ab_experiment: časté chyby

- někteří jste si nevšimli, že $(a,2a)$-stromy mají konst. amort. počet strukt. změn při insertu a deletu.
  - to je rozepsané ve skriptech, sekce 3.2 v kapitole o $(a,b)$-stromech

---

# HASHOVÁNÍ

Značení

- $m$ ... velikost hashovací tabulky,
- $\mathcal{U} = \{0, \dots, U\}$ ... universum. Tedy prvky, jejichž podmnožinu $S$ si chceme pamatovat.
- $h, h_1, h_2, \dots$ ... hashovací fce. Dnes to bude taková fce, že $f(x)$ je uniformní náhodná hodnota z $\{0, \dots, m-1\}$.

  Jednodušše se analyzuje :)

- pokud pro $x, y \in \mathcal{U}$ platí $h(x) = h(y)$, pak nastala kolize.

---

- pokud útočník zná hashovací fci, pak lze vynutit, že operace find / insert / delete budou mít složitost $\Omega(n)$.
- proto volíme hashovací fce náhodně
  - „pokud ani já nevím, co dělám, těžko to bude vědět útočník" :)

- takže například $Pr[h(x)=h(y)]$ ... co je náhodného? Ta hashovací fce.
- úplný zápis by byl $\Pr_{h \in \mathcal{H}}[h(x)=h(y)]$, kde $\mathcal{H}$ je mn. fcí z $\mathcal{U}$ do $\{0,\dots,m-1\}$

  $h \in \mathcal{H}$ ← vdíme uniformně náhodně.

---

Př: Hashujeme $n$ prvků do pole velikosti $m := n^2$. Jaká je pst., že nastane kolize? (Chceme určit nějakou rozumnou horní mez, tj. ne $\leq 1$)

- $Pr[\underbrace{\exists i,j \in \{1,\dots,n\} \text{ t.ž. } h(x_i)=h(x_j)=k}_{X_k :=}] \leq \binom{n}{2} \cdot \left(\frac{1}{n^2}\right)^2 \leq \frac{n^2}{2} \cdot \frac{1}{n^4} \leq \frac{1}{2} \frac{1}{n^2}$

  v té inkluzi exkluzi každý − je větší než následující +

- $Pr[\exists l : X_l] \leq n^2 \cdot \frac{1}{2} \cdot n^2 \leq \frac{1}{2}$

  union bound přes přihrádky

- tedy pst., že nastane kolize je nejvýše $\frac{1}{2}$

---

Př: (Narozeninový paradox) Hashujeme do tabulky velikosti $m$. Kolik je potřeba zahashovat prvků, aby nastala aspoň jedna kolize s pstí aspoň $\frac{1}{2}$?

- $p_k$ ... nenastane kolize, hashujeme-li $k$ prvků

- $p_1 = 1$
- $p_2 = 1 \cdot \underbrace{\left(1 - \frac{1}{m}\right)}$

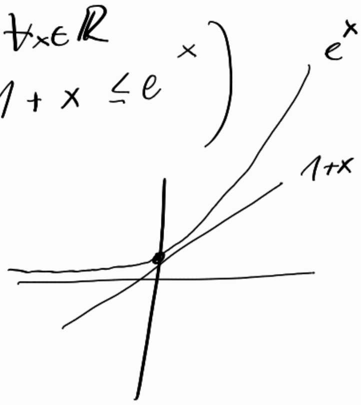
  druhý prvek se smí zahashovat kamkoliv kromě místa, kam se zahashoval první

- $p_3 = 1 \cdot \left(1-\frac{1}{m}\right)\left(1-\frac{2}{m}\right)$

  ⋮

• $p_k = 1\left(1-\frac{1}{m}\right)\cdots\left(1-\frac{k-1}{m}\right)$

$\left(\forall x\in\mathbb{R} \quad 1+x \le e^{x}\right)$




• $p_k \le 1\cdot e^{-\frac{1}{m}}\cdot e^{-\frac{2}{m}}\cdot\ldots\cdot e^{-\frac{k-1}{m}} = e^{-\frac{1}{2m}k^2}$

• při jakém $k$ platí $e^{\frac{k^2}{2m}} \le \frac{1}{2}$ ?

