Co ráno zopakovat:

- BB[α] - strom
  - finální složitost
- dynamické pole
  - finální složitost
- Bloom filtry
  - Maďarův zápis
- Intervalové stromy
  - každý je i-stromy, každý i-strom je uložený podobně, jak to je se složitostí BB[α]
- Multiply-mod-prime / Scalar-mod-prime
  - důkazy 2-nezávislosti, (2,k)-nezávislosti

c-univerzální: $P[h(x)=h(y)] \leq \frac{c}{m}$

k-nezávislý, pokud (k,c)-nezávislý pro $c \geq 1$: $P[h(x_i)=z_i \ \forall i \in k] \leq \frac{c}{m^k}$

c-univerzálnost je tedy speciální případ $(2,c)$-nezávislosti, jelikož

$$P[h(x)=h(y)] = P[\exists z \in M : h(x)=z \wedge h(y)=z] \leq \sum_z P[h(x)=z \wedge h(y)=z] = m \cdot \frac{c}{m^2} = \frac{c}{m}$$

BB[α] - strom

$O(\log(|S_{\ell(u)} - S_{r(u)}|))$

bez rebuildu ve stromě versto $O(\log n)$

pokud nastane rebuild: $\Phi(u) \geq \alpha S_u - (1-\alpha) S_u = \underbrace{(2\alpha-1)}_{\geq 0} S_u$

po rebuildu pokleslo - $\Omega(S_u)$, to zaplatí rebuild

$0 \leq \psi \leq h \cdot h = n \cdot \log n$

h operací: $O((n+h)\log n)$

$x_1, x_2 \in [p]$ libovolné,

$y_1 = ax_1 + b$

$y_2 = ax_2 + b$

pak soustava určuje jednoznačně příslušné mezi (a,b) a (y_1, y_2).

$h_{a,b}(x) = ax + b \bmod p$ → tahový systém je $(2,1)$-nezávislý

$P[h_{a,b}(x) = h_{a,b}(y)] \leq \frac{1}{p^2}$ → takže když volíš a, b, pak je řešení jednoznačné

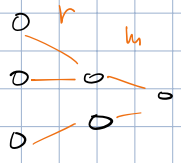
Uvažme $(2,c)$-nezávislý systém, pak skládání modulo m:

$h_{a,b}(x) = (ax + b \bmod p) \bmod m$ je $(2, 4c)$-nezávislý

$2c$-univerzální

$2c$-univerzálnost: chci $h(x_1) = y_1$

$P[h(x_1) = h(x_2)] = \sum_{y_1 \equiv_m y_2} P[h(x_1) = y_1 \wedge h(x_2) = y_2]$



$P[h(x_1) = y_1 \wedge h(x_2) = y_2] \leq \frac{c}{r^2}$

→ y_1 může nabývat r hodnot, y_2 může nabývat $\lceil \frac{r}{m} \rceil$ hodnot → jen ty cyklické se vám tam trefí

$\lceil \frac{r}{m} \rceil \leq \frac{2r}{m}$

$P[h(x_1) \equiv_m h(x_2)] = \frac{c}{r^2} \cdot r \cdot \frac{2r}{m} = \frac{2c}{m}$ → $2c$-univerzálnost

$(2, 4c)$-nezávislost $\frac{4c}{r^2}$

$\leq \frac{c}{r^2}$

$P[h(x_1) = z_1 \wedge h(x_2) = z_2] = \sum_{\substack{y_1 \equiv_m z_1 \\ y_2 \equiv_m z_2}} P[h(x_1) = y_1 \wedge h(x_2) = y_2]$

obr y_i mohou nabýt $\lceil \frac{r}{m} \rceil$ hodnot $= \frac{c}{r^2} \cdot \left(\frac{2r}{m}\right)^2 = \frac{c}{r^2} \cdot \frac{4r^2}{m^2} = \frac{4c}{m^2}$ → $(2, 4c)$-nezávislost

poly-mod-prime   máme zobecnění skládání modulo: (k,c)-nezávislý sys. pokud $r \geq 2km$

