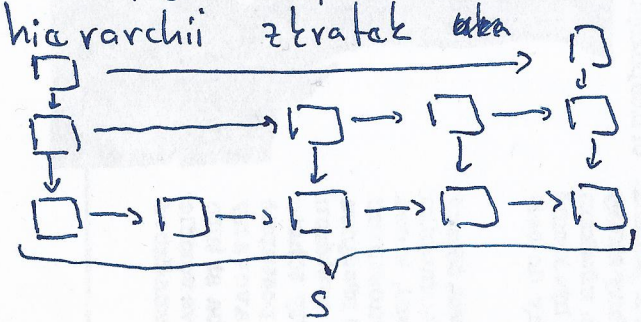
Randomizované stromy

Skip list [Pugh'89]

- cíl: setřízený spoják nad S
- idea: nad spojákem postavíme hierarchii zkratek



Leveling - je $S = S_0 \supseteq S_1 \supseteq \dots \supseteq S_r$, $r = \#$levelů

Struktura

- $\forall S_i$ si udržujeme jako setřízený spoják + zarážky $\pm\infty$
- $k^i \rightsquigarrow$ kopie klíče k v levelu i
- pointery z k^i do k^{i-1}

Konstrukce levelingu

- $\forall k \in S$ si (nezávisle) hodí korunou $(p = 1/2)$
  - úspěch $\rightsquigarrow k \in S_1$ a pokračuju pro další S_i
  - neúspěch $\rightsquigarrow k \notin S_1$ a končím
- Z_k = v kolika levelech je k (mimo S_0)

$\rightsquigarrow r = \max_{k \in S} Z_k$ $\qquad Z = \sum Z_k$ ↳ velikost SL (bez S_0)

Základní vlastnosti

- $Pr[Z_k = t] = 1/2^{t+1}$ $\qquad E[Z_k] = 2$

  $-\,''\,- \geq t] = 1/2^t$ $\qquad Z_k \rightsquigarrow$ geom. distribuce s $p = 1/2$

- Z je negative binomial distrib.
  → házím korunou, dokud mi nepadne n úspěchů a počítám úspěchy
- průměrná hloubka
  - $\rightarrow Pr[r \geq t] \leq \sum_k Pr[Z_k \geq t] = n/2^t$
  - $\Rightarrow Pr[r \geq c \cdot \log_2 n] \leq \frac{1}{n^{c-1}}$
  - $\rightarrow E[r] = \sum_{t=0}^{\infty} Pr[r \geq t] \leq 1 \cdot c \cdot \log n + \overbrace{\sum_{t=c\cdot\log n}^{\infty} Pr[r \geq t]}^{\text{geom. řada}}$

    $= c \cdot \log n + 2 \cdot \frac{1}{n^{c-1}} \underset{c=1}{=} \log n + 2$
- průměrná velikost
  - $\rightsquigarrow E[Z] = \sum E[Z_k] = 2n$
  - → dokonce s velkou pstí (jinak lze třeba $(1 - \frac{1}{n}) \cdot n + \frac{1}{n} \cdot n^2 = 2n - 1$)
- $Pr[Z > 4n] \leq ?$

  $Z > 4n \Leftrightarrow$ hodím $4n$ korunami a padne mi $< n$ úspěchů

  $\rightsquigarrow X = \sum_{i=1}^{4n} X_i$

  $Pr[Z > 4n] = Pr[X < n] \leq e^{-\frac{n}{2}}$

  Černov $\mu = E[X]$

  $Pr[X < (1-\varepsilon) \cdot \mu] \leq e^{-\frac{\mu \cdot \varepsilon^2}{2}}$

Operace

Search(x) ⇝ intuitivně

Delete(x) ⇝ Search(x) + cestou mažu výskyty x

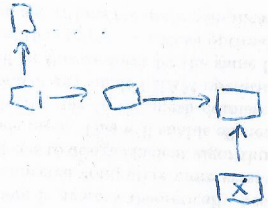
Insert(x) ⇝ předem si naházím, na kterém levelu budou kopie x; Search(x) + od chodného levelu začnu x ukládat

