Lower Bounding - Game Theoretic Approach

- Online Algorithms
- Online Paging Problem
- Competitive Analysis
  - Adversary Models
Online Algorithms

- **Online vs. Offline algorithms**
  - Offline algorithms receive all their inputs at one time.
  - Online algorithms receive and process their inputs in partial amounts

- **e.g. Sorting**
  - Quick Sort is an offline sorting algorithm while Insertion Sort is an online sorting algorithm

- **Why?**
Paging Problem

- Consider a 2 level memory hierarchy in a computer system
  - a fast – therefore expensive – therefore small – memory $M_0$
  - a slow – therefore inexpensive – therefore large – memory $M_1$

- Assume that each level is divided into units of exchange known as *pages*
  - Let $M_0$ have $k$ pages and $M_1$ have at least $k+1$ pages

- When a page is to be used by the processor, it is brought in from $M_1$ to $M_0$ if it is not already available
  - If no free slot is available in $M_0$ then one of the existing pages have to be *replaced*
Online Paging Problem

- The page to be replaced is decided by a *page replacement* algorithm.

- The replacement problem is an online problem because
  - the inputs i.e. the requested pages are not known beforehand

- Typical (online) algorithms are:
  - FIFO
    - replace the page that arrived the earliest
  - LFU
    - replace the page that has been used the least (since its arrival)
  - LRU
    - replace the page that has not been used for the longest time
Paging Algorithms

- Typical performance parameters for paging algorithms include:
  - Time complexity
    - time taken to make a decision
  - Space complexity
    - space used for meta data that is required for making a decision
  - Miss rate
    - the number of times a request for a page is not found in $M_0$ (i.e. is missed and therefore has to be brought in from $M_1$)

- We will analyze miss rates of paging algorithms
Paging Algorithms - Miss Rates

- Given a sequence of page requests $\rho = \rho_0, \rho_1, \ldots, \rho_n$ denote
  - the worst case number of misses by a specific paging algorithm $A$ as $f_A(\rho)$ and
  - the worst case number of misses by an optimal offline algorithm as $f_{OPT}(\rho)$

- The following is an optimal offline paging algorithm based on greedy choice
  - GreedyPaging:
    - Given an input sequence $\rho_0, \rho_1, \ldots, \rho_n$
    - On a miss replace the page whose next occurrence is farthest in the sequence
      - i.e. distance between the index of the current request and the index of occurrence of the page to be replaced is maximum
Paging Algorithm - Miss Rates

- **Assumptions:**
  - We will study the steady-state performance i.e. cold misses are not counted
  - Why is this a reasonable assumption?
  - We will assume that the size of $M_1$ is $k+1$ where $k \geq 2$ is the size of $M_0$
  - Why is this a reasonable assumption?

- **Greedy Paging Lemma:**
  - GreedyPaging is optimal and $f_{OPT}([\rho_0, \rho_1, ..., \rho_n]) = n / k$
  - Proof :
    - On a miss, the page to be replaced is the one that is farthest in the sequence (from the current request)
      - i.e. in the worst case at least $k$ requests can be handled before a replacement is required;
      - so, one of every $k$ requests will be a miss in the worst case.
Paging Algorithm - Miss Rates

- **Online Paging Lemma:**
  - For any deterministic online algorithm $A$ there exist sequences of arbitrary length such that $A$ misses on every request
  - i.e. $f_A([\rho_0, \rho_1, \ldots, \rho_n]) = n$
  - **Proof:**
    - Consider an adversary who chooses the next input $\rho_j$ to be a page that is not one of the $k$ pages in $M_0$
    - Since $|M_1| = k+1$, there always exists one such page.

- **Implication:**
  - Worst case analysis is not useful in comparing these algorithms
Paging Algorithm - Competitive Analysis

- Definition:
  - A deterministic online page replacement algorithm A is said to be C-competitive if there exists a constant b such that on every sequence of requests \( \rho = \rho_0, \rho_1, \ldots, \rho_n \):
    - \( f_A(\rho) - C \cdot f_{OPT}(\rho) \leq b \)
  - where the constant b must be independent of n but may depend on k.
  - The competitiveness coefficient of A, denoted \( C_A \), is the smallest C such that A is C-competitive.

- Online Paging Competitiveness Theorem:
  - For any deterministic online algorithm A for paging, \( C_A \geq k \)
  - Proof:
    - By GreedyPaging Lemma and Online Paging Lemma
Paging Algorithm - Competitive Analysis

Claim:
- $C_{LRU} = k$

Proof:
- Partition the input sequence into rounds $R_0, R_1 \ldots R_t$
  - such that each round $R_j$ results in exactly $k$ misses by LRU.