$(k,2c)$-nezávislý

pro $x_1 - x_k$, $y_1 - y_k$ z kon. tělesa T

$\exists!\ p(x) = \sum_{i=0}^{k-1} a_i x^i$ stupně $k-1$, že $p(x_i) = y_i\ \forall i$?

je určený těmi k koeficienty a_i

takže je to $(k,1)$-nezávislý

pro $p \geq 2km$ máme $(k,2)$-nezávislost

m ← $h(x)+i$ $i \geq 0$ s minimálním i

Délka nejdelšího řetězce:

Složitost se odvíjí od průměrné vzdálenosti prvku x_i od pozice $h(x_i)$

Pokud mám úplně náhodný systém hash.

a $m/n = \alpha < 1$, tak operace jsou $\varnothing\ \mathcal{O}(1)$

$$1 < c < \frac{1}{\alpha}, \quad q = \left(\frac{e^{c-1}}{c^c}\right)^{\alpha}, \quad 0 < q < 1$$

$P_t = P\left[\left|\{x \in S \mid h(x) \in T\}\right| = t\right]$, X_i – prvek i je v T

$X = \sum_i X_i$, $\mu = \mathbb{E}[X] = t\alpha$ → koeficient zaplnění

$c\alpha < t$

$$P_t = P[X = t] \leq P[X > c\mu] \leq \left(\frac{e^{c-1}}{c^c}\right)^{\mu} = q^{\frac{\mu}{\alpha}} = q^t$$

$$P'_k < \sum_{s=0}^{\infty} P_{k+s} < q^k \cdot \sum_{s=0}^{\infty} q^s = \frac{q^k}{1-q}$$ – až k-tá přihrádka je volná

$$\sum_{k=0}^{m} k \cdot P'_k < \frac{1}{1-q} \sum_{k=0}^{m} k \cdot q^k = \frac{2-q}{(1-q)^3} = \mathcal{O}(1)$$ → očekávaná hodnota té vzdálenosti k

Lock-free programming

- read & write
- test and set
- fetch and set
- compare and swap

↗ se všemi by šlo implementovat mutexy v nějakém jazyce.

→ dokážu dopředu

```
push(node)
  while true:
    h = head
    node.next = h
    if CAS(head, h, node) = h
      return

pop()
  while true
    h = head
    n = h.next
    if CAS(head, h, n) = h
      return
```

problém nekonečného cyklu, ABA problém

| můžu ukládat a porovnávat dvojici, včetně timestampu

dealokace paměti:

```
pop()
  while true:
    h = head
    h.href ++
    if h ≠ head
      h.href --
      continue
    h = h.next
    if CAS(head, h, n) = h
      h.href --
      return
    h.href --
```

Algs. v Cache-aware / Cache-oblivious modelech:

blok velikosti B, kapacita M, celkem $P = M/B$ bloků.

Mám pouze pomalý disk / rychlou cache

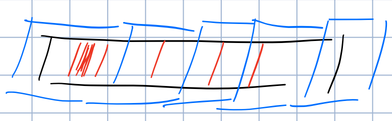
Cache je plně asociativní, procesor čte jen z cache.

$\lceil n/B \rceil + 1$ bloků

a,b-strom → má hloubku $\log_b n$ → pokud $b = B$, tak má vrchol $\leq B$ synů, B klíčů

Celkem načte $O(1 + \log_B n)$ bloků

→ to je cache-aware



provede $\log n$ skoků

posledních $\log B$ je ve dvou blocích max

$O(1 + \log n - \log B) = O(1 + \log \frac{n}{B})$ bloků

Mergesort

$n \leq M/2$ -> přenesu $2\,n/B + 1$ bloků

$n \geq M/2$ $\qquad 2 \leq \frac{M}{2} \leq 2z$

Musím přenést všechno : $+\,2\,n/B$

režie : $+\,2\,n/2$

obecně : $O(1)$

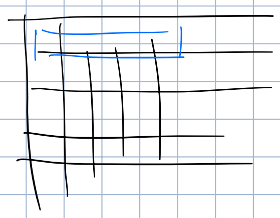
$2\,n/B + 2\,n/2 + O(1) \sim O\left(\frac{n}{B} \log \frac{n}{M}\right)$ ↗ analýza podle