👁 jednotlivé klíče si hází výskyty nezávisle
⇒ ins a del neovlivní analýzu výšky/velikosti

Analýza Search(x)

- analyzuju search⁻¹ od x do vstupu nahoře (BÚNO otočíme směr pointerů)

→ začnu v x
→ pointer nahoru ⇒ jdu nahoru
→ jinak doleva



? kolik kroků "→" udělám?

→ v ∀uzlu je s pstí 1/2 pointer ↑ a s pstí 1/2 musím →

→ intuitivně: na každé vrstvě v průměru 2 kroky →
a mám v průměru $O(\log n)$ vrstev

– problém je že prostě nemůžu říct

$$E\left[\sum_{i=1}^{r} Y_i\right] = \sum_{i=1}^{E[r]} E[Y_i] \qquad (Y_i = \#\to \text{ na vrstvě } i)$$

- $U = \#\to$ kroků

$$E[U] = \underbrace{E[U \mid r \le 2\log n]}_{\le E[U \mid r \le 2\log n] \le 1} \cdot \Pr[r \le 2\log n] + \underbrace{E[U \mid r > 2\log n]}_{\le n \text{ neb } U \le n} \cdot \underbrace{\Pr[r > 2\log n]}_{\le 1/n}$$

$$\cancel{\le \sum_{i=1}^{2\log n} E[Y_i] + 1 \le 4\log n + 1}$$

~~odhadnu Y_i geom. dist. s $p = 1/2$~~

⇒ Search je v průměru $\le O(\log n)$
(a lze i s velkou pstí jako u velikosti)

Random binary search tree

- vezmeme T BST vytvořený inserty z náhodné posloupnosti

? jak bude T hluboký?
→ $O(\log n)$ v průměru (a s velkou pstí)

! toto není strom vybraný uniformně ze všech n-vrcholových BST

👁 T přesně odpovídá stromu volání quicksortu na náhodné pstí, ve vrcholech jsou pivoti

Důsledek $\sum_{v \in T} \text{hloubka}(v) = \Theta(n \log n)$
⇝ pokud vyberu náhodné x, tak v průměru má hloubku $\Theta(\log n)$

Důsledek x pevné, $E[\text{hloubka}(x)] = \Theta(\log n)$
↳ plyne z analýzy QuickSelectu
(jsem v y též y přidat x, potom pod y je v průměru $\le 3/4$ prvků co pod p(y))

$E[U \mid r \le 2\cdot\log n]$ můžeme shora odhadnout náhodnou procházkou ←, ↑, která vyleze právě $2\log n$ levelů ← ignorujeme levý okraj listu

$$\le \sum_{i=1}^{2\log n} E[\text{geom}(1/2) - 1] = 2\log n$$

na každé vrstvě házím kornou, dokud mi nepadne ↑ ≈ geom. distribuce bez posledního kroku

Randomizad binary search tree (RBST) [Martinez Roura '98]

idea: náhodný BST má log hloubku
⇒ místo vyvažování udržujeme strom náhodný

T je náhodný strom na [n] ⟺
$\forall i \in [n]\ Pr[i \text{ je kořen}] = \frac{1}{n}$ & levý i pravý podstrom kořene jsou náhodné stromy na daných prvcích
↳ podmínka R

Přesněji řečeno: $\forall T$ BST na n prvcích:

$$Pr[\text{z distribuce náh. stromů dostanu } T] = Pr[\text{z distribuce stromů dané R dostanu } T]$$

