Randomized Online Algorithms

Theorem: If A is a deterministic, c-competitive online algorithm then $c \geq k$ (k= # pages in cache).

Proof: Idea: Find a sequence that is bad for A but not for OPT.

Assume A and OPT have same pages in cache 1,2,3,...,k.

Request page k+1 = a1. A faults and evicts some page, a2.

Request page a2. A faults and evicts some page a3

Request page a3. Evicts page a4, etc

A faults on every page request.

Total number different pages is k+1

How many times does OPT fault?

OPT $a_1*, a_2, a_3, \ldots, a_j, a_{j+1}*$ (* fault)

Claim: $j \geq k+1$

Proof: OPT evicts the page that is requested furthest in the future. Since there are only k+1 different pages, the next k page requests can be kept in cache.

Theorem: LRU is k-competitive

Proof: Let $p_1, p_2, p_3, \ldots, p_n$ be any sequence of page requests. Partition this sequence into contiguous sequences (phases) such that LRU faults on first page of the phase and the phase contains exactly k different pages.

LRU faults ast most k times per phase (only on 1st occurence of a page in phase)

OPT must have the first page of a phase in cache at beginning of phase. Since the remainder of the phase plus the first page of the next phase consists of k different pages, OPT must fault at least once during these requests.

$f_{LRU}(p_1, p_2, \ldots, p_N) \leq k\#phases$

$f_{OPT}(p_1, p_2, \ldots, p_N) \geq k\#phases - 1$

Thus, LRU is k-competitive.

Randomized Marking Mouse (RMM)

Mouse hides in one of m hiding places. Every time step, Cat looks in one of the m places. If mouse is there, Mouse must move to a different hiding place.

Cost = # times mouse moves

OPT = min # times mouse must move if mouse knows where cat will look.

For a deterministic mouse there is a sequence of cat probes s.

Mouse Cost(s) $\geq$ (m-1) OPT mouse cost (s)