Assignment 3.

Post date: Oct. 2, 2014. Corrections (red fonts) on Oct. 9, 2014.
Due date: Oct. 14, 2014 (Tuesday) by noon. Hand in printed paper at assignment box at 4th floor of MC across from the tutorial centre.

Meeting submission requirement: (5 marks)

The answers need to be typed up with a computer and printed on letter size paper. Only minor corrections can be made with handwriting. The first page of the submission must be a cover page that contains your name, student id, and assignment number.

Question 1. (10 marks)

For the paging problem, design a page access sequence that makes LRU algorithm perform badly. More specifically, suppose both the LRU and MIN algorithms have a cache size of 3 pages. They also share the same initial cache state of $\{a, b, c\}$. To make things simpler, we further assume that in the initial cache the most recent access of the three elements are in the order of "a before b before c". Design an access sequence of length 6 such that LRU produces 3 times as many page faults as the MIN algorithm. Your answers should clearly show

(1) The page access sequence
(2) For each of the two algorithms, the cache state, page accessed and page removed (in case of a page fault) at each time.

Question 2. (20 marks)

In class, we discussed the weighted majority algorithm for combining many experts' opinion (see course notes). The goal was to minimize the number of mistakes. Now consider a variation of the problem.

Now the n experts' opinions are not anymore binary. For example, it can be an advice about which cloth to wear to get to school in early morning. (By the way, the experts can be your room mates, parents, etc.) At end of the day, you can evaluate the loss (too much for a warm day or too little for a cold day). The loss is always a real number between 0 and 1. Now you need an online algorithm to choose which expert to follow at each day. So that after m days, the algorithm's total loss is upper bounded by $c_1 \cdot OPT + c_2 \cdot \log_2 n$, for some constants c_1 and c_2. Here OPT is the total loss if you've only listened to the best expert.

Explain your modification to the algorithm (either a pseudo code or a clear description of the necessary modification is acceptable). Also, prove that your algorithm indeed achieves the desired performance.

Hint: Start with the randomized weighted majority algorithm. At each time, instead of reducing an expert's weight from w to $(1 - \epsilon) \cdot w$, reduce another appropriate amount according to the loss.