CS 323.000: Data Structures and Algorithms Handout #2
Emory University, Math/CS Dept. January 25, 2012

Homework #1: Priority Queues
Due Date: written part: 5pm February 2 (Thursday)

This is just the *written* part of hw1, due in 8 days. The programming part of hw1 is not ready yet, it should probably be available Friday, and due somewhat later.

Try to solve these problems, at most one page each, and turn in your work on paper. If you cannot find me in class or my office, you may slide your work under my office door (W426).

Problem 1. (From CS171?) Suppose we have an input stream of N numbers, and after reading the stream, we want to report the M largest numbers. Also suppose M is relatively small compared to N: $M \leq N/\lg N$. For both parts, describe how to solve the problem using a binary heap. (For one of the parts, you'll also need the linear-time heap construction method of Section 2.4, from the discussion of heapsort.)

1(a). Solve the problem in $O(N)$ time, using $O(N)$ space.

1(b). Solve the problem in $O(N \lg M)$ time, using $O(M)$ space.

Remarks: the book solves one of these. In both parts, "space" counts words of memory. A word is large enough to store a single input number, or an integer at least as large as N. These two solutions constitute a "trade-off": we can solve the problem in less time, if we use more space.

Problem 2. (based on 1.5.15 in the book) Consider a tree of height n constructed by the weighted quick-union algorithm (without path compression). Argue that it has at least 2^n nodes. Furthermore, bound the number of its nodes at distance k from the root by the binomial coefficient $\binom{n}{k}$. Finally, compute the average depth (distance from the root) of a node in the worst case tree with exactly $N = 2^n$ nodes.

Honor Policy This assignment is governed by the Emory Honor Code, it should be your own work. The programming part is governed by the department's Statement of Policy on Computer Assignments.