Transpozice matic: LRU / FIFO

– triviální

```
for i in range(k):
    for j in range(i, n):
        swap(A_ij, A_ji)
```

pokud $k > B$



potřebuji alespoň $\Omega(k^2)$ přenosů -> vždy si těsně před transpozicí vyměním ten daný blok, ve kterém je sloupec / řádek

OPT: $P = M/B$ – jeden blok per řádek, tudíž nanejvýš jeden sloupec tam taky může mít uložený

transpozice i-tého řádku / sloupce zachová max $\max(0, k - P - i)$

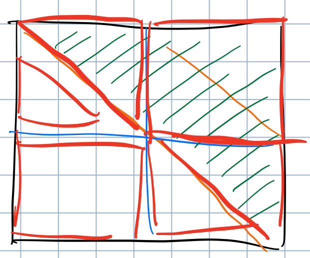
$$\sum_{i=1}^{k-P} k - P - i = (k - P)^2$$

Triviální cache-aware -> $O(k^2/B)$

$2 = B$, submatici 2x2 transponuji s $O(B)$ přenosy.

Skáču po jednotlivých subblocích:

```
for i in range(0, k, 2):
    for j in range(i, k, 2):
        for ii in range(i, min(k, i+2)):
            for jj in range(max(j, ii+1), min(k, j+2)):
                swap(A_ii,jj ; A_jj,ii)
```

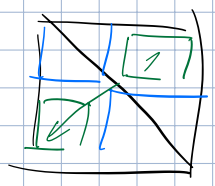


Mám celkem $(k/B)^2$ submatic, každou za $O(B)$ přenosů, $O\left(\frac{k^2}{B}\right)$ přenosů

Cache-oblivious:

Transpose and swap

$M \geq 4B^2$ → tall cache



$2 \leq B \leq 2z$ → co se mi vejde do cache

2×2 submatice řeší $v \leq 2z$ blocích

TransposeAndSwap potřebuje $4z$ přenosů

Máme $\left(\frac{h}{2}\right)^2$ submatic, $4z$ přenosů na jednu

$B \leq 2z$

Celkem máme $\frac{h^2}{2^2} \cdot 4z$ přenosů $= \frac{h^2}{2^2} \cdot 4z = \frac{4h^2}{2} < \frac{8h^2}{B} = O\left(\frac{h^2}{B}\right)$

takže je to asi na konstantu optimální.

Sleator-Trojan věta

P_{LRU}, P_{OPT} → počet bloků

F_{LRU}, F_{OPT} → počet přenosů

$$F_{LRU} \leq \frac{P_{LRU}}{P_{LRU} - P_{OPT}} \cdot F_{OPT} + P_{OPT}$$

$P_{LRU} > P_{OPT}$

důkaz:

Pozorování:

pokud LRU potřebovalo P_{LRU} bloků, tak je tam P_{LRU} různých bloků. Tudíž OPT potřebuje alespoň $P_{LRU} - P_{OPT}$ přenosů.

Rozdělíme posloupnost přístupů na podposloupnosti

$$F'_{LRU} \leq \frac{P_{LRU}}{P_{LRU} - P_{OPT}} \cdot F'_{OPT}$$

$$\frac{F'_{LRU}}{F'_{OPT}} \leq \frac{P_{LRU}}{P_{LRU} - P_{OPT}}$$

hrnu poslední v poslední + P_{OPT}

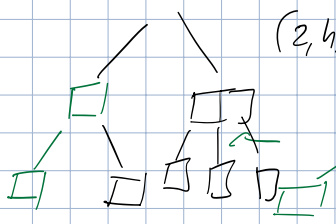
$$1 \leq \frac{P_{LRU}}{P_{LRU} - P_{OPT}}$$

$F''_{OPT} \geq F''_{LRU} - P_{OPT}$

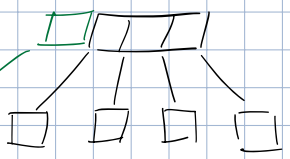
$$F''_{LRU} \leq F''_{OPT} + P_{OPT} \leq \frac{P_{LRU}}{P_{LRU} - P_{OPT}} \cdot F''_{OPT} + P_{OPT}$$

Takle bych chtěl dostávat pls pls pls

Definujte (a,b)-strom, červeno-černý, srovnejte.



