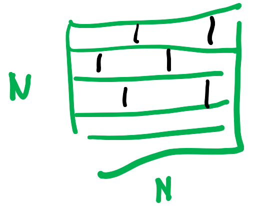
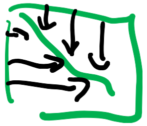
Transpozice matic



přečíst $\Rightarrow$ $O(N^2/B+1)$ I/O složitost

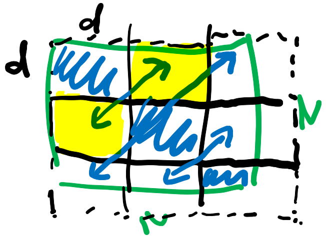
zapsat $\Downarrow$ $\Theta(N^2)$ při $M<(1-\varepsilon)NB$

"naivní" alg.



$\Theta(N^2)$ I/O slož. pro $M=o(NB)$

Dláždění (cache-aware alg.)



IDEA: rozdělit na menší, které se vejdou do cache

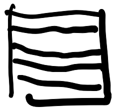
Alg: diag. dlaždice … transpozice, nediag. … transpozice + prohození

\# dlaždic $d\times d = \lceil N/d\rceil^2 = O(N^2/d^2+1)$

cíl: 2 dlaždice do cache $\Rightarrow$ zvolme $d=B$ (stačí cache $M=\Omega(B^2)$, případně vhodný násobek)

1) B dělí N (nebo $N<B$)

1 dlaždice … B bloků $\Rightarrow O(B)$ přenosů

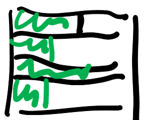


celkem $O(N^2/B+\cancel{B}\,1)$ I/O složitost

(pro $N<B$ stačí 1 dlaždice) $M\geq 2B^2$

2) B nedělí N

1 dlaždice … 2B bloků



$\Rightarrow O(B)$ přenosů

$\Rightarrow$ celkem stejně

$M\geq 4B^2$

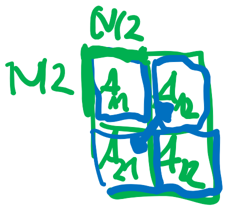
$\Rightarrow$ cache-aware $O(N^2)$ čas, $O(N^2/B+1)$ I/O slož.

cache-oblivious alg.

IDEA: alg. nezná B, M, ale řadič cache ano! Alg. rekurzivně dělí na podproblémy, na určité úrovni může řadič volit stejnou strategii jako cache-aware

=> I/O složitost stejná

Rozděl a panuj



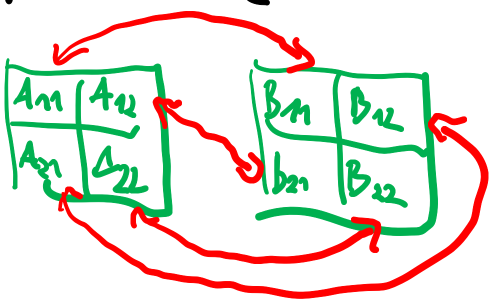
1) $N = 2^k$ pro nějj. $k \in \mathbb{N}$

přesun $O(1)$ bloků v cache stačí

"naivní" alg.: $T(A) \to T(A_{11}), T(A_{12}), T(A_{21}), T(A_{22}) + S(A_{21}, A_{12})$

$T_N \to 4T_{N/2} + S_{N/2}$ čas $O(N^2/4)$ => celkem $O(N^2 \log N)$ čas

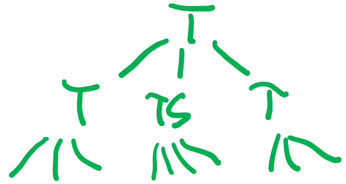
operace $TS(A,B)$ ... "transponuj a prohoď"



$T \to 2T + TS$

$TS \to 4TS$

strom rekurze



v úrovni i

$\leq 4^i$ uzlů vel. $\frac{N}{2^i}$

hloubka $\log N = k$

=> # vnitřních $\leq$ # listů $\leq 4^{\log N} = N^2$

výpočet: list $T(a_{ii})$ ... nic nedělá
list $TS(a_{ij}, a_{ji})$ ... prohoď

vnitřní: rozděl práci (spočti indexy podmatic)

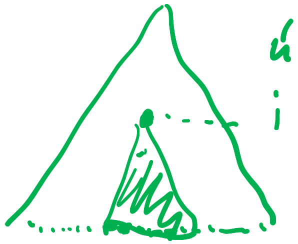
=> celkem $O(N^2)$ čas

analýza I/O složitosti

Je možné mít "dobrou" strategii?

přístup do paměti: list TS (+ zásobník rekurze $O(\log N)$)

