COS 423 Class Notes
R. E. Tarjan

Amortized Analysis of Move-to-Front (MF) List Rearrangement

The setting:

A list with n items. The cost to access the k^{th} item from the front is k. Any two *adjacent* items may be swapped at a cost of 1.

Move-to-front (MF): After accessing an item, move it all the way to the front of the list, via successive swaps. The relative order of other items does not change. The total cost to access the k^{th} item and move it to the front is $2k-1$. This is an *on-line* algorithm: it uses no information about possible future accesses.

Question: for an arbitrary sequence of accesses, how well does move-to-front perform as compared to the optimum algorithm (OPT), which, knowing the entire access sequence in advance, chooses swaps so as to minimize the total cost of accesses and swaps? OPT is *off-line*: it has knowledge of the future, and is thus unrealizable in practice, except for access sequences completely known in advance.

We shall show that the total cost of MF to do all accesses is at most four times the total cost of OPT, for any access sequence. We assume that the initial list order is the same for both MF and OPT, and that only accesses are performed, no additions to or deletions from the list.

We use a potential function argument. We define the potential Φ as a measure of the difference between MF's current list order and OPT's current list order. Specifically, the potential is twice the number of *inversions* between the MF-list and the OPT-list. An inversion is a pair of list items x and y such that x and y are in different orders in the MF-list and the OPT-list: x precedes y in the MF-list and follows y in the OPT-list, or x follows y in the MF-list and precedes it in the OPT-list.

Exercise: Construct a pair of list orders that differ by exactly $n(n-1)/2$ inversions.

The initial potential is zero, since the initial lists are identical. The potential is always non-negative, since there must be a non-negative number of inversions (at most $n(n-1)/2$, one per unordered pair of list items). Thus the total amortized cost of MF over an entire access sequence is an upper bound on the total actual cost of MF.

We shall show that the total amortized cost of MF is at most four times the total actual cost of OPT, thereby giving the desired result. We show this bound for each operation, which implies that it is true for the entire sequence of operations.

Consider a single access operation, of item x. Let item x be k^{th} on the MF-list before the access and j^{th} on the OPT-list. The actual cost for the MF-access is $2k-1$, including the cost of the swaps that move x to the front of the MF-list. The actual cost for the OPT-access is j. (We account separately for any swaps that OPT does before or after accesses; see below.)

How does the potential change as a result of moving item x to the front of the MF-list? The only inversions that can be created or destroyed are those involving x as one item of the pair: the order of other pairs is unaffected. There are $k-1$ items preceding x in the MF-list before the move-to-front; each of these items follows x after the move-to-front. For each of these, either a new inversion is created or an old inversion is destroyed. The relative order of x and items initially following it is unaffected by the move-to-front, so we must only account for these $k-1$ inversion changes.

Let y be an item in front of x in the MF-list before the move-to-front. For a new inversion to be created, y must precede x in the OPT-list. There are only $j-1$ items in front of x in the OPT-list, so at most $j-1$ inversions can be created by the move-to-front. The remaining inversion changes, namely at least $k-1-(j-1)$, must be inversion destructions. Note that if $j > k$ this number is negative, but then it is trivially a lower bound on the number of inversion destructions.

We conclude that the potential change caused by the move-to-front is at most $2((j-1)-(k-1-(j-1))) = 4j-2k-2$. (The factor of 2 comes from the definition of the potential as *twice* the number of inversions.) The amortized cost of the access of item x is thus at most

$$2k-1 \quad \text{(the actual cost)}$$

$$+4j-2k-2 \quad \text{(the maximum potential increase, which may be negative)}$$

$$= 4j-3 < 4j.$$

That is, the amortized cost to MF of accessing x is at most four times the actual cost (j) that OPT pays for the access.

We must still account for any swaps that OPT does between accesses. We treat these swaps one-at-a-time.

The actual cost to MF of such a swap is zero (MF does nothing); the actual cost to OPT is one. Such a swap will create or destroy exactly one inversion, and thus will increase the potential by at most two. The amortized cost to MF of such a swap is thus at most two, which is less than four times the actual cost (one) that OPT pays for the swap.

Putting all the parts together gives the desired result: the total actual cost of MF is at most four times the total actual cost of OPT, on any access sequence.

Problem: By using a different algorithm, for example a randomized one, can you get a better competitive factor than 4 in our model? What is the best possible competitive factor?

Exercise: Suppose we change the model to allow the swapping of *any* two items, adjacent or not, at a cost of one. Show that OPT beats any on-line algorithm by a factor of at least $n/2$ on arbitrarily-long, suitably-chosen access sequences.