

# Lecture 05 Amortized Analysis II: Competitive Analysis

LO5.1  
6.046  
2/20/2020

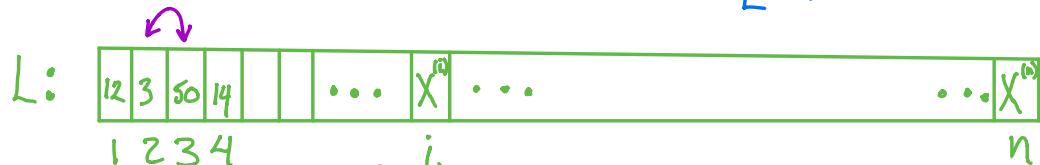
Reading: CLRS Problem 17-5

Usually we try to prove something about the absolute running time of an algorithm. Sometimes it is convenient instead to prove something about the running time relative to that of another algorithm. This is especially helpful if the comparison is to the "best possible" algorithm. This is the essence of competitive analysis.

## The Problem: Self-Organizing lists

List  $L$  contains  $n$  elements

- Only one operation:  $\text{ACCESS}(X)$ 
  - Find element with key  $X$  ↪ We are given the value of the key, not a pointer to its location.
  - Which costs  $\text{rank}_L(X)$



- Access(14) ↪ returns the fourth element
- Transposing 3 & 50 costs "1"

- After every Access we can re-order the list
  - transpose of adjacent elements costs "1"
  - multiple times for larger rearrangements

**GOAL:** Choose a sequence of transpositions that optimizes performance of **ACCESS** by minimizing cost  $C_A(S)$  VS, in on-line manner.

L05.2

**On-line algorithm:** Must respond to sequence of inputs "immediately" as each is presented, before seeing more of input sequence (e.g. game of Tetris).

**Off-line algorithm:** Can see whole sequence of inputs in advance and make possibly better choices (imagine playing Tetris this way!).

---



---

**Worst-case analysis:** The adversary always chooses the key of the last element ( $x^{(b)}$ ) of the list  $L$ .

$$C_A(S) = \Omega(|S| \cdot n)$$

← worst-case, any alg.  
 does poorly, even  
 ignoring cost of any  
 re-ordering

**Average-case analysis:** Suppose key  $x$  is accessed with probability  $p(x)$ . Then

$$E[C_A(S)] = |S| \sum_{x \in L} p(x) \cdot \text{rank}_L(x)$$

which is  
 minimized  
 when  $L$  is  
 sorted in decreasing  
 order of  $p(x)$ .

expectation value of the  
 cost of input sequence  $S$

INTUITION: Make the most frequent/likely inputs less expensive, while accepting the consequence that other, less frequent/likely inputs will be more expensive. It's still a win.

LO5.3

Heuristic: Keep a count of number of times each element is accessed, and adjust  $L$  in decreasing order of count. ← Adversary could still produce poor worst-case performance

Practical: Empirically it is found that the "move-to-front" (MTF) heuristic yields good results.

- After accessing  $x$ , move  $x$  to head of list  $L$ 
  - Cost of  $\text{rank}_L(x)$  to access
  - Cost of  $\text{rank}_L(x)-1$  to perform transpositions
  - Total cost =  $2 \cdot \text{rank}_L(x)-1$

To analyze performance will use competitive analysis.

## COMPETITIVE ANALYSIS (Sleator & Tarjan, 1985)

Definition: An on-line algorithm is  $\lambda$ -competitive if there exists a constant  $k$  such that for any sequence  $S$  of operations

$$C_A(S) \leq \lambda \cdot C_{\text{OPT}}(S) + k,$$

where  $\text{OPT}$  is the optimal off-line algorithm.

Theorem: MTF is 4-competitive for self-organizing lists.

L05.4

Proof: let  $l_i$  be MTF list after  $i^{\text{th}}$  access, and let  $l_i^*$  be OPT list after  $i^{\text{th}}$  access.

$$\begin{aligned} \text{let } c_i &= \text{MTF cost for } i^{\text{th}} \text{ operation} \\ &= 2 \cdot \text{rank}_{l_i}(x_i) - 1 \end{aligned}$$

$$\begin{aligned} \text{let } c_i^* &= \text{OPT cost for } i^{\text{th}} \text{ operation} \\ &= \text{rank}_{l_i^*}(x_i) + t_i \leftarrow \begin{array}{l} \# \text{ transpositions in} \\ \text{OPT algorithm} \end{array} \end{aligned}$$

Amortized Analysis: Potential Function

$$\begin{aligned} \text{Let } \phi(l_i) &= 2 \cdot (\#\text{inversions between } l_i \text{ & } l_i^*) \\ &= 2 / \{(x, y) : x \ll_{l_i} y \text{ and } y \ll_{l_i^*} x\} \end{aligned}$$



Example:  $l_i: E C A \xrightarrow{D} B$

$l_i^*: C A B D E$

Check all pairs  
in  $l_i$  for inversions  
in  $l_i^*$

$(E, C)$ , $(E, A)$ , $(E, D)$ , $(E, B)$ ,	$\checkmark = \text{inversion}$
$(C, A)$ , $(C, D)$ , $(C, B)$ ,	$\times = \text{not}$
$(A, D)$ , $(A, B)$ , $(D, B)$ ,	
	5 inversions

It takes 5 transpositions to turn  $l_i$  into  $l_i^*$

$\Rightarrow \phi = 10$

Properties:  $\phi(L_i) = 0$  if MTF & OPT start with same list order L05.5

$\phi(L_i) \geq 0$  always, because smallest number of inversions is 0.

