6.897
L6.01
~~Feb. 26~~, 2003
Mar. 3

What if OPT can change the list too?

- static optimality is "cheating":
  algorithm can change list dynamically
  but comparing to OPT that can't!
  (in stochastic model, this didn't matter)

- dynamic optimality / c-competitive:
  $\forall$ request sequence: cost(algorithm) $\leq c \cdot$ cost(best algorithm for that request seq.)
  i.e. cost($\underbrace{\text{(online) algorithm}}_{\text{only learns request sequence as the searches happen}}$) $\leq c \cdot$ cost($\underbrace{\text{(offline) OPT}}_{\text{knows request sequence ahead of time}}$)
  $\rightarrow$ omniscient

- Sleator & Tarjan cost model:
  - pay i to find element in position i
  - free to move found element anywhere closer to front
  - pay 1 per swap of 2 adjacent elements
    $\hookrightarrow$ a bit pricey... though no algs. so far do any such swaps

---

- FC is not $O(1)$-competitive
- * MTF is 2-competitive

} [Sleator & Tarjan — CACM 1985]

- No deterministic algorithm is <2-competitive [Karp & Raghavan]

- Bit algorithm:
  - initially assign random bit to each element
  - upon access, flip the bit, & if 1, MTF.
- Bit is 1.75-competitive against oblivious adversary [Reingold, Westbrook, Sleator — Algorithmica 1994]
- Best alg. so far is 1.6-competitive [Albers, von Stengel, Werchner — IPL 1995]
- No alg. is <1.5-competitive [Teia — IPL 1993]

OPEN: close randomized gap between 1.5 & 1.6

See also Chapter 2 of [Borodin, El-Yaniv — C.U.P. 1998] for related open problems