Department of Computer Science and Information Systems
CSIS 6924 (Algorithms),
CSIS 0324 (Topics in Theoretical Computer Science)
Assignment 3

Deadline: December 9, 1:45pm.
We accept both printed and written answers.
Remember to put your name and university number in the answer script.
Submit it to assignment box A6 (cargo lift lobby of 3/F, Chow Yei Ching building).

Answer all questions. They carry the same marks.

1. A cache-replacement algorithm is said to be a marking algorithm if:
   - At any time, each page is either said to be marked or unmarked. At the beginning, all pages are unmarked.
   - Whenever a memory page is requested, if the page is not in cache and all pages are marked, then all pages are immediately unmarked.
   - Whenever a memory page is requested, and no marked page in the cache have the requested page, the algorithm may choose a page from those that are not marked, and load the requested memory page in that cache page. (Of course, this must be done if no unmarked page in the cache have the requested page as well.)
   - Once the page is in cache (newly loaded or not), the requested page is marked.
   - At no other circumstances is cache page marked, unmarked, loaded or replaced.

   (a) Show that LRU is a marking algorithms.

   (b) By extending the analysis of FWF, show that all Marking algorithms are k-competitive. (Thus LRU is a k-competitive algorithm.)

2. Show that the following algorithm, called TRANSPOSE, for list access problem is no better than k-competitive: after the algorithm searches for an element x, exchange it with the element immediately before it.

3. Consider the list access problem again. Assume now that the off-line algorithm is allowed to exchange two adjacent elements at any time, at a cost of 1. (This is usually called paid exchange.) Modify the analysis of MTF to show that MTF is still 2-competitive. (Hint: how will a paid exchange change the potential?)

Additional questions for CSIS 0324 students

1. Extend the analysis of the FWF algorithm so that it compares the performance of FWF using a cache size of k against an optimal off-line algorithm with a cache size of h ($1 \leq h \leq k$). You may assume that initially, the cache is empty for both FWF and the optimal off-line algorithm. Show that FWF is $k/(k+1-h)$-competitive against such a weaker adversay.