
Online and Approximation Algorithms

Due October 30, 2017 at 10:00

Exercise 1 (Ski Rental – 10 points)

The ski rental problem is defined as follows: Assume that renting a pair of skis costs 1 per day while buying a pair of skis costs b . Every day we have to decide, in an online fashion, whether we want to continue renting skis for another day or buy a pair of skis. At some unknown time D , we will break our leg and have to quit skiing. Our goal is to minimize the cost of skiing.

1. What is the optimal offline cost?
2. Develop a $(2 - \frac{1}{b})$ -competitive online algorithm ALG for the ski rental problem and prove its competitive ratio.
3. Is there any (deterministic) algorithm that achieves a competitive ratio smaller than $2 - \frac{1}{b}$?

Exercise 2 (Growing the fast memory – 10 points)

Recall that LRU is the online paging algorithm that evicts the page that was used least recently, FIFO is the online paging algorithm that evicts the page that has been in fast memory for the longest time and OPT is the optimal offline algorithm. Consider the following request sequence

$$\sigma = ABCDABEABCDE$$

Assume that we start with an empty fast memory. Keep track of the content of the fast memory during executions of OPT, LRU and FIFO on σ for fast memory sizes 3 and 4. Count the number of page faults during the executions. What do you observe?

Exercise 3 (First-in First-out – 10 points)

Prove that FIFO is k -competitive, where k is the number of pages that fit in fast memory.

Exercise 4 (Least Frequently Used – 10 points)

The online paging algorithm *Least Frequently Used* (LFU) evicts the page that has been used the least from among all the pages in the cache.

1. Is there any request sequence where LFU has fewer page faults than LRU (Least recently used)?
2. Prove or disprove: LFU is k -competitive.