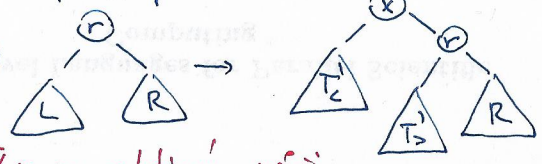
~~(plyne z ekvivalence s quicksortem)~~
všechny posl. na [n] zkonstruuju: $\forall i \in [n]$ rekurzivně vytvořím všechny posl. na $\{1 \dots i-1\}$ a $\{i+1 \dots n\}$ a prodlouží je všemi způsoby
⇒ každou posl. dostanu právě 1-krát

RBST je BST splňující podmínku R

Updaty vyvažují tak, aby zachovali R, což garantuje průměrnou hloubku (průměrujem přes náš náhodný gen.)

Insert (x)

- hoď si kostkou a s pstí $1/n+1$ vlož x jako kořen, jinak rekurze do správného podstromu
  ⇝ musím si pamatovat velikosti podstromů
- vložení x jako kořen:
  → $split(T,x) = T_<, T_>$ ⇝ levý a pravý podstrom x
  ↳ $x \overset{?}{<} root(T) \overset{def}{=} r$
  → $T_<', T_>' = split(L,x)$
  → $T_< = T_<'$
  $T_> = T_>' + r + R$



! r je relativní vůči úrovni rekurze

Analýza insert(x)

$T \rightsquigarrow T'$

- $Pr[y = root(T')] = Pr[y = root(T)] \cdot Pr[x \neq root(T')]$
  $y \neq x$ pevné $= \frac{1}{n} \cdot \left(1 - \frac{1}{n+1}\right) = \frac{1}{n+1}$

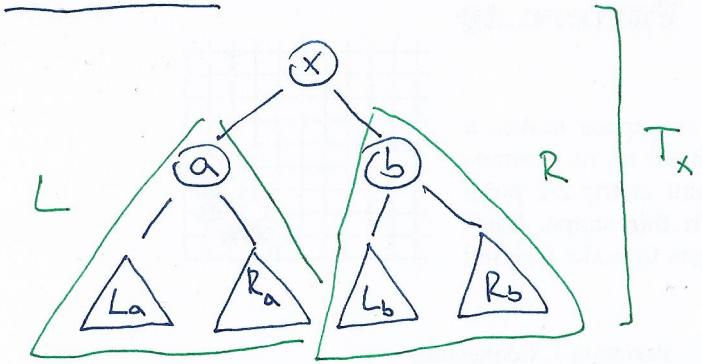
- $T_<, T_>$ jsou nezávislé RBST
  → $T_<$ podstrom T ⇒ ✓
  → $T_>$ indukcí
  → rozdělím $Pr[z = root(T_>)]$ podle $root(T) \overset{?}{<} x$
  ⇝ $Pr[z = root(T_>) \mid r > x]$ $\quad m = |T_>|$
  $= \underbrace{Pr[r > x]^{-1}}_{\text{mám m kandidátů}} \cdot \underbrace{Pr[z = root(T_>) \ \&\ r > x]}_{\text{to nastane} \iff z = r}$
  $= n/m \cdot \frac{1}{n} = \frac{1}{m}$

r je relativní dle úrovně indukce

Formálněji:
→ chci pro $\forall$ podstrom U stromu T dokázat, že je RBST (plus umáchat nezávislost)
→ $pr[y = root(U)]$ rozdělím na
- insert se U nedotkl
- x prošlo kořenem U
- U vzniklo v rámci joinu

→ ukážeme, že všechny dávají stejnou pstí takže nebudu řešit, jakou pstí nastávají

Delete (x)



→ T_x nahradím join(L, R)

Join(R, L) = J $l = |L|, r = |R|$

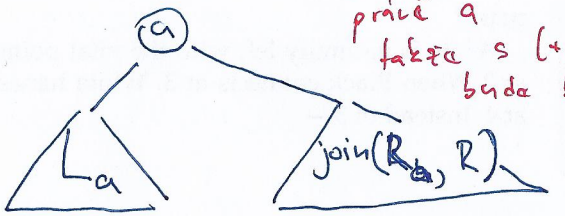
→ s pstí $l/(r+l)$ je root(J) = a, jinak b

