8. CVIČENÍ Z APROXIMACÍ — online stránky a online servery

Nejprve se zaměříme na stránkování (paging). Přehled algoritmů pro stránkování (níže je popis, co se stane, pokud je cache plná a požadovaná stránka v ní není):

- LRU (Least Recently Used) — vyhodí stránku, na níž byl poslední přístup nejdále v minulosti,
- FIFO (First-In First-Out) — vyhodí stránku, jež je v paměti nejdéle,
- LIFO (Last-In First-Out) — vyhodí stránku, jež je v paměti nejkratší dobu,
- LFU (Least Frequently Used) — vyhodí stránku, na kterou bylo přistoupeno nejméně krát, od doby, co se dostala do rychlé paměti (každá stránka v rychlé paměti má tedy počítadlo),
- FWF (Flush When Full) — když je cache plná, vyhodí z rychlé paměti všechny stránky,
- RAND — vyhodí uniformně náhodnou stránku,
- MARK — vyhodí uniformně náhodnou *neoznačenou* stránku, přičemž požadovanou stránku vždy označí a pokud jsou označené všechny stránky, tak smaže značky.
- LFD (Longest Forward Distance) — vyhodí stránku, která bude nejpozději znovu požadovaná (není online),

PŘÍKLAD PRVNÍ Jaká je časová a prostorová složitost jednotlivých algoritmů pro stránkování? (Uložení samotných stránek nepočítáme.)

PŘÍKLAD DRUHÝ Ukažte, že LIFO a LFU nejsou kompetitivní.

A teď něco ke k-serveru:

PŘÍKLAD TŘETÍ Vymyslete algoritmus, který spočte offline optimum pro k-server.

Hint: redukce na maximální tok minimální ceny (ceny hran mohou být i záporné).

PŘÍKLAD ČTVRTÝ Připomeňte si algoritmus *DoubleCoverage-Tree* pro k-server na stromech z přednášky a fakt, že stránkování je speciální případ k-serveru na stromech — jak vypadá strom pro stránkování? Jakému algoritmu pro stránkování odpovídá *DoubleCoverage-Tree*?

PŘÍKLAD PÁTÝ Mějme libovolnou metriku s n body. Jaký nejlepší pravděpodobnostní online algoritmus pro k-server jste schopni navrhnout pro takovou metriku s pomocí toho, co bylo nedávno na přednášce?

A nakonec zase stránkování:

PŘÍKLAD ŠESTÝ Ukážeme, že pravděpodobnostní algoritmus RAND pro stránkování (který vyhodí uniformně náhodnou stránku) je k-kompetitivní. Použijeme následující potenciál:

$$\Phi_i = k(k - x_i)\,,$$

kde x_i je počet stránek, které má v cachi jak RAND, tak optimum (adversary).

a) Jak pomocí potenciálu amortizovat cenu algoritmu? Co chceme ukázat o amortizované ceně pro důkaz k-kompetitivnosti?
b) Při požadavku r analyzujte změnu potenciálu a amortizovanou cenu v jednotlivých případech podle toho, jestli je r v cachi algoritmu či adversaryho.
c) Dokážete najít vstup, na kterém je střední hodnota ceny algoritmu RAND k krát cena optima?