10 The List Access Problem

10.1 Inversion Distance

Let X be any set of cardinality n, and let $\mathcal{P}(X)$ be the set of all 1-1 and onto functions from X to $\{1, 2, \ldots n\}$. By a slight abuse of the term we refer to the elements of $\mathcal{P}(X)$ as *permutations* of X. (It would be more accurate to call them *orderings* of X.)

$\mathcal{P}(X)$ is a metric space under the *inversion metric*, defined as follows. Let $\alpha, \beta \in \mathcal{P}(X)$. If $x, y \in X$, $x \neq y$, then define

$$\tau(\alpha, \beta, x, y) = \begin{cases} 0 & \text{if } \alpha(x) < \alpha(y) \iff \beta(x) < \beta(y) \\ 1 & \text{otherwise} \end{cases}$$

The *inversion distance* from α to β is defined to be

$$||\alpha, \beta|| = \sum_{\substack{\{x,y\} \\ x \neq y}} \tau(\alpha, \beta, x, y)$$

Intuitively, we can think of α and β as ordered listings of X, and $||\alpha, \beta||$ is the number of transpositions of adjacent elements needed to change α to β.

As an example, suppose that $X = \{A, B, C, D, E\}$, and that $\alpha = (A, E, C, B, D)$ and $\beta = (B, A, D, C, E)$. We can represent the values of $\tau(\alpha, \beta, x, y)$ for all x, y in a matrix:

	A	B	C	D
E	0	1	1	1
D	0	0	1	
C	0	1		
B	1			

$||\alpha, \beta|| = 6$, the number of ones in the matrix.

10.2 The Simple Online List Access Model

We assume that we start with an empty list, and perform a sequence of operations. Each operation is either *insert* or *find*, or a paid exchange by the optimal offline algorithm. Costs are determined as follows.

1. The cost to insert an item x is equal to 1 plus the length of the list, and we place it at the end. We may, however, using free exchanges, move it forward to any position.

2. The cost to find an item x that is already in the list is ℓ, where ℓ is its position in the list. We may, using free exchanges, then move it to any position before ℓ.

3. At any time, we may exchange (transpose) any two adjacent items in the list, at a cost of 1. These are called paid exchanges.

10.3 Move to Front

The online algorithm Move to Front (MTF) always inserts a new element in the front of the list, at cost $\ell+1$ where ℓ is the length of the list. When MTF finds an item x, it pays ℓ to find x, where ℓ is the position of x in the list, and then moves x to the front of the list using free exchanges.

Theorem 10.1 *MTF is 2-competitive.*

Proof: Let $r^1, r^2, \ldots, r^n$ be a sequence of operations. Let L^i_{MTF} be MTF's list after the first i of those operatios, and let L^i_{OPT} be the optimal offline algorithms's list after the first i operations.

Define $\phi_i = \left|\left|L^i_{\text{MTF}}, L^i_{\text{OPT}}\right|\right|$, the potential after i operations, the inversion distance between the two lists. Let $\Delta\phi_i = \phi_i - \phi_{i-1}$, and let $\Delta cost^i_{\text{MTF}}$ and $\Delta cost^i_{\text{OPT}}$ be the cost of the i^{th} step of MTF, respectively, OPT. We will prove Inequality (1) below.

$$\Delta cost^i_{\text{MTF}} + \Delta\phi^i \quad \leq \quad 2\Delta cost^i_{\text{OPT}} \tag{1}$$

Case I: Find an item x.

We partitition the set of items other than x into six sets, as illustrated in Figure 10.1.

- A = the set of items that are in front of x before the move in MTF's cache, and which are in front of x before and after the move in OPT's cache.
- B = the set of items that are in front of x before the move in MTF's cache, in front of x before the move in OPT's cache, and behind x after the move in OPT's cache.
- C = the set of items that are in front of x before the move in MTF's cache, and which are behind x before and after the move in OPT's cache.
- D = the set of items that are behind x before the move in MTF's cache, and which are in front of x before and after the move in OPT's cache.
- E = the set of items that are behind x before the move in MTF's cache, in front of x before the move in OPT's cache, and behind x after the move in OPT's cache.
- F = the set of items that are behind x before the move in MTF's cache, and which are behind x before and after the move in OPT's cache.

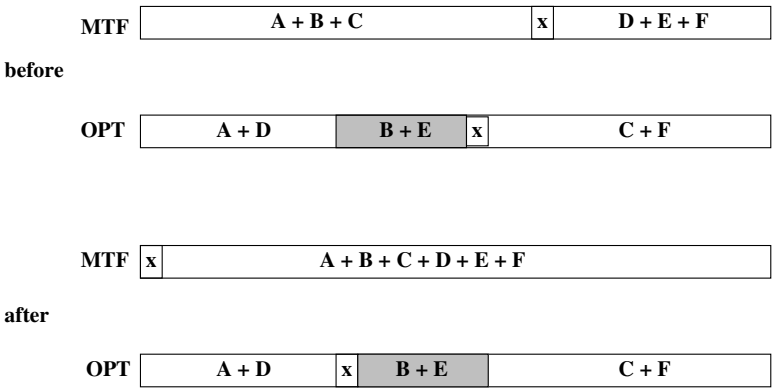


Figure 10.1: Illustation of Case I of the proof of Theorem 10.1

We thus have:

- $\Delta cost_{\text{MTF}} = |A| + |B| + |C| + 1$.
- $\Delta cost_{\text{OPT}} = |A| + |D| + |B| + |E| + 1$.
- $\Phi = |C| + |D| + |E|$ before the move.
- $\Phi = |A| + |D|$ after the move.
- $\Delta\Phi = |A| - |C| - |E|$.

Trivially, (1) holds.

Case II: Insert an item x.

Inserting x is simply a combination of the following two steps:

1. Place x at the end of the list, and pay nothing.
2. Find x.

The first step does not change the potential, since $\{x, y\}$ cannot be an inversion for any y. All costs for both MTF and OPT are in the second step, and those costs are exactly the same as in Case I, so we are done.

Case III: Paid exchange by OPT. Without loss of generality, the operation is the exchange of two adjacent elements, x and y, in L_{OPT}. Without loss of generality, x is in front of y in L_{OPT} before the exchange. If x is in front of y in L_{MTF} before the exchange, then $\Delta\phi = 1$; otherwise, $\Delta\phi = -1$. In either case, (1) holds.

□