

## 6. CVIČENÍ Z DATOVÝCH STRUKTUR 1, ZS23/24

Kešujeme

1.  $(a, b)$ -join. Navrhněte operaci  $\text{JOIN}(X, Y)$ , která dostane dva  $(a, b)$ -stromy  $X$  a  $Y$  a sloučí je do jednoho. Může se přitom spolehnout na to, že všechny klíče z  $X$  jsou menší než všechny z  $Y$ . Zkuste dosáhnout složitosti  $O(\log |X| + \log |Y|) = O(\log(|X| + |Y|))$ .

2. *Analýza algoritmů dle starověkého „Rozděl a panuj!“* Proveďte analýzu počtu přenesených bloků, tedy I/O složitost, pro následující algoritmy (pokud jsou pravděpodobnostní, určete střední hodnotu):

- QuickSelect (s náhodným výběrem pivota)
- LinearSelect (deterministický lineární algoritmus pro hledání  $k$ -tého nejmenšího)
- Bonus: QuickSort (s náhodným výběrem pivota)

3. *Násobení matic.* Chceme spočít matici  $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ , kde  $\mathbf{A}$  a  $\mathbf{B}$  jsou zadané čtvercové matice  $n \times n$ . Pro následující přístupy určete I/O složitost:

- Nejprve předpokládejte, že  $\mathbf{A}$  je uložena po řádcích a  $\mathbf{B}$  po sloupcích, a použijte přímočarý algoritmus pro násobení dle definice.
- Nyní každou z matic rozdělíme na 4 bloky, které vynásobíme rekurzivně. (Pro jednoduchost předpokládejme, že  $n$  je mocnina dvojky.) Pro analýzu předpokládejte, že cache je „štíhlá“, tedy  $M \geq c \cdot B^2$  pro nějakou konstantu  $c$  (tzv. tall cache assumption).

4. *Binární vyhledávání optimálně.* Binární vyhledávání v uspořádaném poli má I/O složitost  $\Theta(\log N - \log B + 1)$ . Navrhněte způsob uložení prvků do pole tak, abychom mohli vyhledávat s I/O složitostí  $O(\log_B N + 1) = O(\log N / \log B + 1)$  (tato složitost je dokonce optimální). Hint: sestrojte nejprve dokonale vyvážený BVS.