- In each round $R_j$, all the $k+1$ pages must have been accessed.
  - Why?

So, the ratio of misses by LRU to optimal misses is at most $k$ i.e. for any input sequence $\rho$
- $f_{LRU}(\rho) / f_{OPT}(\rho) \leq k$
- i.e. $C_{LRU} \leq k$

But by OPC Theorem: $C_{LRU} \geq k$. 
Paging Algorithm - Competitive Analysis

- **Claim:** \( C_{\text{FIFO}} = k \)
  - **Proof:** (similar to the proof for LRU: left as exercise)

- **Claim:** \( C_{\text{LFU}} > k \)
  - **Proof:**
    - Consider a sequence \( \rho \) where
      - \( \rho_0, \rho_1, \ldots, \rho_j \) are \( k-1 \) distinct pages with 2 accesses each and
      - \( \rho_{j+2^i-1}, \rho_{j+2^i} \) are a pair of different pages repeated for each \( i = 1, 2, \ldots \)
    - and are different from \( \rho_0, \rho_1, \ldots, \rho_j \)

  - \( f_{\text{LFU}}(\rho) = n - 2k + 1 \) and \( f_{\text{OPT}}(\rho) = 2 \)
  - Therefore the ratio \( f_{\text{LFU}}(\rho) / f_{\text{OPT}}(\rho) \) is \( O(n) \)
  - i.e. not bounded
Paging Algorithm - Competitive Analysis

- Online Paging Competitiveness Theorem:
  - For any deterministic online algorithm A for paging, $C_A \geq k$
  - Alternative Proof:
    - Imagine that the online algorithm A and the optimal algorithm O are managing separate caches with the same initial set of items for the same request sequence.
    - The first request is to an item not in either cache.
      - Both algorithms incur a miss.
    - Let $S = \{ \text{the new item} \} \cup \{ \text{all items initially in O’s cache} \}$
    - Every new request is for an item not in A’s cache.
      - A misses on every request.
    - Split the request sequence into rounds:
      - A round is a maximal sequence of requests in which at most $k$ distinct items are requested.
        - i.e. in each round A misses at least $k$ times but O misses exactly once (the first request in each round, which would have been O’s victim for the first request of the previous round).
Paging Algorithm - Competitive Analysis

- Online Paging Competitiveness Theorem:
  - For any deterministic online algorithm A for paging, $C_A \geq k$
  - Alternative Proof (contd.)
    - At the end of each round both A and O have the same set of items in their caches
    - i.e. there are arbitrarily long sequences on which A has k times as many misses as O

- Proof Technique:
  - We used only the fact that the online algorithm does not know future requests
  - We did not exploit any computational limitation of the online algorithm

- Thus the lower bound applies to any deterministic online algorithm regardless of space and time complexities
Paging Algorithm - Competitive Analysis

- Online Paging Competitiveness Theorem:
  - For any deterministic online algorithm A for paging, $C_A \geq k$

- Question:
  - Can the negative result in OPC Theorem be overcome using randomization?

- Proof Technique:
  - One can view the offline algorithm as an adversary who is not only managing a cache but is also choosing the inputs
    - and in this case an adversary who also knows the state of the algorithm (being analyzed).
  - In case of a randomized algorithm, if the adversary knows the state (in this case, current set of pages in the cache)
    - then the adversary knows the random choices made by the algorithm (in this case, the pages to be evicted)
Adversary Models

- There are different models for such an adversary
  - An oblivious adversary
    - is an adversary who does not know the random choices made by the algorithm
  - An adaptive adversary
    - is an adversary who gets to know the random choices made by the algorithm
      - and therefore gets to choose the next input based on the current choice

- An adaptive adversary may choose
  - to generate all the inputs and then execute its own algorithm (referred to as an adaptive offline adversary)
    OR
  - to execute its own algorithm as it generates its inputs (referred to as an adaptive online adversary)

- Question:
  - Why does this choice not matter for an oblivious adversary?
Competitiveness of Randomized Algorithms

- A randomized online paging algorithm $R$ would make a possibly random choice of which of the $k$ items in the cache it will evict:
  - Given a sequence $\rho$ of requests the number of misses is now a random variable, say, $f_R(\rho)$
  - Then $R$ is $C$-competitive against the oblivious adversary if for every sequence $\rho$ of requests
    
    $$E[f_R(\rho)] - C \cdot f_{\text{OPT}}(\rho) \leq b$$

    for some constant $b$ that is independent of the length of $\rho$
  - The *oblivious competitiveness coefficient* of $R$, denoted $C_R^{\text{obl}}$, is the smallest $C$ such that $R$ is $C$-competitive.

