**Today:** Online algorithms
- offline vs. online
- self-adjusting lists
- FC, Trans, MTF algorithms
- competitive analysis
- single vs. multiple fingers

*Offline algorithm:* given entire input up front
*before solving the problem - most of this class*

*Online algorithm:* given input one piece at a time
- output/decision must be made before next piece
- goal: online algorithm almost as good as
*competitive with* optimal offline algorithm
*on same input, for any input*

**Competitive ratio** $\alpha / \alpha$-competitive:
\[ \text{online-alg}(x) \leq \alpha \cdot \text{optimal offline alg}(x) \]
for all inputs $x$

- "regret ratio" compared to hindsight/omniscience
- like approximation algorithm/ratio, but
w.r.t. knowing future instead of NP-hardness
- weak competitiveness: allow $+\beta$ term
**Self-adjusting lists:**
- Maintain linked list of \( n \) elements
- Online access sequence \( x_1, x_2, \ldots, x_m \)
- Access \( (x_i) \) must scan list from front to node storing \( x_i \)
- Can also swap adjacent nodes (visited)

**Frequency Counting:**
- Count \( \# \) accesses \( f_i(a) \) to item \( a \) (by \( i \)th access)
- Keep list reverse-sorted by \( f_i(a) \)
- If we assume access sequence is chosen stochastically (a chosen indep. with prob. \( p(a) \)) then weakly 1-competitive [Bircher-SICOMP 1979]
- If we compare to optimal offline static list (i.e. reverse sorted by \( f_m(a) \)) then 2-competitive [Bentley & McGeach - CACM 1985]
- But not \( o(n) \)-competitive [Sleator & Tarjan - CACM 1985]

- Access (1) \( n \) times (position 1)
- Access (2) \( n \) times (position 2)

\[ \Rightarrow \text{cost} \sum_{i=1}^{n} n \cdot i = \Theta(n^3) \]
- OPT moves current item to front

\[ \Rightarrow \text{cost} \sum_{i=1}^{n} (i + n) = \Theta(n^2) \]

\[ \Rightarrow \text{ratio} = \Theta(n) \]
Transpose: move $x_i$ one position toward front
- stochastically, beats MTF \[\text{[Rivest - CACM 1976]}\]
- not $o(n)$-competitive even against static list \[\text{[Bentley & McGeoch - CACM 1985]}\]
- always access last item
  $\Rightarrow$ toggle last two items
  $\Rightarrow$ cost $\Theta(m \cdot n)$
- static OPT puts those items first
  $\Rightarrow$ cost $\Theta(m)$
  $\Rightarrow$ ratio $\Theta(n)$

Move To Front (MTF): move $x_i$ all the way front
- stochastically, \(\frac{1}{2}\)-competitive \[\text{[Chung, Hajela - SIGMOD 1981]}\] \[\text{[Seymour - JCSS 1988]}\]
- tight \[\text{[Gonnet, Munro, Suwanda - SIGCOMP 1981]}\]
- 2-competitive against static OPT \[\text{[Bentley & McGeoch - CACM 1985]}\]
- $O(1)$-competitive \[\text{[Sleator & Tarjan - CACM 1985]}\]
MTF is 4-competitive: [Sleator & Tarjan - CACM 1985]

- run MTF & OPT on same sequence, in parallel, starting from same initial list order
- potential function $\Phi$
  \[ = 2 \cdot \text{# inversions between MTF & OPT lists} \]
  \[ \text{pairs (a, b) where } (a_{\text{MTF}} b) \neq (a_{\text{OPT}} b) \]
- access($x_i$) in MTF:

\[
\begin{array}{cccccccc}
< & < & > & > & > & > & x_i & \overset{\text{MTF}}{=} \\
\end{array}
\]
- label nodes in front of $x_i$ in MTF list as $<$ (same) or $>$ (inverted) $x_i$ in OPT list
- $\Delta \Phi$ from MTF($x$) = $2(\#<'s - \#>')$
- actual MTF cost = $2(\#<'s + \#>') + 1$
- OPT access cost $\geq \#<'s + 1$
  \[ \Rightarrow \text{amortized MTF cost} = 4 \cdot \#<'s + 1 \]
  \[ < 4 \cdot \text{OPT access cost} \]

