6.854 Advanced Algorithms

Problem Set 10

John Wang

Collaborators:

Problem 1-a: Prove that LRU and FIFO are conservative.

Solution: First, we will consider LRU. Suppose that the LRU cache of size k has been filled up and is entering a new phase. Now consider a subsequence that contains at most k pages. Consider pages $p_1, p_2, \dots, p_k$ and LRU will check each whether p_i is already in the cache. If it is, then there is a cache hit. Otherwise, there is a fault. However, since there are at most k pages in the input sequence, there can be at most k faults. Therefore, we see that LRU is conservative.

Now, let us examine FIFO with the same analysis. Consider a subsequence of pages $p_1, p_2, \dots, p_k$. We know that there are exactly k pages in the input sequence. Since each page in the sequence can cause at most a single fault, there can be a maximum of k faults on FIFO. THerefore, we see that FIFO is conservative as well. $\square$

Problem 1-b: Prove that any conservative algorithm is k-competitive

Solution: Suppose that some algorithm A is conservative. We will break up the conservative algorithm's operation into phases of size k. Now suppose that the algorithm has just started phase i. In this phase, there will be k page requests. Since the algorithm is conservative, A, will have a maximum of k faults in this upcoming phase (which is really a subsequence of k pages). Now let us suppose OPT does not fault in phase i. Then it must fault on the $k+1$st request (first request in the next phase) since the cache is only of size k and it must have all k requests already stored in the cache. If OPT does fault during phase i, then it has at least one fault. Therefore, for each phase i, there will be at least one fault by OPT and at most k faults by A.

Since this is true for each phase i, we see that A is k-competitive, and hence, that any conservative algorithm is k-competitive. $\square$