Question 4, Assignment 1

For this exercise assume that the cache is of size k. Otherwise the claim is not true: for example, suppose that we have a cache of size 1, and the following sequences:

$$1, 2, 1, 2$$

Then, in that sequence of 4 requests there are only 2 distinct requests, yet with a cache of size 1, there would be 4 faults, for *any* page-replacement algorithm.

Question 3, Assignment 1

We will show that if ALG is any paging algorithm (on-line or off-line), then on any sequence of requests σ, ALG$(\sigma) \geq$ OPT(σ). As usual, ALG(σ) denotes the number of page faults of ALG on the sequence of requests σ. We assume throughout that all algorithms are working with a cache of a fixed size k.

We need to prove the following claim:

Claim: Let ALG be any paging algorithm. Let $\sigma = p_1, p_2, \ldots, p_n$ be any request sequence. Then, it is possible to construct an off-line algorithm ALG$_i$ that satisfies the following three properties:

1. ALG$_i$ processes the first $i - 1$ requests of σ exactly as ALG does,
2. if the i-th request results in a page fault, ALG$_i$ evicts from the cache the page with the "longest forward distance",
3. ALG$_i(\sigma) \leq$ ALG(σ)

First show how to use the **Claim** to prove that OPT is in fact the optimal algorithm. Let $\sigma = p_1, p_2, \ldots, p_n$ be once again any sequence of requests. By tha claim, we know that:

$$\text{ALG}_1(\sigma) \leq \text{ALG}(\sigma)$$

Applying the claim agin, we get:

$$\text{ALG}_2(\sigma) \leq \text{ALG}_1(\sigma)$$

YOU FINISH THIS!

Proof of Claim: Divide σ into three segments as follows:

$$\sigma = \sigma_1, p_i, \sigma_2$$

Since on σ_1, ALG$_i$ acts just like ALG, it follows that ALG$_i(\sigma_1) =$ ALG(σ_1), and also, they both do or don't incur a page fault on p_i.

1. They don't incur a fault.
2. They do incur a page fault, ALG$_i$ evicts the LFD page p from its cache, and replaces it with p_i.
   (a) ALG also evicts the same page (the LFD page). What should ALG$_i$ do on σ_2?
   (b) ALG evicts a different page, q.
      i. $p, q \notin \sigma_2$
      ii. $q \in \sigma_2$ but $p \notin \sigma_2$
      iii. $p, q \in \sigma_2$
      What should ALG$_i$ do on σ_2?

Any marking algorithm ALG is $\frac{k}{k-h+1}$-competitive
(where k is the size of its cache, and h is the size of the cache of OPT)

PROOF: Fix any request sequence σ and consider its k-phase partition. Assume, for now, that the last phase of σ is complete (in general, the last phase may be incomplete).

Claim 1: For any phase $i \geq 1$, a marking algorithm ALG incurs at most k page faults.

Proof of Claim 1: This follows because there are k distinct page references in each phase. Once a page is requested, it is marked and therefore cannot be evicted until the phase has been completed. Consequently, ALG cannot fault twice on the same page.

If we denote the i-th k-phase of σ by σ_i, we can express the above claim as ALG$(\sigma_i) \leq k$.

Thus, if there are s phases, ALG$(\sigma) \leq s \cdot k$.

Claim 2: OPT$(\sigma) \geq s \cdot (k-h+1)$, where again we assume that $\sigma = \sigma_1, \sigma_2, \ldots, \sigma_s$, where σ_s is complete.

Proof of Claim 2: Let p_a be the first request of phase i, and p_b the last request of phase i. Suppose first that phase $i+1$ exists (that is, i is *not* the last phase). Then:

$$\sigma = \ldots, p_{a-1}, \underbrace{p_a, p_{a+1}, \ldots, p_b}_{i}, \underbrace{p_{b+1}, \ldots, \ldots}_{i+1}$$

After processing request p_a, OPT has at most $h-1$ pages in its cache, not including p_a. From (and including) p_{a+1} until (and including) p_{b+1}, there are at least k distinct requests. So OPT must incur at least $k-(h-1) = k-h+1$ faults on this segment.

If i is the last phase (so $i = s$), we don't have p_{b+1}, so we can only say that we have at least $k-1$ faults, but we make it up with p_1 which has not been counted.

It follows from the above two claims, that:

$$\text{ALG}(\sigma) \leq s \cdot k \text{ and } \text{OPT}(\sigma) \geq s \cdot (k-h+1)$$

so that:

$$\frac{\text{ALG}(\sigma)}{s \cdot k} \leq 1 \leq \frac{\text{OPT}(\sigma)}{s \cdot (k-h+1)}$$

so finally:

$$\text{ALG}(\sigma) \leq \left(\frac{k}{k-h+1}\right) \cdot \text{OPT}(\sigma)$$

In the case that σ can be divided into s complete phases.

As was mentioned above, in general, the last phase may not be complete. Then, we repeat this analysis with $\sigma = \sigma_1, \sigma_2, \ldots, \sigma_{s-1}$, and for σ_s we use α at the end, so we get:

$$\text{ALG}(\sigma) \leq \left(\frac{k}{k-h+1}\right) \cdot \text{OPT}(\sigma) + \alpha$$

Exercise: work this out.

Therefore, in either case we obtain that any marking algorithm ALG is $\left(\frac{k}{k-h+1}\right)$-competitive.