- OPT may also do swaps
  - each may increase $\Phi$ by 2 (inversion)
  \[ \Rightarrow \text{may increase amortized MTF cost by 2} \]
  - but OPT cost increases by 1
  \[ \Rightarrow \text{still MTF} \leq 4 \cdot \text{OPT} \]
Best known:

- MTF is $2$-competitive if moving $x_i$ toward front is considered free \[ \text{[Sleator & Tarjan - CACM 1985]} \]
- no deterministic algorithm is $<2$-competitive \[ \text{[Karp & Raghavan]} \]
- Bit algorithm is $1.75$-competitive in expectation (against oblivious adversary) \[ \text{[Reingold, Westbrook, Sleator - Algorithmica 1994]} \]
  - assign random bit to each element, once at beginning
  - upon access, flip bit, and MTF if 1.
- best algorithm is $1.6$-competitive in expectation \[ \text{[Albers, von Stengel, Werchner - IPL 1995]} \]
- no algorithm is $<1.5$-competitive

**OPEN:** close gap for randomized

- offline OPT is NP-hard \[ \text{[Ambuehl - ESA 2000]} \]

**OPEN:** approximability?
Order By Next Request (OBNR):  [Munro-ESA 2000]

- offline (omniscient) access algorithm
- access(x_i) finds x_i, say at position k
  - continue scan to next power of 2: 2^[lg k]
  - sort these nodes by next request time
    with x_i at front (say)
- INVARIANT: elements 2^b+1, 2^b+2, ..., 2^{b+1}-1 sorted
  \Rightarrow sorting = merging blocks of sizes 2^0, 2^1, ..., 2^[lg k]
- merge smaller into larger \Rightarrow O(k) time
- KEY: need 2 fingers + long-dist swap

Example:

<table>
<thead>
<tr>
<th>x_i</th>
<th>cost</th>
<th>list</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1 2 3 4 5 6 7 8 ...</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2 1 3 4 5 6 7 8 ...</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3 4 1 2 5 6 7 8 ...</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>4 3 1 2 5 6 7 8 ...</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>5 6 7 8 1 2 3 4 ...</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>6 5 7 8 1 2 3 4 ...</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>7 8 5 6 1 2 3 4 ...</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>8 7 5 6 1 2 3 4 9 ...</td>
</tr>
</tbody>
</table>

... etc.

Total cost per permutation \sim n \cdot (\frac{1}{2} \cdot 2 + \frac{1}{4} \cdot 4 + ...) = n \cdot \lg n

MTF costs \sim n^{3/2} \Rightarrow ratio \sim \frac{n}{\lg n}

Ditto for any online algorithm on some permutation
**OBUR analysis:** amortized cost \( c(x_i) \leq 4 \lceil \lg w(x) \rceil + 1 \)

where \( w(x) = \# \) distinct \( y \)'s accessed since last access(\( x \)) including \( x \) itself — **working-set bound**

**Proof:** charge cost \( c = 2^{\lceil \lg k \rceil} \) of access(\( x_i \)) to elements in penultimate block, of size \( (c+1)/4 \)

(special case: \( i=1 \) \( \Rightarrow \) just pay cost of 1)

- if element \( x \) in block \( b \) gets charged
  - then either \( x \) advances to block \( b+1 \)
  - or it is followed by \( 2^{b+1} \) elements
  \( \Rightarrow \) not charged in block \( b \) before next access(\( x \))

\( \Rightarrow x \) charged at most once in each block
- \( x \) advances to block \( \lceil \lg w(x) \rceil \) before access(\( x \))
\( \Rightarrow \) amortized cost \( 4 \lceil \lg w(x) \rceil \).

- improve 4 to 2.6641... by changing 2 \( \rightarrow 4.24429... \) [Munro]
- lower bound of \( \Omega(\lg w(x) + 1) \) [Demaine & Harmon 2006]

**OPEN:** nail/improve constant