HEŠOVÁNÍ: RŮZNÉ DRUHY TABULEK

Příklady rodin hešovacích funkcí

- skalární součin: $\mathbb{Z}_p^k \to \mathbb{Z}_p$, $h_{\vec{a}}(\vec{x}) := \vec{a}\cdot\vec{x}$ pro $\vec{a}\in\mathbb{Z}_p^k$ ... 1-univerzální
  $h_{\vec{a},b}(\vec{x}) := \vec{a}\cdot\vec{x}+b$, $\vec{a}\in\mathbb{Z}_p^k$, $b\in\mathbb{Z}_p$ ... (2,1)-nezávislý
- lineární: $\mathbb{Z}_p \to [m]$, $h_{a,b}(x) := ((ax+b) \bmod p) \bmod m$, $a,b\in\mathbb{Z}_p$, $a\neq 0$ ... 1-univerzální
  připustíme-li i $a=0$ ... 2-univerzální, (2,4)-nezávislý
- tzv. multiply-shift: $[2^w] \to [2^l]$, $h_{a,b}(x) := (ax+b)\langle w-l, w\rangle$ ← bity na pozicích $w-l$ až $w-1$
  ... 2-univerzální
  $a,b\in[2^w]$, a liché
  - $h'_{a,b}(x) := (ax+b)\langle t-l, t\rangle$, $a,b\in[2^t]$, a liché, $t\geq w+l$ ... 2-nezávislý
- polynomy: $\mathbb{Z}_p \to [m]$, $h_{\vec{a}}(x) := \left(\sum_{i=0}^{d-1} a_i x^i \bmod p\right) \bmod m$, $\vec{a}\in\mathbb{Z}_p^d$, $p\geq dm$ ... d-nezávislý
- tabelace: $[2^{kl}] \to [2^t]$, nagenerují tabulky $T_1 - T_k : [2^l] \to [2^t]$ } čas $O(c)$, prostor $O(c\cdot U^{1/c})$ slov délky t
  $h(x) := \bigoplus_{i=1}^{k} T_i[x\langle (i-1)l, il\rangle]$
  ... je 3-nezávislé, ale není 4-nezávislé

Separované řetězce

pro $m = \Theta(n)$ ... zvětšuji/zmenšuji tabulku dle potřeby (jako nafukovací pole)

- $\mathbb{E}[C_t] \in O(1)$ } stačí c-univerzalita
  C_t = délka řetězce v t-té přihrádce
- $\operatorname{var} C_t \in O(1)$

Dk: $\operatorname{var} C_t = \underbrace{\mathbb{E}[C_t^2]}_{L} - \mathbb{E}[C_t]^2$

$$L = \frac{1}{m}\sum_{s=1}^{m}\mathbb{E}[C_s^2] = \frac{1}{m}\sum_{i\neq j}\underbrace{\Pr[h(x_i)=h(x_j)]}_{\leq c/m} \leq \frac{cn^2}{m^2} \in O(1).$$

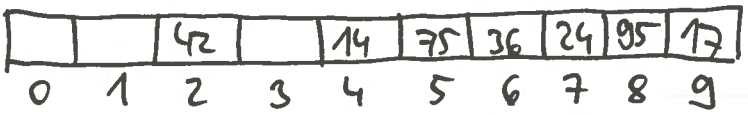
- pro zcela náhodnou funkci h: $\max_t C_t \in \Theta\left(\frac{\log n}{\log\log n}\right)$ s velkou pravděpodobností
  ↳ též pro $\Omega\left(\frac{\log n}{\log\log n}\right)$-nezávislý systém
  $\exists c>0\ \forall^* n$ (až na konečně mnoho výjimek)
  $\Pr[\text{platí pro } n] \geq 1 - n^{-c}$

a pro tabelaci

Lineární přidávání

- vše v jednom poli velikosti m
- Insert hledá první volnou pozici z $h(x)$, $h(x)+1$, $h(x)+2$, ... (mod m)
- Find se zastaví na první prázdné pozici
- Delete prvky „škrtá", když je škrtnutých moc, přebudujeme tabulku.

Příklad: hesuji podle posl. číslice, vkládám 75, 36, 14, 42, 24, 95, 17:



Analýza pro $m \geq (1+\varepsilon)n$: Insert trvá v průměru

- $O(1/\varepsilon^2)$ pro úplně náhodnou fci h
- $O(f(\varepsilon))$ pro $(\log n)$-nezávislý systém
- $O(1/\varepsilon^{\frac{13}{6}})$ pro 5-nezávislý systém
- $\Omega(\log n)$ pro některé 4-nezávislé
- $\Omega(\sqrt{n})$ pro některé 2-nezávislé, $\Theta(\log n)$ pro Multiply-Shift
- $O(1/\varepsilon^2)$ pro tabelaci

Perfektní hešování FKS (Fredman, Komlós, Szemerédi) ← 1984

Chceme najít funkci $f: U \to [m]$, která na dané množině $X \in \binom{U}{n}$ nemá žádné kolize, = je "perfektní"

- chceme lineární paměť, tedy $m = \Theta(n)$
- perf. fcí je obecně málo: třeba pro $m = 2n$ je

klesající mocnina: $a^{\underline{k}} = a(a-1)(a-2)\cdots(a-k+1)$

$$\Pr[\text{náhodná } f \text{ je perfektní}] = \frac{(2n)^{\underline{n}}}{(2n)^n} = \frac{(2n)!}{n!\,(2n)^n} \approx \left(\frac{2n}{e}\right)^{2n} \Big/ \left(\frac{n}{e}\right)^n (2n)^n$$

