

# Algorytmy online: lista 2

**Zadanie 1. (2 pkt.)** Aby zrobić miejsce w pamięci podręcznej, algorytm FIFO usuwa z niej stronę, która była tam najdłużej. Pokaż, że FIFO jest nie jest algorymem zaznaczającym. Pokaż, że FIFO jest  $k$ -konkurencyjny.

**Zadanie 2. (2 pkt.)** LFU (*Least Frequently Used*) to algorytm, który z każdą stroną wiąże licznik określający, ile odwołań wystąpiło do tej strony. Zapisanie strony w pamięci podręcznej lub wyrzucenie jej stamtąd nie powoduje zerowania takiego licznika. Aby zrobić miejsce w pamięci podręcznej, LFU wyrzuca z niej stronę o najmniejszej wartości licznika. Pokaż, że LFU nie jest konkurencyjny.

**Zadanie 3. (2 pkt.)** Celem tego zadania jest udowodnienie, że konkurencyjność dowolnego algorytmu deterministycznego DET dla problemu pamięci podręcznej wynosi co najmniej  $k$ . Założymy, że algorytm DET startuje z pełną pamięcią podręczną. Rozważ ciąg wejściowy  $\sigma^n$  o długości  $n$  składający się z  $k+1$  różnych stron, w którym adwersarz zawsze pyta o stronę, której DET nie ma w pamięci podręcznej.

- a) Pokaż, że  $\text{OPT}(\sigma^n) \leq 1+n/k$ . Wskazówka: jak na ciągu  $\sigma^n$  zachowa się algorytm Longest Forward Distance?
- b) Wywnioskuj, że  $\text{DET}(\sigma^n) \geq k \cdot \text{OPT}(\sigma^n) - k$ .
- c) Jak z poprzedniego punktu wynika dolne ograniczenie na konkurencyjność DET?

**Zadanie 4.** Założymy, że algorytm dysponuje pamięcią podręczną o rozmiarze  $k$ , a optymalny algorytm pamięcią o rozmiarze  $h \leq k$ .

- a) (1 pkt.) Pokaż, że dowolny deterministyczny algorytm zaznaczający jest  $\frac{k}{k-h+1}$ -konkurencyjny.
- b) (3 pkt.) Pokaż, że konkurencyjność dowolnego algorytmu deterministycznego (niekoniecznie zaznaczającego) wynosi co najmniej  $\frac{k}{k-h+1}$ .

Marcin Bieńkowski