$(2,4)$-strom

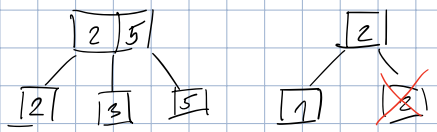


vyhledávací strom, $a \geq 2$, $b \leq 2a-1$

vnitřní vrchol alespoň a synů, nanejvýš b

root nanejvýš b synů

ve vrcholu mám klíče, které mě vedějí do podstromů



hloubka stromu h: $\log_b n \leq h \leq \log_a n$

Find mě stojí $\log b$ na vrstvě, celkem $\log_a n$ vrstev

Insert/Delete...

$$\log b \cdot \log_a n = \log b \cdot \frac{\log n}{\log a} = \log_a \cdot \frac{\log b}{\log a} \quad b = poly(a) \quad = \log n$$

Amortizace: insert/delete: potřebuji b času na spojení/rozdělení.

Můžu rozdělovat potenciálně všude

$$b \cdot \log_a n = b \cdot \frac{\log n}{\log a} = \underline{\boxed{\log_n \frac{b}{\log a}}}$$

listy jsou červení

otec červeného je černý

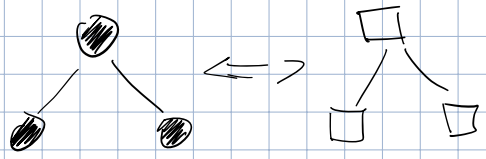
všechny cesty vrchol–list mají stejný počet černých

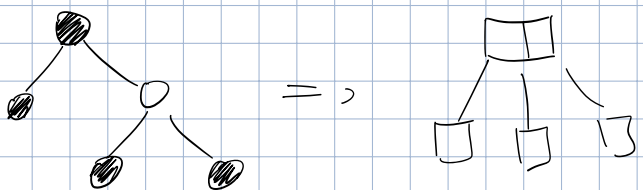
Je to $(2,4)$-strom

složitost je tedy zase logaritmická,

vhodný dělat mapping do $(2,4)$-stromu

pro pochopení.





Separované řetězce:

Každá přihrádka obsahuje pole, do nějž přidáváme podle hashovací fce

všechny tři operace trvají tak dlouho, jako je průměrná délka řetězce. $\to O(1)$

Mám-li c-universální H systém, v jedné přihrádce bude $\leq {cn}/{m}$ prvků.

pokud $m = \Omega(n)$, operace jsou $O(1)$.

Potřeba dokázat, že bude opravdu $O(1)$ očekávaná největší délka řetězce,

mám-li dynamický systém

$n/2 \leq m \leq n$ $\to$ měním faktorem 2. Amortizovaná slož. je $O(1)$.

$P[E_n] \geq 1 - \frac{1}{n^c}$, $c > 1$ s velkou pravděpodobností

My ukážeme skoro velkou pravděpodobnost

Nechť $m = \Theta(n)$, pak $\max_j A_j = \Theta\left(\frac{\log n}{\log\log n}\right)$

A_j = počet prvků v j-té přihrádce

$$P\left[\max_j A_j \leq (1+\varepsilon)\cdot\frac{\log n}{\log\log n}\right] > 1 - \frac{1}{n^{\frac{\varepsilon}{3}}} \quad \text{pro } \varepsilon > 0$$

