

Dynamické lineární uspořádání

List Order Problem

- Chceme udržovat posloupnost prvků
- Operace:
 - Insert nového prvku za řadání
 - Compare - srovnání, zda je x před y --- chceme v $O(1)$
 - možná také Delete (ne via glob. přestavba)

řeší se pomocí

List Labelling

- Posloupnost prvků, každému přidělena značka, značky rostou zleva doprava
- Insert, Delete mohou přeznačovat

1) exponenciální rozsah značek, především vlně max. # prvků M

- první prvek dostane 2^M
- druhý buď 0 nebo 2^{M+1}
- každý další je přírůstek sousedů

} nikdy není třeba přeznačovat, vše $O(1)$ w.c.
 ↓
 použitelné pro $M = O(\text{word size})$

2) polynomiální rozsah značek

- BVS, značka = posl. L/P na cestě z kořene do prvku - $O(\log n)$ bitů → poly rozsah
- B+ stromy přepočítávají značky během rekonstrukce → $O(\log n)$ amort. na Ins/Del
 ↳ porovnání, rotace jsou drahé

3) Lineární rozsah - Ordered File Maintenance → poradí udávkou $O(\log^2 n)$ amort.

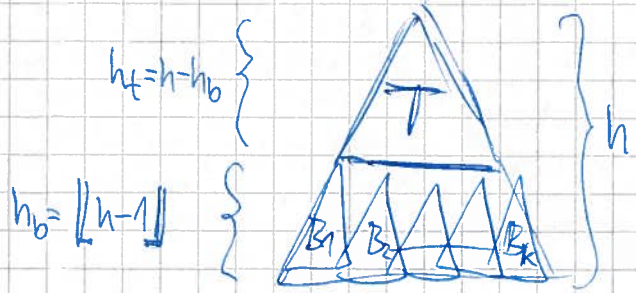
↳ lze zrychlit indikací & bloky velikosti $O(\log n)$ v nich
 1) nad bloky $O(n/\log n)$ bloků
 stojí $O(1)$ amort.
 Insert v D zprávy $O(1/\log n)$ amort. ~~ins~~ operaci ve 2
 ⇒ 2) stojí $O(1)$ amort., 1) také.

} label je dvojice, porovnáváme lexicograficky → Compare v $O(1)$ w.c.
 [Práv, 1 značka ve 2) má $O(\log n)$ dvojic, ale to lze udělat najednou]

Cache-oblivious datové struktury

- I/O model, parametry B (velikost bloku), M (velikost cache)
- c/I/O model - parametry uvažujeme, cache se obsluhuje optimalně } D IMO počítáme jen čteč
- cache-aware (I/O): (a, b) -strom s $a, b \in O(B) \rightarrow O(\log n / \log B)$ I/O na operaci
- cache-oblivious & staticky $\xrightarrow{\text{eplos}}$ BVS ve van Emde-Boosově uložení

dynamický? ↓ *



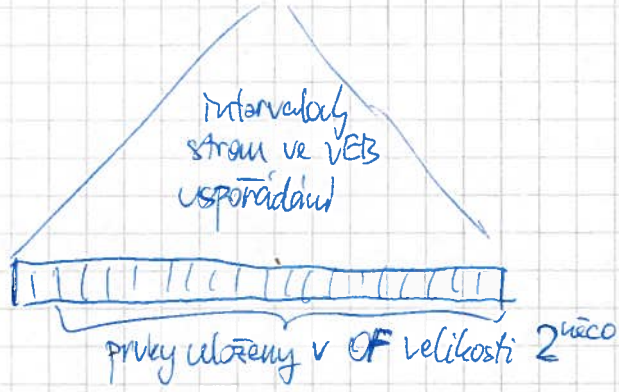
nejprve T , pak $B_1 - B_{k-1}$
 vše rekursivně...

Věta: Průchod kořen - list vyžaduje $O(\log_B N)$ I/O

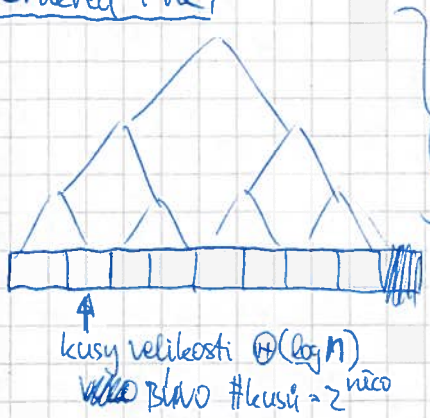
Náčrtek D: "zaostříme" na úroveň rozkladu, když se stromu poprvé vejde do bloku → má hloubku $O(\log B)$
 ⇒ na cestě takových průchodů $O(\log N / \log B)$

