Exercise: Randomized Online Algorithms

1 Modified Bit Guessing Problem

To show that the gap between the payoff of a randomized algorithm and the payoff of a deterministic algorithm can be arbitrarily large, consider the Modified Bit Guessing Problem.

Modified Bit Guessing Problem.
Input: $(x_1, x_2, \ldots, x_n)$ where $x_i \in \{0, 1\}$.
Output: $(y_1, y_2, \ldots, y_n)$ where $y_i \in \{0, 1\}$.
Objective: Find y such that $y_i = x_{i+1}$ for some $1 \leq i < n$. If such y_i exists the payoff of the algorithm is $g(n)/(1 - 1/2^{n-1})$, otherwise the payoff is 1.

In this problem the adversary presents the input bits one by one and the goal is to guess the bit arriving in the next time step based on the past history. If the algorithm manages to guess at least one bit correctly, it receives a large payoff of $g(n)/(1 - 1/2^{n-1})$. Otherwise, the algorithm pays 1.

(a) Prove that every deterministic algorithm ALG has a worst-case objective value of 1.
The adversarial strategy is as follows. Present $x_1 = 0$ as the first input item. The algorithm replies with y_1. The adversary defines $x_2 = \neg y_1$. This continues for $n - 2$ more steps. Hence the algorithm does not guess any of the bits, and the objective value is 1.

(b) Prove that there is a randomized algorithm RAND that receives expected objective value of at least $g(n)$ against an oblivious adversary.
Consider the randomized algorithm RAND that selects y_i uniformly at random, i.e., it picks $y_i = 0$ with probability 1/2 and $y_i = 1$ with probability 1/2. Then, the probability that it picks y_i different from x_{i+1} is 1/2. Therefore, the probability that it guesses none of the $y_1, y_2, \ldots y_{n-1}$ correctly is $1/2^{n-1}$. Hence the probability that it guesses at least one bit correct is $1 - (1/2^{n-1})$. Then the expected value is at least $1 - 1/2^{n-1} \cdot g(n)/(1 - 1/2^{n-1}) = g(n)$.

2 RMark

In the lecture we have seen that the algorithm RMark is $2H_k$-competitive in expectation.

(a) Prove that RMark is H_k-competitive in expectation when the size of the main memory is $k+1$.
Observe that the number of new pages ℓ_j ,in each phase P_j, equals 1. Then,

$$\mathbb{E}[ALG] = \sum_{j=1}^{N} \mathbb{E}[\mathcal{C}_j] = \sum_{j=1}^{N} \ell_j H_k = \sum_{j=1}^{N} H_k$$

Furthermore, algorithm LFD (which is offline optimal) causes precisely 1 page fault in each phase, i.e., the first request in each phase. Thus, OPT causes precisely 1 page fault per phase. Then, OPT causes N page faults in total.
Therefore,

$$\frac{\mathbb{E}[ALG]}{OPT} \leq \frac{\sum\limits_{j=1}^{N} H_k}{N} = \frac{NH_k}{N} = H_k$$

(b) Prove that, in general, RMark is not H_k competitive in expectation.
(Hint: Consider the case where $k=2$ and the size of the main memory is 4.)
Assume w.l.o.g that OPT and ALG start with pages 1 and 2 in the cache.
Consider the input sequence $(3,4,1,2)^m$ repeated for some large m.
Each phase consists either of pages 3 and 4 or of pages 1 and 2.
Observe that due to the marking property, and there being k new pages requested per phase, ALG always has to replace both the pages it started the phase with. Thus, ALG causes 2 page faults per phase.
The optimal algorithm, except for the first phase, causes just a single page fault per phase.
Thus, for sufficiently large N,

$$\frac{\mathbb{E}[ALG]}{OPT} = \frac{\sum\limits_{j=1}^{N} 2}{2 + \sum\limits_{j=2}^{N} 1} = \frac{2N}{1+N} > \frac{3}{2} = H_k$$