• pro $k=\sqrt{2m}$ : $e^{-1} = \frac{1}{e} \le \frac{1}{2}$

→ hashujeme-li $\sqrt{2m}$ prvků, pak pst kolize je aspoň $\frac{1}{2}$

---

• Srovnejme s předchozí úlohou

• tam jsme taktéž hashovali cca $\sqrt{m}$ prvků, ale pst kolize byla nejvýše $\frac{1}{2}$ ??!

• dost brutálně jsem ignoroval konstanty ... ty to dost ovlivní

• a celkově jsem počítal jako 

<u>Cv</u>: Mějme hash tabulku velikosti $m$. Ukažte, že existuje konst. $c_1$ t.ž. pst, že nenastane kolize při hashování $c_1\sqrt{m}$ prvků, je nejvýše $\frac{1}{e}$. Podobně ukažte, že existuje konst. $c_2$ t.ž. pst, že nenastane kolize při hashování $c_2\sqrt{m}$ prvků, je aspoň $\frac{1}{2}$.

- Pokuste se minimalizovat $|c_1 - c_2|$.
- V nějakém kroku výpočtu by se mohlo hodit uvažovat $m$ dost velké.
- Mohlo by se hodit vědět, že $e^{-x-x^2} \leq 1-x$ pro $x \in [0, \frac{1}{2}]$
- Tuhle úlohu můžete najít v

  M. Mitzenmacher, E. Upfal: Probability and Computing

  jako cvičení 5.3.

---

KUKAČČÍ HASHOVÁNÍ

- chceme <u>worst-case</u> konstantní find, ne jen v průměrném případě
- princip: máme dvě hashovací fce $h_1, h_2$
- pro $x \in \mathcal{U}$ platí, že v tabulce leží buď na indexu $h_1(x)$ nebo $h_2(x)$ a nikde jinde!
- pseudokód insertu je ve skriptech nebo poznámkách.

<u>Proč to funguje?</u>

<u>Def.:</u> <u>Kukaččí graf $G$</u> má vrcholy $V(G) = \{0, \dots, m-1\}$, tj. indexy tabulky. Pro každý prvek $x$ obsažený v tabulce přidáme hranu $\{h_1(x), h_2(x)\}$, tedy neorientovanou.

?: Kdy selže insert?

Pokud pro vkládaný prvek $y$ platí, že $h_1(y)$ leží na cyklu kukaččího grafu.

**Lem:** Nechť $c > 1$ je konstanta a $\frac{n}{m} \leq \frac{1}{2c}$ ($\frac{n}{m}$ je tzv. faktor zaplnění).

Pro dva vrcholy $s$ a $t$ v kukaččím grafu platí, že pst existence cesty z $s$ do $t$ délky $k$ je nejvýše $\frac{1}{mc^k}$.

**Dk:**

- indukcí podle $k$
- $k = 1$
- union bound přes všech $n$ prvků dává

$$\Pr[\text{existuje } s \leftrightarrow t \text{ cesta délky } 1] \leq 2n \frac{1}{m^2} \leq \frac{1}{mc} \quad \checkmark$$

- indukční krok:
- cesta z $s$ do $t$ délky $k$ existuje, pokud existuje $u$ t.ž. existuje cesta z $s$ do $u$ délky $k-1$, ~~která neprochází přes $t$~~, a hrana z $u$ do $t$.

definice z teorie grafů říká, že se na cestě neopakují vrcholy

$X_{k-1}(s,u) :=$

- $\Pr[\text{existuje } s \to u \text{ cesta délky } k-1] \leq \frac{1}{mc^{k-1}}$ z indukčního předpokladu
- $\Pr[X_1(u,t) \quad | X_{k-1}(s,u)] \leq \frac{1}{mc}$
  - ! vrchol $t$ může být na $s \to u$ cestě, ale pro horní

odhad to může těch $\frac{1}{mc}$ jen snížit.

