Self-adjusting data structures (new topic)

6.897
L5.2
Feb. 26, 2003

Linear search (comparison model)

Searching through a linked list of n elements costs
- n in worst case
- $n/2$ for average search if uniformly distributed.

What if queries are not uniformly distributed?

Stochastic model: $\Pr[\text{search for element } i] = p_i$
and these events are independent.
- Best list ordering is decreasing probability:
  1 2 3 ... n if $p_1 \geq p_2 \geq p_3 \geq \cdots \geq p_n$.
- Expected cost per query $= \sum_{i=1}^{n} i \cdot p_i = \text{"OPT"}$

What if we don't know p_i's? (but they exist)

① Frequency Counting (FC)
- store # times each element requested
- order elements in list by decreasing counts

② Move To Front (MTF)
- upon request, move element to front of list

③ Transpose
- upon request, move element 1 step closer to front

Idea: self-adjusting data structure adapts to request sequence on-the-fly to make it fast.

Over sufficiently long request sequences:

- cost(FC) ~ OPT [Bitner – SICOMP 1979]
- cost(Transpose) ≤ cost(MTF)
- cost(Transpose) < cost(MTF) unless $n \leq 2$ or all p_i's equal
  } [Rivest – CACM 1976] ← (& others)
- cost(MTF) < 2 · OPT [Chung, Hajela, Seymour – JCSS 1988]
- cost(MTF) ≤ $\pi/2$ · OPT [Gonnet, Munro, Suwanda – SICOMP 1981]
- cost(MTF) = $\pi/2$ · OPT for a particular distrib.