- We can define similar coefficients against adaptive adversaries.
Competitiveness of Randomized Algorithms

- Given a randomized online paging algorithm $R$ and a sequence $\rho$ of requests generated by an adaptive offline adversary
  - Let the number of misses by $R$ be denoted by the random variable $f_R(\rho)$
  - Let the number of misses by the optimal offline algorithm be denoted by the random variable $f_{OPT}(R)$
  - Question: Why is this a random variable?
- Then $R$ is $C$-competitive against the adaptive offline adversary if for every sequence $\rho$ of requests
  - $E[f_R(\rho)] - C \times E[f_{OPT}(\rho)] \leq b$
  - for some constant $b$ that is independent of the length of $\rho$
- The adaptive offline competitiveness coefficient of $R$, denoted $C_R^{aof}$, is the smallest $C$ such that $R$ is $C$-competitive.
Competitiveness of Randomized Algorithms

- Given a randomized online paging algorithm $R$ and a sequence $\rho$ of requests generated by an adaptive offline adversary
  - Let the number of misses by $R$ be denoted by the random variable $f_R(\rho)$
  - Let the number of misses by the optimal online algorithm be denoted by the random variable $f_{OPTON}(R)$
  - Then $R$ is $C$-competitive against the adaptive offline adversary if for every sequence $\rho$ of requests
    - $E[f_R(\rho)] - C \ast E[f_{OPTON}(\rho)] \leq b$
    - for some constant $b$ that is independent of the length of $\rho$
  - The adaptive offline competitiveness coefficient of $R$, denoted $C_{R}^{aon}$, is the smallest $C$ such that $R$ is $C$-competitive.
Competitiveness of Randomized Algorithms

- Clearly, by definition, for any randomized algorithm \( R \),
  \[ C_R^{obl} \leq C_R^{aon} \leq C_R^{aof} \]

- Let \( C^{obl} \) denote the lowest oblivious competitive coefficient among all randomized algorithms
  \[ i.e. \ C^{obl} = \min_R C^{obl} \]

- And similarly, let
  \[ C^{aon} = \min_R C^{aon} \]
  \[ C^{aof} = \min_R C^{aof} \]

- Also \( C^{det} \) denote the lowest competitive coefficient of any deterministic online paging algorithm:

- Then
  \[ C^{obl} \leq C^{aon} \leq C^{aof} \leq C^{det} \]
Paging against an oblivious adversary

- Consider an online deterministic paging algorithm A
  - Let $p$ be a probability distribution for choosing a request sequence
    - i.e. a distribution for choosing $\rho_i$ (which may depend on $\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i-1}$)
  - Both A’s cost and the optimal cost are now random variables.
  - Define A’s \textit{competitiveness coefficient} under $p$, denoted $C_A^p$, to be the smallest $C$ such that
    $$E[f_R(\rho)] - C \cdot E[f_{\text{OPT}}(\rho)] \leq b$$
    for some constant $b$ that is independent of the length of $\rho$. 

Paging against an oblivious adversary

- Also note that $C_{R^{obl}}$ can be interpreted as
  - $\max_{\rho} \{ \text{smallest } C \text{ such that } E[f_R(\rho)] - C \times f_{OPT}(\rho) \leq b \}$
  for some $b$ that is independent of the length of $\rho$

- i.e. by Yao’s minimax principle:
  - $\min_R C_{R^{obl}} = \max_{\rho} \min_A C_A^\rho$
    - i.e. the competitiveness of the best randomized online paging algorithm
  - is the same as
    - the competitiveness of the best deterministic online algorithm for a worst-case distribution on request sequences

- i.e. for any distribution $\rho$
  - $\min_R C_{R^{obl}} \geq \min_A C_A^\rho$

- i.e. for any distribution $\rho$ and any algorithm $R$
  - $C_{R^{obl}} \geq \min_A C_A^\rho$
Paging against an oblivious adversary

- **Theorem:**
  - Let $R$ be a randomized online algorithm for paging. Then $C_R^{obl} \geq H_k$, where $H_k$ is $k^{th}$ Harmonic number.

