CS 361A - Advanced Data Structures and Algorithms

Autumn Quarter, 2005

Homework #2 (Due: 11/7/05)

1. For the list update problem, we had mentioned two other heuristics besides move-to-front (MTF). The first was *transpose* (T) and the second was *frequency count* (FC).
   (a) We showed in class that for every algorithm A and each operation sequence S, the ratio of the cost of MTF on S to the cost of A on S is no more than 2. Prove the best lower bound that you can for this ratio.
   (b) For both T and FC, prove that there exist sequences of operations (starting with an empty list) on which its total cost is $\Omega(n)$ times the total cost of MTF, where n is the number of insert operations.
2. (a) Recall the definition of *demand paging*. Show that premature eviction of pages (i.e. when there is no page fault) can never help to reduce the number of page faults.
   (b) Recall the online paging algorithm LIFO (last in first out). Show that this algorithm is not c-competitive for any constant c. (The constant c can depend upon the size of the fast memories of LIFO or MIN, but not the length of the input sequence.) What can you say about the competitiveness of LFU (least frequently used)?
3. Prove the best upper bound that you can on the competitive ratio of the paging algorithm FIFO (first in first out).
4. (a) Consider the following special case of the disjoint-set data structure. The request sequence starts off by performing n MAKESET operations, and subsequently has an arbitrary intermix of $n-1$ UNIONs and $n \log n$ FINDs. Show that this can be implemented in total time $O(n \log n)$.
   (b) Suppose we implement the disjoint-set data structure described in class using only the union-by-rank heuristic (i.e. no path compressions). Provide a sequence of m operations including exactly n MAKESET operations such that the total time required is $\Omega(n \log n)$.
5. In the disjoint-set data structure, show that any sequence of m operations where all UNION operations precede all FIND operations will require total time $O(m)$, when both union-by-rank and path compression are used. What happens if only path compression is used?