argument**) Chapter 4, problem 14. Do not forget to prove the correctness of your algorithm, and analyze its time and space complexity.

5. (**Greedy: An Exchange Argument**) Chapter 4, problem 13. Do not forget to prove the correctness of your algorithm, and analyze its time and space complexity.