TD 10 : LRU, Location de skis, MAX-SAT

1 Gestion de mémoires caches : Least Recently Used (LRU)

Consider a cache of size k that can hold up to k pages. We are given a sequence of page requests $\sigma = (p_1, \ldots, p_n)$ where each p_i is a request to a page.

> Least Recently Used (LRU) algorithm :
> — If a requested page is already in the cache (a cache hit), no eviction occurs.
> — If the requested page is not in the cache (a cache miss), the algorithm evicts the page for which the longest time has passed since it was last requested and inserts the requested page into the cache.

Assume that the cost of serving a request is 0 if it is a cache hit and 1 if it is a cache miss.

Problem 1.1. What is the competitive ratio of the LRU caching algorithm ?

2 Problème de la location de skis (Ski Rental Problem)

You are going skiing in the mountains and you want to keep skiing until the day you crash badly. Renting a ski involves paying 1 unit of currency at the start of each day, and buying it involves paying $B > 1 \in \mathbb{N}$ units at the start of the day. Suppose you ski for x (entire) days, but x is not known in advance.

Problem 2.1. What is the best strategy for buying (or not buying) the skis ?

3 MAX-SAT

The Maximum Satisfiability problem is defined as follows. Given a conjunctive normal form $f = C_1 \wedge \cdots \wedge C_m$ formula on Boolean variables $x_1, \ldots, x_n$, find a truth assignment to the Boolean variables that maximizes the total number of satisfied clauses.

Problem 3.1. Give a randomized $(1 - 1/e)$-approximation algorithm $\mathcal{A}_1$ for the MAX-SAT problem using the randomized rounding technique.

Problem 3.2. Give a simple randomized 1/2-approximation algorithm $\mathcal{A}_2$.

Problem 3.3. Show how to derandomize both of the algorithms above.

Problem 3.4. Explain why $\mathcal{A}_1$ is good for dealing with small clauses while $\mathcal{A}_2$ is good for dealing with with large clauses. Show how to combine $\mathcal{A}_1$ and $\mathcal{A}_2$ to obtain a randomized 3/4-approximation algorithm.

Problem 3.5. Give a deterministic 3/4-approximation algorithm for MAX-SAT.

Homework

Homeworks may be written in either French or English, and are to be submitted in person or by email to `dghosh@lmf.cnrs.fr` *by the end of 23 April 2026. And please, avoid using an LLM so that everyone can be graded fairly.*

Consider the following two-player game. The game begins with k tokens placed at the number 0 on the integer number line spanning $[0, n]$. Each round, one player, called the *chooser*, selects two disjoint and nonempty sets of tokens A and B. (The sets A and B need not cover all the remaining tokens ; they only need to be disjoint.) The second player, called the *remover*, takes all the tokens from one of the sets off the board. Then tokens from the other set all move up one space on the number line from their current position. The chooser wins if any token ever reaches n. The remover wins if the chooser finishes with one token, and moreover the token has not reached n.

1. Give a winning strategy for the chooser when $k \geq 2^n$.
2. Use the probabilistic method (see TD 9) to show that there must exist a winning strategy for the remover when $k < 2^n$.
3. Explain how to use the method of conditional expectations to derandomize the winning strategy for the remover when $k < 2^n$.