→ dále rekurze → ~ s $\frac{1}{l}$ je kořen L

právě a

takže s $l+r$

bude kořen J



Join(L, R) vytvoří validní RBST

~ triv, indukce

$T' = del(T, x)$ je validní RBST

! změnil se # prvků ⇒ musí se změnit pst

→ $y \neq x$: $Pr[y = root(T')] = Pr[y = root(T)]$

$+ Pr[x = root(T)$ & y jsme vybrali jako nový kořen$]$

Analýza insert(x)

$T \to T'$

• představa: - mám strom vzniklý insertem

- u každého vrcholu původního stromu jsem měl pravděpodobnost, s jakou tam ten vrchol skončil

⇝ úplný kořen $1/n$ atd.

- teď musím ty psti přepočítat, aby sedělo

případy: dle vrcholu y

i) insert se podstromu s y nedotkl ⇝ ☺

ii) x prošlo skrz y: (BÚNO y = root(T))

$Pr[y = root(T')] = Pr[y = root(T)$ & x "nenahradilo" $y] = \frac{1}{n} \cdot \frac{n}{n+1}$

$= 1/(n+1)$ ☺

iii) $y = x \Rightarrow$ ☺

iv) y "prošlo" joinem ⇝ $y = root(T_>)$ ≈ y pod x

↳ BÚNO $r > x$ a $y = root(T_>)$ jinak jde o podstrom T

→ indukcí dokážu, že $T_<, T_>$ jsou RBST

↳ $T_<$ i $T_>$ závisí na tom, jestli $r \gtrless x$, ale r je náhodné

⇝ počítáme $Pr[y = root(T_>) \mid r < x]$

$= \underbrace{Pr[y = root(T_>) \text{ \& } r < x]}_{y = r} \cdot Pr[\underbrace{r < x}_{m \text{ kandidátů}}]^{-1} =$

$= 1/n \cdot (m/n)^{-1} = \frac{1}{m}$

(případně lze umáchat rukama přes quicksort)

$= \frac{1}{n} + \frac{1}{n} \cdot \underbrace{\frac{1}{n-1}}$

$T' = join(L, R)$ a o výsledku joinu víme, že je RBST

Treapy

[Seidel & Aragon '96]

- BST + halda
- ∀ klíč má náhodnou prioritu (pevná univerzum) ↳ p_k
- podle klíčů BST + podle priorit max. halda

⚠ různé priority ⇒ jednoznačný tvar stromu

(všude budeme předpokládat různé priority)

⚠ priority jsou uniformní (třeba z [0,1]) ⇒ treap je náhodný strom

↳ stačí, aby všechny klíče používaly stejnou distribuci

vážené priority

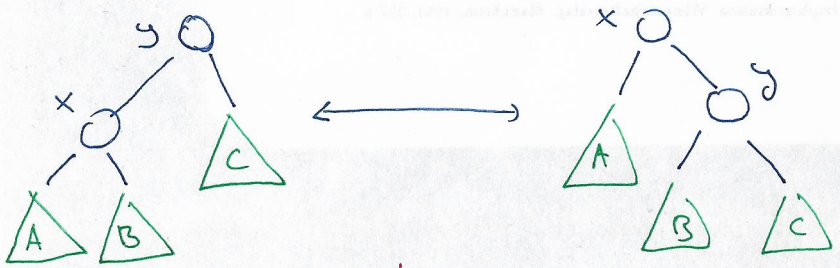
→ ∀ klíč k má váhu $w_k \in \mathbb{N}$

→ $p_k = \max\{p_k^{(i)} \text{ náhodné} \mid 1 \le i \le w_k\}$

↳ libovolná (spojitá) distribuce

Operace

⚠ haldové bublání můžu dělat rotacemi



$p_x > p_y$, jinak haldové usp. platí

Insert(x) ⇝ klasický + bublám

Delete(x) ⇝ nastavím $p_x = 0$, zabublám, smažu list

Split(x) ⇝ vložím x s $p_x = \infty$ ⇝ vezmu podstr. kořene

Join(T_1, T_2) ⇝ invariant ↑ vložím T_1, T_2 pod kořen x a pak smažu x

Složitost (průměrná)