úplně náhodný systém

$\mu = \mathbb{E}[A_1] = n/m$

$c = (1+\varepsilon)\frac{\log n}{\log\log n}$

$$P[\max_j A_j \geq c\mu] = P[\exists_j : A_j > c\mu] \leq \sum_j P[A_j > c\mu] = m \cdot P[A_1 > c\mu]$$

$$P[A_1 > c\mu] \leq \left(\frac{e^{c-1}}{c^c}\right)^{\mu}$$

$$m \cdot \frac{e^{(c-1)\cdot\mu}}{c^{c\mu}} = m \cdot \frac{e^{c\mu}\cdot e^{-\mu}}{c^{c\mu}} = me^{-\mu}\cdot e^{c\mu - c\mu\log c} = \dots = 1 - \frac{1}{n^{\frac{\varepsilon}{3}}}$$

$$c^{c\mu} = e^{\log c^{c\mu}} = e^{c\mu\log c}$$

Hashování řetězců:

Chci zahashovat d-tici $x_1 \dots x_d \in \mathbb{Z}_p$, p je prime

Scalar-mod-prime

$\sum_i^d a_i x_i \bmod p$ → 1-universální $1/p$ $\quad a \in \mathbb{Z}_p^d$

$(b + \sum_i^d a_i x_i) \bmod p$ → (2,1)-nezávislé $\quad b \in \mathbb{Z}_p$

$((b + \sum_i^d a_i x_i) \bmod p) \bmod m$ → (2,4)-nezávislé

Poly-mod-prime

$\sum_{i=0}^{d-1} x_{i+1} a^i \bmod p$ → d-universální ?

polynom stupně d-1 je jednoznačně určen d body

$(\sum_{i=0}^{d-1} x_{i+1} a^i \bmod p) \bmod m$ → d-universální

$(\sum_{i=0}^{d-1} x_{i+1} a^i \bmod p) \bmod m$

$$P[h_{a,b,c}(x_1 \dots x_d) = h_{a,b,c}(x_1' \dots x_{d'}')] \le 2/m$$ pro řetězce délek $d, d' \le p/m$

Hashování separovaných řetězců

H - úplně náhodný systém, počet prvků v přihrádce je cn/m.
Pokud $m = \Omega(n)$, pak všechny operace jsou $O(1)$. ?

Šance, že prvek spadne do stejné přihrádky, je c/m.

Mám celkem n prvků...

Uvažme dynamický systém, kde tabulku prodloužíme a zmenšíme faktorem dva, pokud neplatí $n/4 \le m \le n$. Amortizovaná složitost tohoto systému je $O(1)$.

My chceme se dnes úplně přesně ukázat, že $P[\max_j A_j \le (1+\varepsilon) \cdot \frac{\log n}{\log \log n}] > 1 - \frac{1}{n^{\varepsilon/3}}$

$c = (1+\varepsilon)\frac{\log n}{\log\log n}$ nechť mám úplně náhodný systém

$\mu = \mathbb{E}[A_1] = n/m$ $\quad P[\max_j A_j > \mu c] = P[\exists j : A_j > \mu c] \le \sum_j P[A_j > \mu c] = m \cdot P[A_1 > \mu c]$

$\varepsilon > 0$

$$P[A_1 > \mu c] \le \left(\frac{e^{c-1}}{c^c}\right)^\mu = \frac{e^{(c-1)\mu}}{c^{c\mu}} \overset{?}{\le} \frac{e^{c\mu}\cdot e^{-\mu}}{e^{c\mu\log c}} = m e^{-\mu}\cdot e^{c\mu - c\mu\log c} \dots \le \frac{1}{n^{\varepsilon/2}}$$

$c\mu\log c$

$$P[A_1 \le \mu c] \ge 1 - \frac{1}{n^{\varepsilon/2}}$$

Scalar-mod-prime:

$x_1 - x_c \in \mathbb{Z}_p$, $a \in \mathbb{Z}_p^d$

$\sum_i^d a_i x_i \bmod p$ → 1-univerzální

$b + \sum_i^d a_i x_i \bmod p$ → $(2,1)$-nezávislý (dvě volby s prvků $1/p$)

