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.

(b) Prove that there is a randomized algorithm RAND that receives expected objective value of at least $g(n)$ against an oblivious adversary.

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 $k + 1$.

(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.)