↑ udržuj v cache



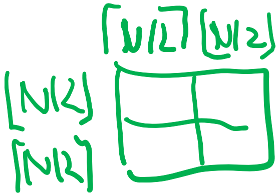
úroveň $i = \arg\min_i \frac{N}{2^i} \leq B$

strategie: od úrovně i, udržovat v cache bloky pro tento podstrom během výpočtu v celém podstromě

$\Rightarrow$ ekvivalentní jako cache-aware při $d = \frac{N}{2^i}$

$B/2 < d \leq B \underset{d=\Theta(B)}{\Rightarrow}$ stejné $O(N^2/B + 1)$ I/O složitost (za předp. štíhlé cache)

2) N není mocnina 2



$\Rightarrow$ podproblémy čtvercové nebo skoro-čtvercové

$\Rightarrow$ asymptoticky stejné

cache aware vs cache oblivious alg.:

$\Rightarrow$ $O(N^2)$ čas, $O(N^2/B+1)$ I/O složitost

$\Rightarrow$ $d = B$ optimální, $d = \Theta(B)$ asymp. optimální

$\Rightarrow$ cache-oblivious větší režii

Model vs realita

model: OPT strategie (offline): "vyhoď řádku, která bude potřeba co nejpozději."

realita: pouze online strategie: znají stav cache + aktuální požadavek na řádek (minulé požadavky)

Def: cena strategie vzhledem k posloupnosti přístupů = # cache missů

STR je k-kompetitivní: pro každou vel. cache, každou posloupnost $T_{STR} \le k \cdot T_{OPT}$

LRU strategie: "vyhoď řádku, která se nejdéle nepoužívá" (pomocí prioritní fronty s časovými razítky)

Je LRU k-kompetitivní pro nějaké k?

Věta: Pro lib. vel. C a lib. $\varepsilon > 0$ existuje posloupnost t.ž. $T_{LRU} \ge (1-\varepsilon) C \, T_{OPT}$

Dů: $\overbrace{1, \dots, C+1, \ 1, \dots}^{k\text{-krát}} C, C+1, \dots$ — epochy délky C

—, 1, 2, …… — LRU vyřadí ⇒ až na první C, vždy miss

vše, C, —, C−1, — — EPOCH ⇒ až na první C, 1 miss ✓ každé epoše

EPOCH: "vyhoď řádku, která není třeba v aktuální epoše"

$\Rightarrow T_{LRU} \ge C \, T_{EPOCH}$ (kromě začátku) $\Rightarrow T_{LRU} \ge (1-\varepsilon) C \, T_{OPT}$ □

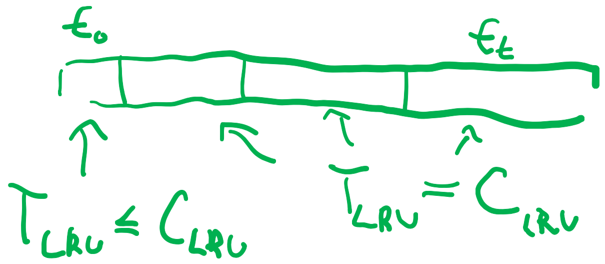
LRU s větší cache

Např.: Pro $C_{LRU} = 2C_{OPT}$ je LRU je $(2+\varepsilon)$-kompetitivní pro dostatečně dlouhé posl.

<u>Věta</u>: Pro $C_{LRU} > C_{OPT} \ge 1$ a lib. přístupovou posloupnost

$$T_{LRU} \le \frac{C_{LRU}}{C_{LRU}-C_{OPT}} T_{OPT} + C_{OPT} \qquad (*)$$

Dk:



epochy t.ž.

Kolik výpadků bude mít OPT v každé epoše?

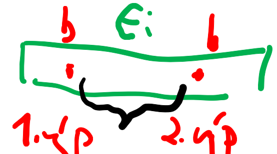
1) $E_i,\ i>0$ $\Rightarrow$ v E_i: $T_{LRU} \le \frac{C_{LRU}}{C_{LRU}-C_{OPT}} \cdot T_{OPT}$

a) LRU výpadky na různých blocích

$\Rightarrow$ $T_{OPT} \ge T_{LRU} - C_{OPT}$ v této epoše

b) LRU výpadek na něj. bloku dvakrát

$\Rightarrow$ $T_{OPT} \ge T_{LRU} - C_{OPT}$ –//–



$\ge C_{LRU}$ různých bloků
$\shortparallel$
T_{LRU}

2) E_0: $T_{OPT} \ge T_{LRU} - C_{OPT}$

$\Rightarrow$ $T_{LRU} \le \overbrace{\frac{C_{LRU}}{C_{LRU}-C_{OPT}}}^{>1} T_{OPT} + C_{OPT}$ v E_0

přes všechny epochy $\Rightarrow$ $(*)$ □