A transpose creates or destroys 1 inversion,  
so  $\Delta\phi = \pm 2$

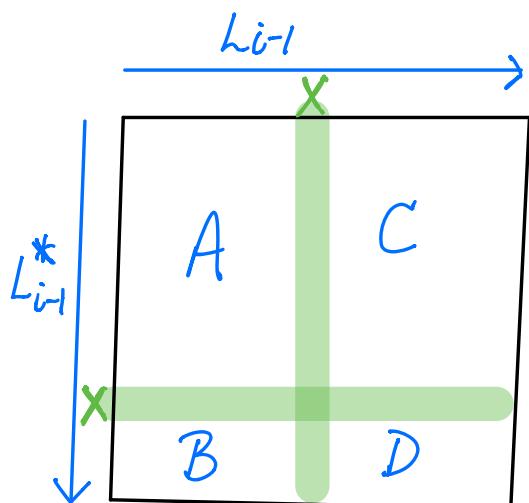
Once we access  $X$  in both  $L_{i-1} \notin L_{i-1}^*$ , all other elements fall into one of 4 categories.

A: elements before  $X$  in  $L_{i-1} \notin L_{i-1}^*$

B: elements before  $X$  in  $L_{i-1}$  but after  $X$  in  $L_{i-1}^*$

C: elements after  $X$  in  $L_{i-1}$  but before  $X$  in  $L_{i-1}^*$

D: elements after  $X$  in  $L_{i-1} \notin L_{i-1}^*$



$$L_{i-1}: \boxed{A \cup B \quad X \quad C \cup D}$$

$$\leftarrow \text{rank}_{L_{i-1}}(x) = r$$

$$L_{i-1}^*: \boxed{A \cup C \quad X \quad B \cup D}$$

$$r^* = \text{rank}_{L_{i-1}^*}(x) \rightarrow$$

On access:  $r = |A| + |B| + 1$  ← when MTF moves  $x$  to front, creates  $|A|$  inversions and destroys  $|B|$  inversions

$r^* = |A| + |C| + 1$  ← each transpose by OPT creates

$$\phi(L_i) - \phi(L_{i-1}) \leq 2(|A| - |B| + t_i) \leq 1 \text{ inversion}$$

First examine per-access cost for  $i^{\text{th}}$  Access:

L05.6

$$\begin{aligned}
 \hat{C}_i &= C_i + \phi(L_i) - \phi(L_{i-1}) \\
 &\stackrel{\text{amortized cost}}{\approx} \stackrel{\text{actual cost}}{\approx} \\
 &\leq (2r-1) + 2(|A|-|B|+t_i) \\
 &= 2r-1 + 2(|A|-(r-1-|A|)+t_i) \\
 &= 2r-1 + 4|A|-2r+2+2t_i \\
 &= 4|A|+1+2t_i \\
 &\leq 4(r^*+t_i) \\
 &= 4C_i^*
 \end{aligned}$$

$r = |A|+|B|+1$

$r^* = |A|+|C|+1 \geq |A|+1$

Now examine total cost of  $|S|$  operations:

$$\begin{aligned}
 C_{\text{MTF}} &= \sum_{i=1}^{|S|} C_i = \sum_{i=1}^{|S|} (\hat{C}_i + \cancel{\phi(L_{i-1}) - \phi(L_i)}) \\
 &\leq \left( \sum_{i=1}^{|S|} 4C_i^* \right) + \cancel{\phi(L_0) - \phi(L_{|S|})} \quad \text{from telescoping sums} \\
 &\leq 4C_{\text{OPT}}
 \end{aligned}$$

NOTE:

- (1) We never found the optimal algorithm, yet successfully argued about how MTF competes with it.
- (2) Suppose the cost of transposition is doubled, from 1 to 2. Then how competitive is MTF?
- (3) If  $L_0 \neq L_0^*$ , then  $\phi(L_0)$  might be  $\Theta(n^2)$  worst case, but  $C_{\text{MTF}}(S) \leq 4 \cdot C_{\text{OPT}}(S) + \Theta(n^2)$  is still 4-competitive, because our competitiveness analysis treats  $n$  as a fixed constant and analyzes cost as  $|S| \rightarrow \infty$ .