$W = \sum w_x$

	nevážený	vážený
přístup	$O(\log n)$	$O(\log \frac{W}{w(x)})$
insert/delete	$O(\log n)$	$O(\log(\frac{W}{w_x} / (\min(w_x, w_{x^+}, w_{x^-}))))$
insert/del s prstikem	$O(1)$	$O(\log(\frac{w_x}{w_{x^-}} + \frac{w_x}{w_{x^+}} + \frac{w_{x^-}}{w_{x^+}}))$
# rotací	2	— " —

x^+, x^-: předchůdce/následník

Pozn. Složitost ↑ funguje pro nevážený případ i se slabší "náhodností" priorit (z 8-nezávislé z dost velkého univerza)

↳ stačí rozumný pseudonáhodný gen plus náhodný seed

Analýza (navážená)

- klíče: x_i tž $x_1 < x_2 \dots < x_n$

  $p_i \equiv p_{x_i}$, w_i analogicky

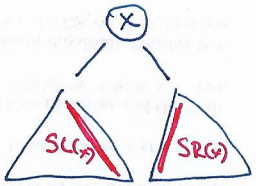
- $D(x)$ = hloubka x $\qquad$ $S(x)$ = # vrcholů pod x $(+x)$

  $P(x,y)$ = délka cesty $x \leftrightarrow y$

  $SL(x), SR(x)$ = pravá páteř levého podstromu x

  levá — " — pravého — " —

  levá páteř = nejlevější cesta



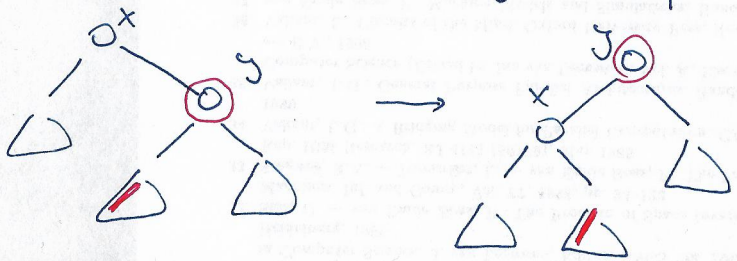
- průměrný čas na find/insert/delete = $E[D(x)]$

  (resp. místo x předchůdce/následník)

- operace s prstíky $\rightsquigarrow P(x,y)$, $y \in \{x^-, x^+, \dots\}$

- # rotací při ins(x)/del(x) $\le SL(x) + SR(x)$

  -> levá rotace "dolů" ztrácejí levou páteř pravého podstr.



=> # levých rotací $\le SL(x)$

$A_{i,j}$ = indikátor x_i předek x_j $\qquad a_{i,j} = E[A_{i,j}]$

$C_{i,l,m}$ = — " — x_i je společný předek x_l, x_m $\qquad c_{i,l,m} = E[C_{i,l,m}]$

Věta $l < m$:

i) $D(x_l) = \sum_{i=1}^{n} A_{i,l}$ $\qquad$ ii) $S(x_l) = \sum_{j=1}^{n} A_{l,j}$

