

DM205 – Ugeseddel 7

Mandag d. 21/5

Forelæsning:

- Sætning 9: $WR_{LRU,LRU^\ell} = \min\{\ell + 1, k\}$.
- Sætning 1: $WR_{LRU,RLRU} = (k + 1)/2$.
- Sætning 2: $\mathcal{R}(RLRU) = k + 1$.
- Sætning 3: $WR_{LRU} = WR_{RLRU} = k$.

Opgaver:

1. Sidste gang glemte vi det sidste spørgsmål i opgave 1:
Der gælder ikke nødvendigvis, at $WR_{A,C} \geq WR_{A,B} \cdot WR_{B,C}$ (selv hvis $1 \leq WR_{A,B} \leq c$ og $WR_{B,C} \geq 1$, hvor c er en konstant).
Hvorfor ikke?
2. Lemma 4 i artiklen gælder ikke, hvis den konservative algoritme har look-ahead.
Hvordan ved vi det? Hvor fejler beviset?
3. Definer en algoritme baseret på FIFO, som bruger look-ahead.
Hvad er dens relative worst-order ratio i forhold til FIFO?
Hvad er dens relative worst-order ratio i forhold til LRU?

Torsdag d. 24/5

Vi begynder på k -server problemet:

- Afsnit 10.1-10.4.

Opgaver:

1. Definer en *strict* relative worst-order ratio (analogt til *strict* competitive ratio).
2. I denne opgave ser vi på et problem, som hedder *bin coloring*:
Vi har q bins til rådighed.
Hver bin har størrelse B .
Vi skal pakke en sekvens af farvede elementer.
Hvert element har størrelse 1.
Når en bin er fyldt, forsvinder den og erstattes af en ny. D.v.s. vi har altid q bins til rådighed.
Målet er at minimere det maksimale antal farver, som pakkes i nogen bin.

Betragt følgende to algoritmer:

One-Bin fylder blot bins'ne en for en.

Denne algoritme er strictly $(2q - 1)$ -competitive.

Greedy gør følgende, når den skal pakke et element med farve c : Hvis der findes en bin, som allerede indeholder farven c , pakkes elementet der. Ellers pakkes elementet i en bin med færrest muligt farver.

Denne algoritmes strict competitive ratio er mindst $2q$.

D.v.s. meget overraskende, har Greedy en dårligere competitive ratio end den meget simple One-Bin.

Vis, at med hensyn til strict relative worst-order ratio er Greedy den bedste af de to algoritmer.

Fredag d. 25/5

Load balancing:

- Afsnit 12.1-12.2