Assignment 1

CS4SH3

Due on Feb 6 (in class)
Each question 10 marks, total 40 marks

1. In class we showed that the FIFO page-replacement algorithm, Section 10.4.2 in the textbook, incurs Belady's Anomaly. We used an example where the cache sizes were 3 and 4. Now, for a general i, provide a sequence of page requests that illustrates Belady's Anomaly incurred by FIFO on cache sizes i and $i+1$.

2. Recall the LRU page-replacement algorithm, presented in Section 10.4.4 in the textbook. Show that LRU does *not* incur Belady's Anomaly (on any cache size and any request sequence).

3. The Optimal Page Replacement algorithm (OPT), Section 10.4.3 in the textbook, evicts the page that will not be used for the longest period of time. As such, this algorithm is difficult to implement, because it requires future knowledge of the request sequence. However, it is very useful for comparison studies (i.e, competitive analysis).

   Show that OPT is indeed the optimal (off-line) page replacement algorithm. (Aside: OPT is often called LFD, which stands for "Longest Forward Distance", because it evicts the page whose next request is the latest.)

   Hint: You have to show that if ALG is any other page replacement algorithm, then on any request sequence $\sigma = p_1, p_2, p_3, \ldots, p_n$, OPT incurs at most as many page faults as ALG. The idea is to assume that ALG itself is optimal, and show that ALG can be modified easily to be just like OPT.

4. Define Flush When Full (FWF) to be a (very naive) page replacement algorithm that works as follows: whenever there is a page fault and there is no space left in the cache, evict all pages currently in the cache (call this action a "flush")[1].

   We say that a page replacement algorithm ALG is **conservative** if it satisfies the following condition: On any consecutive input subsequence containing k or fewer distinct page requests, ALG will incur k or fewer page faults.

   Prove that LRU and FIFO are conservative, but FWF is not.

[1] Clearly FWF is not a demand paging algorithm, because it evicts everyone when there is a page fault and the cache is full. However, FWF can be easily "upgraded" to be a demand paging algorithm as follows: instead of flushing, mark all the pages, and evict—when necessary—an arbitrary marked page.