- $Pr[X_k(s,t)] \leq \sum_{u \in V} \left( Pr[X_1(u,t) | X_{k-1}(s,u)] \, Pr[X_{k-1}(s,u)] \right) \leq$

$\leq m \cdot \frac{1}{mc^{k-1}} \cdot \frac{1}{mc} \leq \frac{1}{mc^k}$ □

Jak dlouho trvá insert?

Lem: Nechť $c > 1$ je konstanta. Střední délka cesty začínající v $h_1$ vkládaného prvku je $O(1)$.

Dk:

- použijme předcházející lemma pro všechny cesty délky $k$ a všechny začátky $s$
- střední délka cesty z $h_1(x) = s$ je nejvýše $m \cdot \sum_{k=1}^{n} k \frac{1}{mc^k} \leq \sum_{k=1}^{\infty} \frac{k}{c^k} =$

všechny vrcholy

$= \frac{c}{(c-1)^2} \in O(1)$

□

Kolikrát rehashujeme?

Lem: Nechť $c > 2$ je konstanta. Střední počet přehashování při vkládání $n$ prvků do prázdné tabulky velikosti $m$ je $O(1)$, pokud $\frac{n}{m} \leq \frac{1}{2c}$

Dk:

- aplikujeme předpředchozí lemma na všechny délky cesty a pro všechna $s = t$.
- $Pr[$ existuje cyklus v grafu $] \leq m \cdot \sum_{k=1}^{\infty} \frac{1}{mc^k} \leq \frac{1}{c-1}$
- pst, že nastane rehash je tedy nejvýše tolik $\left( \frac{1}{c-1} \right)$

- očekávaný počet rehashů je tedy $\le \sum_{k=1}^{\infty} \left(\frac{1}{c-1}\right)^k \le 1 + \frac{c-1}{c-2}$ $\square$

---

- vkládání má tedy konstantní průměrnou amortizovanou pst :)

---

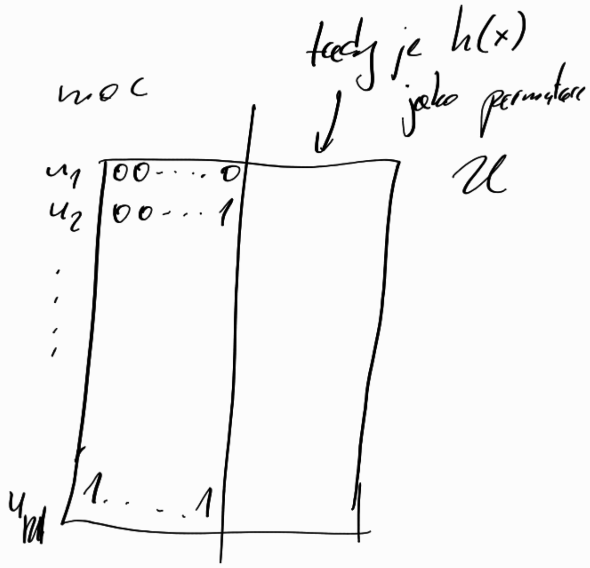
- princip kukaččího hashování se možná nezkouší.
- ale jakmile víte o kukaččím grafu, tak byste ten důkaz jistě složili :)

---

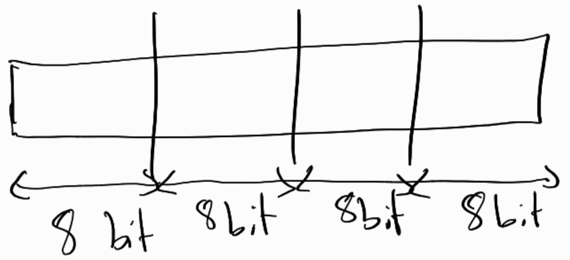
Tabulační hashování

- mít ideální (náhodnou) hash. fci je jako mít tabulku velikosti

$\lg(|\mathcal{U}|!) \cdot \lg|\mathcal{U}|$ bitů. To je moc




př: 32-int




- můžeme však mít ideální hash fci pro každý blok 8 bitů zvlášť a výsledky třeba proxorovat
- paměť už je jen $4 \cdot 2^8$ vs. předchozích $2^{32}$
- potřebujete v DÚ cuckoo_hash