* zkrácíme Ordered File s VEB uspoř. BVS:

- Find je plně v režii streamu
- Insert vloží do OF, to způsobí přečíslování nejmenšího intervalu klicí \Rightarrow důležitý update streamu



Ordered File



čisté konceptuální úplný B+ strom
vnitřní vrchol \approx interval

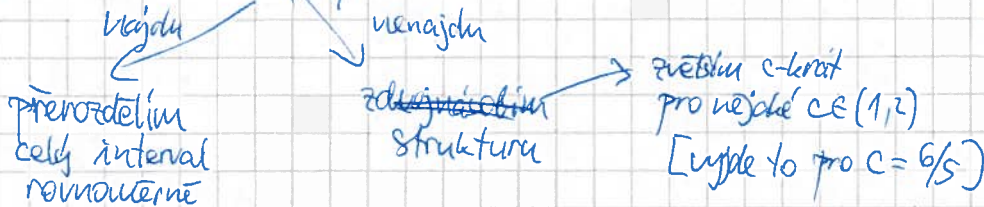
každý interval má kapacitu $\log n \cdot 2^{h-1}$
(h = hloubka stromu, i = hloubka vrcholu)
a hustoty $\rho = \#prvků / kapacita$

Standardní hustota $[\frac{1}{2} - \frac{1}{4h}, \frac{3}{4} + \frac{1}{4h}]$

v koreni ($i=0$) $[\frac{1}{2}, \frac{3}{4}]$
v listech ($i=h$) $[\frac{1}{4}, 1]$
↑ čím výš, tím přisluješ

Insert (Delete analogicky):

- vložíme do příslušného kusu ($O(\log n)$, úplně ho přepíšeme)
- pokud má stále stl hustotu, kladem jinak jdu nahoru a hledám první interval se stl hustotou



upravuji konst. ve velikosti kusu tak, aby #kusů stoupl po maximální dvojnásobky

Amortizace: Necht přerodělujeme interval v hloubce i , který má kapacitu K_0 .

Jeho ~~$\rho \leq \frac{1}{4} + \frac{1}{4h}$~~ $\rho \leq \frac{3}{4} + \frac{1}{4h}$, alespoň 1 syn má $\rho > \frac{3}{4} + \frac{1+1}{4h}$

Přerodělení stojí $\Theta(K)$

Vstoupíme $\Theta(h) \leq O(\log n)$ jednému prvku

• prvek přispěje na přerodělování v celkem $\log n$ úrovních, tedy celkově $O(\log^2 n)$

po minulém přerodělení otec měl ρ stejnou jako otec \Rightarrow vzrostla o alespoň $1/4h \Rightarrow$ v podstromu přibýlo alespoň $\Theta(K/h)$ prvků

! velikost kusu nastavíme tak, aby se změna hustoty o $1/4h$ projevila přidáním/ubráním aspoň 1 prvku ∞ i po započtení

Cache-oblivious: Hledání + přerodělování prvků jsou 2 proložené scény (popředu + pozpátku) \Rightarrow ~~$\Theta(K/B)$~~ $O(K/B)$ bloků, $O(\log^2 n/B)$ na prvek.

Zpět k C/O strukturám

Insert : $O(\log^2 n)$ času, $O(\log^2 n / B)$ I/O na update OFM
 + $O(\text{#změněných prvků} \cdot \frac{OFM}{B} + \log n / \log B)$ na důležitý update VEB] \rightarrow amortizované
 se schová
 do ceny OFM
 + $\log n / \log B$

Find : $O(\log n)$ času, $O(\log n / B)$ I/O na VEB

Zrychlení \rightarrow jako obvykle indirekce \rightarrow Fragmenty velikosti $F = \Theta(\log n)$
 nad jejich reprezentantů původní struktura

- uvnitř fragmentu vše v čase $O(\log n)$ a $O(\log n / B)$ I/O
- jednou za amort. $O(1/F)$ operací, provedeme $O(1)$ operaci na pův. struktuře
 \rightarrow jedna stojí amort. $O(\log n)$ času a $O(\log n / \log B)$ I/O
 [$\log n / B \approx OFM + \log n / \log B \approx VEB$]
- dotazy : nejprve ve VEB, pak sekvence ^{1/4} fragmentů : $O(\log n)$ času, $O(\log n / \log B)$ I/O
- jednou za čas globální přestavba, abychom udrželi $F = \Theta(\log n)$ apod.

\rightarrow Asymptoticky stejně rychlé jako C/A B-stromy, ale je to C/O.