DS 320 Algorithms for Data Science	**Lecture #7**
Spring 2022	Prof. Kira Goldner

Optimal Caching and Greedy Exchange

In the optimal caching problem, our computer has a main memory of size n, a cache of size k, and we are presented with a sequence of data $D = d_1, d_2, \ldots, d_m$ that we must process. When an item is not in the cache, we have a cache miss, and must bring the item into the cache and evict something else if the cache is full. Our goal is to give an algorithm that minimizes the number of misses.

Example 1:	a, b, c, b, c, a, b	$k = 2$	cache $= \{a, b\}$
Example 2:	$a, b, c, d, a, d, e, a, d, b, c$	$k = 3$	cache $= \{a, b, c\}$

A cache maintenance algorithm with an optimal greedy eviction schedule is the *Farthest-in-Future* Algorithm. When a new item needs to be brought into the cache, it greedily evicts the item that is needed the farthest into the future.

Goal: Prove that the Farthest-in-Future Algorithm is optimal. We'll call this schedule S_{FF}.

Definition 1. A schedule is *reduced* if it does the minimal amount of work necessary in a given step.

Lemma 1. *For every non-reduced schedule, there is an equally good reduced schedule (that brings in at most as many items as the original schedule).*

Prove this by construction.

Hint: You might *charge* a miss from one schedule to a miss in another schedule to show that it doesn't have any extra misses.

Observation 1. *For any reduced schedule, the number of items that are brought in is exactly the number of misses.*

Lemma 2. *Suppose S is a reduced schedule that makes the same eviction decisions as S_{FF} through the first j items in the sequence for some j. Then there exists a reduced schedule S' that makes the same eviction decisions as S_{FF} through the first $j+1$ items and incurs no more misses in total than S does in total.*

Prove this by constructing S'. *This is an **exchange** argument.*

a. What happens if the $j+1^{st}$ item is in cache?

b. What happens if the $j+1^{st}$ item isn't in cache, but S evicts the same item as S_{FF}?

c. What happens if the $j+1^{st}$ item isn't in cache, and S evicts a different item as S_{FF}? What should S' do?

d. How can you get S''s cache back to the same as S's without incurring more total misses?

e. How do we know that S' is a reduced schedule?

f. Sanity check: Are all parts of the lemma true?

Theorem 2. *S_{FF} incurs no more misses than any other schedule S^* and hence is optimal.*