In-Class Worksheet

CS 181 Advanced Algorithms — Fall 2025

Plenty of Practice with Paging Problems

1. Explain why the LIFO (Last-In-First-Out) paging policy is not α-competitive for any constant $\alpha > 1$.

2. Prove that the FIFO (First-In-First-Out) algorithm is k-competitive for the paging problem, where k is the cache size. Also prove that Flush-when-Full is k-competitive. Since these all have the same competitive ratio, what are some other pros and cons of these caching policies?

   Hint: Break the input sequence into phases. How many faults does FIFO incur on each phase?

3. **Belady's Anomaly**. It turns out that for some algorithms, increasing the cache size can actually lead to an increased number of faults! Run the FIFO policy on the following sequence starting with an empty cache of size of $k = 4$, and then when $k = 3$. How many faults are incurred?

   $$\sigma = 4, 3, 2, 1, 4, 3, 5, 4, 3, 2, 1, 5$$

   What do you think is the intuitive reason that this can happen in the FIFO policy?

4. Show that LRU (Least-Recently-Used) does not suffer from Belady's anomaly. That is, for LRU, the number of faults is nonincreasing in the cache size.

   Hint: A critical property of LRU is that at any time, the cache of size k contains exactly the most recently used k distinct pages. Use this to show that the cache contents $C_k(t)$ at time t with capacity k satisfies the relation $C_k(t) \subseteq C_{k+1}(t)$.

5. **Resource augmentation**. Consider an online paging setting in which an LRU algorithm has a cache of size k, while the optimal offline algorithm OPT is restricted to a cache of size $h \leq k$. Show that the competitive ratio of LRU in this setting is at most

$$\frac{k}{k-h+1}.$$

6. **Lookahead**. Let an online algorithm for the paging problem have a weak lookahead of ℓ if it knows the next ℓ requests at each time t. A lookahead is called strong if the algorithm at time t knows the next ℓ distinct requests to be made after time t, other than the request made at time t. For example, if the request sequence is $\sigma = \{\sigma_1, \sigma_2, \ldots\}$, then with weak lookahead the information at time t is $\{\sigma_1, \sigma_2, \ldots, \sigma_{t+\ell}\}$, while with strong lookahead it is $\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$, where $|\{\sigma_t, \sigma_{t+1}, \sigma_{t+2}, \ldots, \sigma_m\}| = \ell + 1$.

   (a) Show that with weak lookahead, no deterministic online algorithm can have a competitive ratio better than k in the worst case.

   (b) Show that a natural extension of LRU can achieve a competitive ratio of $k - \ell$ with strong lookahead.

7. **Cat and Mouse**. Consider the following cat and mouse game on a complete graph over n vertices. At time 0, the cat and mouse are at two distinct vertices. At any time, the cat does not know the vertex at which mouse is located while the mouse knows the vertex where cat is located. Moreover the mouse also gets to observe if the cat is moving towards its current location. What this means is that if the cat chooses an edge between its current vertex and the vertex at which currently the mouse is, the mouse gets to see that and can jump to any other vertex before the cat reaches its chosen vertex.

   In particular, at each discrete time t, $1 \leq t \leq T$, the cat chooses a vertex, say c_t, and the mouse sees the cat's choice and chooses a vertex $m_t \neq c_t$ to avoid the cat. Both move to their destination simultaneously. If $m_t \neq m_{t-1}$, the mouse incurs a cost of 1. The objective of the mouse is to stay alive (never end up at the same vertex as the cat), while incurring minimum total movement cost. Consider the moves made by the mouse to be an algorithm and treat the cat moves as an adversary.

   (a) Show that no deterministic algorithm can have competitive ratio less than $n - 1$.

   (b) Propose a deterministic online algorithm with competitive ratio of at most $n - 1$.