- **Proof:**
  - We will use Yao’s minimax principle i.e.
    - we will choose a probability distribution and
    - find the best performance for it by a deterministic online algorithm
    - which will be the lower bound for the performance of any randomized online algorithm
  - Let $I = \{ I_1, I_2, ..., I_{k+1} \}$ be the set of possible pages to be requested.
  - We will construct a probability distribution $\rho$ on request sequences $\rho$ of length $N \geq k$
    - and prove that for $\rho$ the best expected performance by a deterministic online algorithm is $H_k$. 
Paging against an oblivious adversary

Proof (contd.):

- Construction of a distribution $\rho$ on request sequences $\rho$:
  - the first request $\rho_1$ is chosen uniformly randomly from the items in $I$
  - for $i > 1$, request $\rho_i$ is chosen uniformly randomly from the items in $I - \{\rho_{i-1}\}$

- Divide the sequence into rounds:
  - each round is made of a maximal subsequence containing requests of $k$ distinct pages

- Claims:
  1. The expected length of such a round is $k^*H_k$
  2. For each round, the optimal offline algorithm would miss once.
  3. The expected number of misses per round by the best deterministic online algorithm, is $H_k$

- By Claims 2 and 3, $\min_{A} C_A^\rho = H_k$
Paging against an oblivious adversary

- **Claim 1:**
  - The expected length of a round is $kH_k$

- **Proof:**
  - Consider a complete graph $G$ with $k+1$ vertices.
    - Assume that a person walking along (the edges of) $G$ chooses any neighbor with equal probability (1/k in this case)
    - What is the expected number of steps before a person starting at a vertex ends up visiting every vertex at least once?
      - (Random Walks) Cover Time for a Complete Graph with $n$ nodes is $n-1 H_{n-1}$
      - We will study Random Walks later.
Paging against an oblivious adversary

- **Claim 2:**
  - For each round, the optimal offline algorithm would miss once

- **Proof:**
  - Each round includes requests for $k$ distinct pages.
    - So the greedy optimal algorithm would retain those $k$ pages in cache.
    - The first request in each round would be an item not in cache
    - and hence the optimal algorithm would incur a miss on that.
Paging against an oblivious adversary

- **Claim 3:**
  - The expected number of misses per round by the best deterministic online algorithm, say $A$, is $H_k$

- **Proof:**
  - At any point in time, $A$ must leave one of the $k+1$ items out of the cache.
    - Whenever a request falls on this item, $A$ incurs a miss
  - In distribution $p$, every request is chosen uniformly randomly from the $k$ items other than the last requested item
    - The probability that a request falls on the item left out by $A$ is $1/k$.
  - The expected number of misses on a round of length $kH_k$ is then $H_k$
The marker algorithm
- maintains a *marker bit* with each cache location
- and proceeds in a series of rounds.

For each round:
- At the beginning, all marker bits are set to 0.
- For each request:
  - if the item is in a cache location, then the corresponding marker is set to 1
  - if the item is a miss
    - choose an unmarked cache location, u, uniformly randomly
    - evict the item in u and set its marker to 1
    - bring the requested item into u
- If all locations are marked, the round ends.
Online Paging - Marker Algorithm

- Theorem: The Marker algorithm is $(2H_k)$-competitive.

- Proof:
  - (By comparing the Marker algorithm’s performance against the optimal offline algorithm on the same sequence of inputs)
  - The Marker algorithm implicitly divides the request sequence into rounds such that
    - there are exactly $k$ distinct items in each round
  - Define an item to be
    - stale – if is unmarked (in this round) but was marked in the previous round
    - clean – if it is neither stale nor marked
  - Let $L$ be the number of requests to clean items in a round.
Online Paging - Marker Algorithm

- Theorem: The Marker algorithm is \((2H_k)\)-competitive.

- Proof (continued):
  - Claims:
    - The amortized number of misses per round by the offline algorithm is at least \(L/2\)
    - The expected number of misses per round by the Marker algorithm is \(L*H_k\)
    - So, the competitive ratio is \(2*H_k\).
Paging against an oblivious adversary

- Compare the results of the previous two theorems i.e.
  - $C_R^{obl} \geq H_k$ for any randomized algorithm $R$ and
  - $C_M^{obl} = 2*H_k$ for the Marker algorithm

- with that of the lower bound result for deterministic algorithms:
  - $C_A \geq k$ for any deterministic algorithm $A$

- How do you interpret the results?
Relating the Adversaries

- How do randomized online paging algorithms compete against adaptive adversaries?

- Theorem (without proof): \[\text{applicable for a general class of request-answer problems including paging}\]
  - If there is a randomized algorithm that is $\alpha$-competitive against every adaptive offline adversary, then there exists an $\alpha$-competitive deterministic algorithm.

- Implication (for paging):
  - There is no randomized online algorithm for paging with a competitiveness coefficient smaller than $k$. 
Relating the Adversaries

- Theorem (without proof): [applicable for a general class of request-answer problems including paging]

  - If there is a randomized algorithm R that is $\alpha$-competitive against every adaptive online adversary, and there is a $\beta$-competitive randomized algorithm against any oblivious adversary then R is $(\alpha \cdot \beta)$-competitive against any adaptive offline adversary.

- Implication (for paging): ??