Online Problems and Online Algorithms

- Online Paging Problem:
  - Definition
  - Performance Parameter
  - Optimal Offline Algorithm

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 >= 2 is the size of M_0
    - Why is this a reasonable assumption?
- **Greedy Paging Lemma:**
  - GreedyPaging is optimal and $f_{OPT}([\rho_0, \rho_1, \ldots \rho_n]) = n / k$
  - 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 Algorithms – Miss Rates- RECALL

- 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
  - that by an optimal offline algorithm as $f_{OPT}(\rho)$
- An offline paging algorithm (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
- Steady-State Performance Assumption: *Cold misses are not counted*
- Greedy Paging Lemma:
  - GreedyPaging is optimal and $f_{OPT}([\rho_0, \rho_1, \ldots \rho_n]) = n / k$