CS 182
Advanced Topics in Algorithms
Spring 2002
Problem Set 2a
Due Thursday, January 30 in class

Please remember that all problem set submissions must be typeset, preferably in LaTeX. Please keep an electronic version of your solution set for your records.

1. **[25 Points] Minqueues.** In class we considered an extension of the queue abstract data type, called a *minqueue*, which supports operations ENQUEUE, DEQUEUE, and FINDMIN. We considered a clever data structure for this abstract data type which works as follows: There are two regular queues used. The first queue is called the "real queue" and the second queue is called the "helper queue". When the ENQUEUE(x) operation is performed, the element x is enqueued in the regular way on the real queue. The element x is also enqueued at the end of the helper queue. However, if the element y immediately "in front" of x on the helper queue is larger than x, then y is removed from the helper queue. This process of x annihilating the element immediately in front of it is repeated until the element immediately in front of x is less than or equal to it or x is at the front of the helper queue. To DEQUEUE an element, we remove it from the real queue and, if that element is also at the head of the helper queue, we remove it from there as well. Finally, FINDMIN simply returns the value at the head of the helper queue.

   (a) In class, we began proving the correctness of this data structure by proposing an invariant and showing that the FINDMIN and ENQUEUE operations preserve that invariant. Now complete the proof of correctness by proving that DEQUEUE also preserves the invariant. Write our your proof carefully and precisely. You may wish to break the analysis up into some cases.

   (b) In class we used the accounting method to prove that any sequence of n FINDMIN, ENQUEUE, and DEQUEUE operations take $O(n)$ time. Now, prove this again *twice*: once using an aggregation argument and once using the potential method.