$\left(b + \sum_i^d a_i x_i \bmod p\right) \bmod m$ → $(2,4)$-nezávislý (vychází z předchozího a předpokladu o skládání modulů)

Poly-mod-prime

$\sum_{i=0}^{d-1} x_{i+1} a^i \bmod p$ → d-univerzální

$\left(b + c\cdot\sum_{i=0}^{d-1} x_{i+1} a^i \bmod p\right) \bmod m$

$P\left[h_{a,b,c}(x_1 - x_2) = h_{a,b,c}(x_1' - x_2')\right] \le 1/m$ pro různé vektory délek $d, d' \le p/m$

Důkaz 1-univerzality: $P[h_a(x) = h_a(y)] = P[a\cdot x \equiv_p a\cdot y] = P[a(x-y) \equiv_p 0] = P\left[a_1 \equiv_p \frac{\sum_{i=2}^{d}(x_i - y_i)}{(x_1 - y_1)}\right] = \frac{1}{p}$

↳ mám p voleb a_1

BB[α] - stromy

$\frac{1}{2} < \alpha < 1$

BST

každý podstrom syna může mít nanejvýš αs_n vrcholů.

Pokud se podmínka poruší, provedu rebuild celého podstromu.

Find trvá $O(\log n)$ ⟶ výšku máme $O(\log n)$

Amortizovaná složitost Insertu a deletu je $O(\log n)$

Ukážeme potenciálem. Rebuild stojí $\Omega(s_n)$

$$\phi(n) = \begin{cases} 0 & \text{pokud } \ldots \leq 1 \\ |s_{r(n)} - s_{\ell(n)}| & \text{jinak} \end{cases}$$

Po rebuildu je potenciál daného vrcholu 0.

Operace může potenciál zvýšit nejvýše o dva. 2

Celkem tedy v celém stromě o $O(\log n)$

Rebuild nastává: $\phi(n) \geq \alpha s_n - (1-\alpha) s_n > (2\alpha - 1) s_n \geq 0$

Ukládání do dynamického pole

pole velikosti m, zvětšující/zmenšující faktorem 2, přidávám jako na zásobník

$$\frac{n}{4} \leq p \leq n$$

Ukážeme, že amortizovaná složitost je $O(1)$.

Určíme potenciál

$$\begin{cases} 0 & p = 2n \\ n & p = n \\ n & p = 4n \end{cases} \quad \text{lineární vložení} \quad \begin{cases} 2n - p & p \leq 2n \\ p/2 - n & p > 2n \end{cases}$$

tedy se mi zvedne o 2

Potenciál se může nejvýše zvýšit jednou operací o 2. $\phi' - \phi \leq 2$

pro počet zkopírovaných prvků T platí: $T + (\phi' - \phi) \leq 2$ $\quad$ $T = 0$ bez realokace, jinak $T = \phi$

Nechť provedu k operací: Pak mám $2k + \phi_0 - \phi_k \leq 2k + n_0$ $\to O(n_0 + k)$, tedy $O(1)$ na operaci

BB[α] stromy

α … S_n vrcholů

$\frac{1}{2} < \alpha < 1$

hloubka stromu je $O(\log n) \to \alpha^i n \geq 1$ pouze pro $i \leq \log_{\frac{1}{\alpha}} n$

$\Phi(u) \begin{cases} 0 & \text{pokud } \dots \leq 1 \\ |S_{l(u)} - S_{r(u)}| & \text{jinak} \end{cases}$

bez rebuildu se mi potenciál ve vrcholu zvětší nejvíc o 2.
↪ v celém stromu tak stoupne potenciál max. o $\log n$.

Pokud musí rebuild, tak $\Phi(u) \geq \alpha S_u + (1-\alpha) S_u = \underbrace{(2\alpha - 1)}_{\geq 0} S_u$ $\quad \frac{1}{2} < \alpha$

Po rebuildu mi klesne potenciál na 0, zmenší se o $\Omega(S_u)$ a tím
amortizuji ty rebuildy.

Pro h-operaci mám $O((h+n)\log n)$ čas