

7. cvičení

Datové struktury I, 13. 11. 2023

<https://iuuk.mff.cuni.cz/~chmel/2324/ds1/>

Úloha 1 (LRU vs OPT)

Jak velký může být poměr počtu výpadků LRU a OPT strategií, používají-li stejně velkou cache? (Na začátku přepokládejte prázdnou cache.)

Úloha 2 (Jinak velká transpozice)

V cache-oblivious algoritmu¹ jsme tak nějak předpokládali, že velikost matice je mocnina dvojky (tj. je velikosti $2^n \times 2^n$). Co se stane, pokud tomu tak není? Dokažte, že pro matici $n \times n$ jsou všechny podmatice v průběhu *skoro čtvercové*, tedy počet řádků a sloupců se liší nejvýše o jedna. Potom nahlédněte, že tato vlastnost algoritmus nijak nerozbije.

Úloha 3 (Doslovný transpose-and-swap)

V cache-oblivious algoritmu je snadnou chybou provedení transpose-and-swap doslova. Tedy, nejdřív obě podmatice rekurzivně ztransponujeme, a pak je prohodíme. Zanalyzujte *časovou* složitost tohoto algoritmu a poté i I/O složitost (tedy počet přenesených bloků).

Úloha 4 (Rekurzivně násobíme matice)

Spočtete I/O složitost rekurzivního násobení matic. (Toho naivního², kdy matice rozdělíme na čtvrtiny a násobíme spolu tyto čtvrtiny, které pak k sobě přičítáme.)

Můžete předpokládat vysokou cache: $M \in \Omega(B^2)$.

¹Pokud si ho nepamätujete, připomeňte si ho, a pokud chcete, zkuste si odsimulovat, co se tam vlastně děje, na 8×8 matici s velikostí bloku $B = 3$.

²Ale pokud se vám chce počítat Strassena, bránit vám v tom nehodlám.