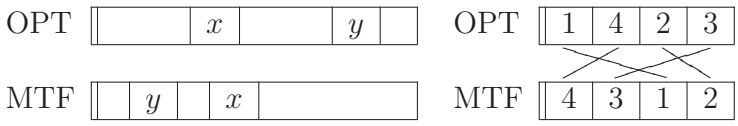

(a) A single inversion (b) Total inversion count

Figure 2: In the first figure, we illustrate an inversion of two elements y and x. In the second figure, we show how the elements are permuted between the two lists. If we count the number of intersections between the lines, we count the number of inversions. Hence, there are three inversions in the second figure. The non-inverted elements are $(1, 2)$, $(2, 4)$, and $(3, 4)$.

Theorem 2.1. *MTF is 2-competitive.*

Proof. With the definition of the MTF algorithm, and the previous lemma, we can show that MTF is 2-competitive by analyzing a list with ℓ element, over a request sequence σ without accesses to elements not in the list. Interestingly, we do *not* need to know the optimal offline algorithm, OPT. To prove this result we use a potential function argument. A potential function Φ is a function of some aspect of the problem such that at the start of the algorithm, $\Phi = 0$, and at the end of the algorithm $\Phi \geq 0$. If we have operations $1, \ldots, f$, then we can get an upper bound on the cost of the algorithm by examining the *amortized cost*

$$\hat{c}_i = c_i + \Phi_i - \Phi_{i-1} \qquad 1 \leq i \leq f,$$

where c_i is the true cost of operation i, and Φ_i is the potential after operation i. Notice that the total amortized cost is $\sum_i \hat{c}_i = \sum_i (c_i + \Phi_i - \Phi_{i-1}) = \sum_i c_i + \sum_i (\Phi_i - \Phi_{i-1}) = \sum_i c_i + \Phi_f$, where the last equality is because we had a telescoping sum. Because $\Phi_f \geq 0$ according to the definition of the potential function, we have that $\sum_i \hat{c}_i \geq \sum_i c_i$, or that the sum of all $\hat{c}_i$ upper bounds the actual cost of the algorithm.

In the following analysis, we will use the potential function

$$\Phi = \text{the number of } \textit{inversions} \text{ between } MTF \text{ and } OPT.$$

Formally, an *inversion* is an *ordered* pair of items (y, x) such that y precedes x in MTF's list and x precedes y in OPT's list. In Figure 2 we graphically illustrate an inversion and provide a helpful way to count the total number of inversions.

With the potential function Φ and the definition of inversion, we can now show that MTF is 2-competitive. Let $\sigma = \sigma_1 \sigma_2 \ldots \sigma_m$. On $\sigma_i, 1 \leq i \leq m$, MTF finds σ_i in the list and moves it to the front, whereas OPT makes a series of 0 or more paid transpositions, finds σ_i, and may make a free transposition. We will show that on each $\sigma_i, \hat{c}_i \leq 2\text{OPTCOST}_i$ where $\hat{c}_i$ is the amortized cost of MTF, and $OPTCOST_i$ is the cost for all of OPT's work.

Recall $\hat{c}_i = c_i + \Delta\Phi = c_i + \Delta\Phi_{\text{MTF}} + \Delta\Phi_{\text{OPT}}$, where c_i is the cost of MTF to find element σ_i, $\Delta\Phi_{\text{MTF}}$ is the change in potential due to MTF moving x to the front, and $\Delta\Phi_{\text{OPT}}$ is the change in potential due to moves by OPT. Let $\sigma_i = x$. If x is in position k, then $c_i = k$. Let v be the number of elements in front of x in MTF's list, but after x

in OPT's list. Since x is at position k, then there are $k-1-v$ other elements in front of x in MTF's list. Hence, $\Delta\Phi_{\text{MTF}} = -v + (k-1-v)$, because we remove inversions due to the v elements which are "realigned," but introduce inversions in the $k-1-v$ other elements. However, notice that if x is in position j in OPT's list, then $k-v \leq j$, because v was everything in front of x in MTF's list, but not in OPT's. So, what is left $(k-v)$ can be no larger than what is left in front of x in OPT's list.

Now, if OPT makes any paid transposition, those, at worst, increase the potential by 1 for each transposition. If OPT moves x to the front, then this just decreases the potential because we remove all inversions due to x. If

$$\text{OPTCOST}_i = \text{OPT\_FIND}_i + \text{OPT\_PAID}_i,$$

then because $k-v \leq j$, $\text{OPT\_FIND}_i \leq j$ — thus we have $k-v$ as a lower bound on the cost for OPT_FIND_i. Also, $\Delta\Phi_{\text{OPT}} \leq \text{OPT\_PAID}_i$.

At this point, we have all the pieces we need.

$$\begin{aligned}
\hat{c}_i &= c_i + \Delta\Phi_{\text{MTF}} + \Delta\Phi_{\text{OPT}} \\
&\leq k + (-v) + (k-1-v) + \text{OPT_PAID}_i \\
&= 2(k-v) - 1 + \text{OPT_PAID}_i \\
&\leq 2\text{OPT_FIND}_i + 2\text{OPT_PAID}_i \\
&= 2\text{OPTCOST}_i.
\end{aligned}$$

Thus,

$$MTF(\sigma) = \sum_i c_i \leq \sum_i \hat{c}_i \leq \sum_i 2\text{OPTCOST}_i = 2\sum_i \text{OPTCOST}_i = 2OPT(\sigma),$$

or MTF is 2-competitive. □

2.2 Online Deterministic List Access Algorithms are 2-competitive

Although we don't formally prove the lower bound for the deterministic List Access problem, we do sketch the result.

Claim. *For the list accessing problem with a list of ℓ items, any deterministic online algorithm has a competitive ratio of at least $2 - \frac{2}{\ell+1}$.*

The proof proceeds using an adversary that requests the last element in ALG's list at every step, so ALG costs ℓm on a request sequence with m elements. Then, it uses an averaging argument over offline algorithms that permute the list initially and then never reorder. With this setup, the cost of accessing any item in the list (over all $\ell!$ algorithms) is $\frac{\ell(\ell+1)}{2} \cdot (\ell-1)!$. Taking the average over all algorithms for m requests shows that an algorithm cannot be better than $2 - \frac{2}{\ell+1}$-competitive against the average of a set of offline algorithms, and cannot be any more competitive against a true optimal offline algorithm.