Online Access of Linked Lists

1 Online Access of Linked Lists (Sleator and Tarjan 1985)

1.1 Preliminaries

One important application of online algorithms is in self-adjusting data structures like splay trees and cache. Given a linked list of n items, we receive a sequence $\sigma(1), \ldots, \sigma(m)$ of accesses to items with m a priori unknown. If $\sigma(t)$ is in position i in the linked list, we pay i for finding it because we must linear search to it. Subsequently, we can move t forward in the list for free if we wish, and we can also swap any two adjacent elements (other than t) at a cost of one per swap. Our goal is to approximately minimize total cost.

Some reasonable heuristics include:

- FC (frequency count): Keep the elements sorted by the number of accesses so far.
- Static ordering: A possibly optimal static ordering with hindsight allowed.
- MTF (move to front): When t is accessed, move it all the way to the front.
- TRANS (transposition): Move t one element forward when accessed.
- Spectrum between MTF and TRANS (move t two elements forward, move t half the distance to the front, etc).

Of these heuristics, only one has nice guarantees.

1.2 Initial Bounds

For all algorithms on a sequence length k, we know that their cost must be $\leq kn$ – the upper bound occurs when the last element in the linked list appears in every request. Furthermore we know that the cost of OPT is $\geq k$, because it could possibly request the first element every time. So the upper bound on the competitive ratio is n, but we really want the algorithm to be constant competitive. We'll study several heuristics to see that they are n or $\frac{n}{2}$ competitive.

For FC, we could receive an input which accesses the first element t times, the second $t-1$, and so on, accessing the last element $t-(n-1)$ times. Then, the cost is $\geq \frac{n}{2} * n * (t-(n-1)) \approx \frac{n^2t-n^3}{2}$. In this case, OPT would have cost $n^2 - tn$. Thus the competitive ratio would be:

$$\lim_{t\to\infty} \frac{n^2t - n^3}{2(n^2 + tn)} = \frac{n}{2}$$

Static is exactly the same. For TRANS, we could receive an input which alternates accessing element n and element $n-1$. They would just infinitely switch places, so there would be a cost n for each iteration, while OPT would have a cost of 1 per iteration. Thus, we can have arbitrarily long sequences with ratio $\Theta(n)$.

By process of elimination, MTF must be the heuristic to give good guarantees. We'll prove that it is 2-competitive.

2 MTF Competitive Analysis

Theorem 1. *MTF is 2-competitive for any sequence σ. $\forall \sigma, c_{MTF}(\sigma) \leq 2 * c_{OPT}(\sigma)$.*

Proof. This proof would be easy if we could show that in each step i, $c_{MTF}(i) \leq 2 * c_{OPT}(i)$. However, this cannot work because it is possible for OPT to invest cost in early steps to make later items cheaper by moving certain items around. Thus, we at least will need amortized analysis.

Define a **potential function** $\Phi(i)$ that measures how similar the MTF list and OPT list are at any step i. Notice that $\Phi(i) = 0$ means that the lists are identical. Then, let $c'_{MTF}(i) = c_{MTF}(i) + \Phi(i+1) - \Phi(i)$. In other words, we get bonus points for making the MTF list more similar to the OPT list, and extra costs for making them different. Then, we have:

$$\sum_{i=1}^{t} c_{MTF}(i) = \sum_{i=1}^{t} (c'_{MTF}(i) + \Phi(i-1) - \Phi(i)))$$

$$= \Phi(0) - \Phi(t) + \sum_{i=1}^{t} c'_{MTF}(i)$$

$$\leq \sum_{i=1}^{t} c'_{MTF}(i)$$

Because of this inequality, it will be sufficient to show $c'_{MTF}(i) \leq 2 * c_{OPT}(i)$. Of course, we will first need to define $\Phi(i)$, since all the discussion so far has been abstract. Here, we have two natural choices:

1. Spearman's Footrule: The sum of distances between the positions in the two lists, for all elements.
2. Kendall's tau: The total number of inversions, i.e., the number of pairs of elements which are in one relative order in MTF's list, and in the other order in OPT's list.

In this case, it turns out that we should use Kendall's tau. So, let $\Phi(i)$ denote the Kendall's tau distance between the two lists after the i^{th} access.

We now want to analyze the i^{th} step, where element $\sigma(i)$ is accessed. Assume that it is located in position m in the MTF list, and in position k in the OPT list. Thus, the respective access costs are m and k. In 5 addition, MTF moves $\sigma(i)$ to the front, thus eliminating some inversions and creating others. Also, OPT may move $\sigma(i)$ forward, and perform some paid transpositions.

First, let us analyze the move that MTF makes. By moving $\sigma(i)$ to the front, MTF eliminates all inversions with elements which are behind position k in OPT's list, but before position m in MTF's list. Suppose that there are $v \geq 0$ of them. Then, the remaining $m - 1 - v$ elements preceding $\sigma(i)$ in MTF's list must also precede $\sigma(i)$ in OPT's list. In particular, this proves that $k \geq m - v$.

While the move to the front eliminates v inversions, it also may create some new ones. All of the $m - 1 - v$ elements that were previously before $\sigma(i)$ in both of the lists are now still before $\sigma(i)$ in OPT's list, but behind $\sigma(i)$ in MTF's list. Thus, the move has created $m - 1 - v$ new inversions. Thus, the net change in inversions is $m - 1 - 2v$, and the total amortized cost incurred by MTF, summing both the actual cost and the change in inversions, is $m + (m - 1 - 2v) = 2(m - v) - 1$.

In addition, suppose that OPT makes f free moves forward, and p paid adjacent transpositions. Each move forward actually eliminates one inversion for us. Each paid transposition can introduce at most one new inversion. Thus, the total amortized cost is $2(m - v) - f - 1 + p \leq 2(m - v + p) \leq 2(k + p)$. The last inequality holds because we proved above that $k \geq m - v$. And $2(k + p)$ is exactly twice the cost incurred by the optimum solution, so we proved that the inequality holds for each i. Thus, MTF is 2-competitive. □

Open question: Can we compute OPT in polynomial time?