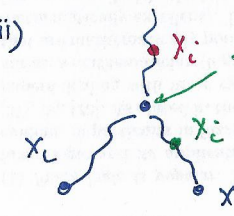
iii) $P(x_l, x_m) = 1 + \sum_{i=1}^{m-1} (A_{i,l} - C_{i,l,m}) + \sum_{i=l+1}^{n} (A_{i,m} - C_{i,l,m})$

iv) $SL(x_l) = \sum_{i=1}^{l-1} (A_{i,l-1} - C_{i,l-1,l})$

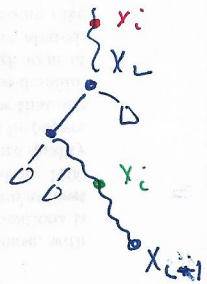
$SR(x_l) = \sum_{i=l+1}^{n} (A_{i,l+1} - C_{i,l,l+1})$

Důkaz i), ii) triv.

iii)



iv)



pro $i < l$ x_i je předek $x_l \Leftrightarrow$ ~~spolup~~ $C_{i,l,m} = 1$

=> stačí nám spočítat $Pr[x_i$ je předek $x_j]$

resp $Pr[x_i$ je spol. před. $x_l, x_m]$

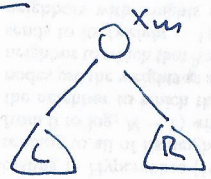
Ancestor lemma

x_i je předek x_j $\iff$ $p_i = \max_{i \le h \le j} p_h$

↳ BÚNO tenhle směr

Důkaz

x_m kořen treapu $\Rightarrow$ má max prioritu



$x_i = x_m$ nebo $x_j = m$ $\Rightarrow$ ☺

$x_i \in L$ & $x_j \in R$ nebo naopak $\Rightarrow$ ☺

→ rekurze do L a R

≈ všechny prvky mezi x_i a x_j leží v podstromu $LCA(x_i, x_j)$ a já chci $x_i = LCA(x_i, x_j)$

Common ancestor lemma

x_i je CA x_L, x_m, $L < m$ $\iff$

$p_i = \max p_h \mid$ $i \le h \le m$ pro $i \le L < m$

$L \le h \le m$ pro $L \le i \le m$

$L \le h \le i$ pro $L \le m \le i$

→ přímo z AL

! je třeba různá priority

do teď vše platí i náhodně

Důsledek (navázaný)

$a_{ij} = 1/(|i-j|+1)$

↳ když mám náh. prom. $X_1, \dots, X_k$ tažené z stejného i.i.d. tak je každá permutace stejně pravděpodobná

↳ $Pr[X_1 - X_2 > 0] = Pr[X_2 - X_1 > 0]$ neb (X_1, X_2) a (X_2, X_1) jsou z nezávislosti stejné

$$C_{i,l,m} = \begin{cases} 1/(m-i+1) \\ 1/(m-l+1) \\ 1/(i-l+1) \end{cases}$$ ... podle CAL

Dosazením do definic: $l < m$

$H_k = \sum_{i=1}^{k} \frac{1}{k}$

i) $E[D(x_l)] = \underbrace{H_l}_{l \ge i} + \underbrace{H_{n-l+1}}_{l \le i} - 1 = O(\log n)$

ii) $E[S(x_l)] = E[D(x_l)]$ → prům. velikost podstromu pod x_l je stejná jako prům. hloubka x_l ! ale mají dost jinou distro.

iii) $E[P(x_l, x_m)] = O(\log(m-l))$

↳ roztrhnu sumy a tfujtajbl počítání

iv) $E[SL(x_l)] = \sum_{i=1}^{l-1} \left(\frac{1}{l-i} - \underbrace{\frac{1}{l+1-i}}_{\text{3. případ z } C_{i,l,m}} \right) = 1 - \frac{1}{l}$

$E[SR(x_l)] = 1 - 1/(n+1-l)$

Pozn. Použitím vhodného Černova na $D(x)$ dostaneme, že $Pr[D(x) > c \cdot \log n] < n^{-d}$, $d > 1$

⇝ s velkou psí je hloubka stromu $O(\log)$