$n! \approx \left(\frac{n}{e}\right)^n$

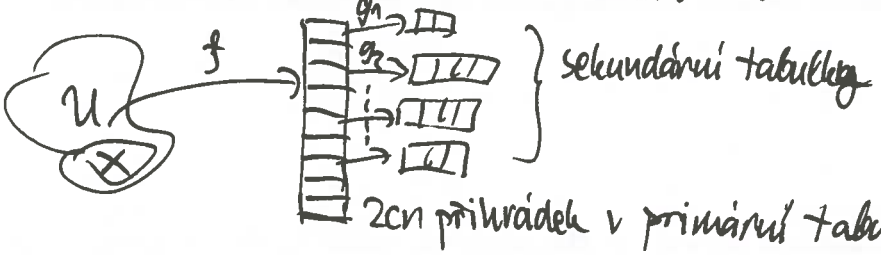
$$= \frac{2^{2n} \cdot n^{2n} \cdot e^n}{e^{2n}\, n^n \cdot 2^n \cdot n^n} = \left(\frac{2}{e}\right)^n$$

Zvolme c-univerzální systém $\mathcal{H}$ do $[m]$, $X = \{x_1, \ldots, x_n\}$.
Pro obecné n, m počítejme $\mathbb{E}_h[K]$, kde $K := \#\{i,j : i \neq j \,\&\, h(x_i) = h(x_j)\}$ ← # kolidujících párů x_i, x_j

$$\mathbb{E}[K] = \sum_{\substack{i,j \\ i \neq j}} \mathbb{E}[\text{indikátor kolize } x_i, x_j] = \sum_{i \neq j} \underbrace{\Pr[h(x_i) = h(x_j)]}_{\leq c/m \text{ z univerzality}} \leq \frac{cn(n-1)}{m} \leq \frac{cn^2}{m}.$$

- pro $m \geq 2cn^2$ je $\mathbb{E}[K] < \frac{1}{2}$
  - $\Rightarrow \Pr[h \text{ není perfektní}] = \Pr[K \geq 1] < \frac{1}{2}$ z Markovovy nerovnosti
  - $\Rightarrow$ průměrný # pokusů, než sčítám perfektní h, je ≤ 2
  - $\Rightarrow$ průměrný čas $O(n)$ .. 1 pokus stihnu v $O(n)$ w.c.

- pro $m \geq 2cn$ je $\mathbb{E}[K] < \frac{n}{2}$ ... umím rychle sehnat h s $K < n$.
  - kolize řeším sekundárními tabulkami kvadratické velikosti



Pamatujeme si:
- parametry funkce f
- pole $\lceil 2cn \rceil$ primárních přihrádek, pro každou:
  - počet prvků a_i
  - parametry funkce g_i (perfektní)
  - odkaz na sekundární tabulku velikosti $\lceil 2ca_i^2 \rceil$

Celková velikost sekundárních tabulek:

$$\sum_{i=1}^{n} \lceil 2ca_i^2 \rceil \le \sum_i (2ca_i^2 + 1) = n + 2c \underbrace{\sum_i a_i^2}_{= \#\{i,j : f(x_i)=f(x_j)\} = n + (\#\{i,j : i \neq j \wedge f(x_i)=f(x_j)\})} \le n + 2c \cdot 2n = (4c+1)n \in O(n).$$

$(\#\text{kolizí fce } f) < n$

Celkově: prostor $O(n)$, konstrukce v průměrném čase $O(n)$.

Kukačkové hešování (Pagh, Rodler 2004)

Pořídíme si 2 hešovací funkce $f, g : \mathcal{U} \to [m]$ (hešují buď do společné tabulky T nebo dvou oddělených)

Prvek x hledáme v $T[f(x)]$ a $T[g(x)]$ a nikde jinde → Find trvá $O(1)$ w.c.

Insert:
- pokud $T[f(x)] = \emptyset$, vložíme x do $T[f(x)]$
- jinak prvek $T[f(x)]$ "vykostrčíme" a pokusíme se ho zahešovat podle opačné funkce
  - ... tím možná vystrčíme nějaký další atd.
- po $\sim \log n$ krocích vyhlásíme timeout a vše přehešujeme s jinými f, g.
- když se tabulka příliš zaplní, přehešujeme.

Delete: škrtátkem, občas přehešujeme.

Analýza: Timeout nastavíme na $6 \log n$ kroků, předpokládáme $(6 \log n, 1)$-nezávislý systém. (funguje i pro jiné konstanty, ale jsme líní)

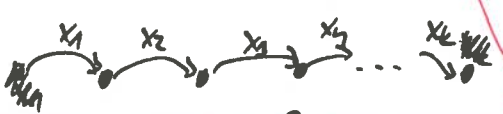
(pro max. $6 \log n$ prvků se chová úplně náhodně ... stačí na malé konstanty)

Nastavíme $m = 4n$. ← stačilo by $m \ge 2(1+\varepsilon)n$.

Vkládáme prvek x_1, postupně vyhazujeme prvky $x_2, x_3, \dots, x_k$.

Uvažujeme kukačí graf: vrcholy jsou přihrádky, hrany $\{f(x_i), g(x_i)\}$ ← tedy odpovídají prvkům

(1) Pokud $x_1 \dots x_k$ jsou navzájem různé:



$\Pr[k \ge 2] = \Pr[T(x_i) \text{ obsazeno}] \ge \Pr[\exists y : f(x)=g(y) \vee f(x)=f(y)] \le \frac{2n}{m} = \frac{1}{2}$.

(# přesunů, tedy $k'=k-1$)

$\Pr[k \ge t] \dots$ indukcí $\le \frac{1}{2^t}$

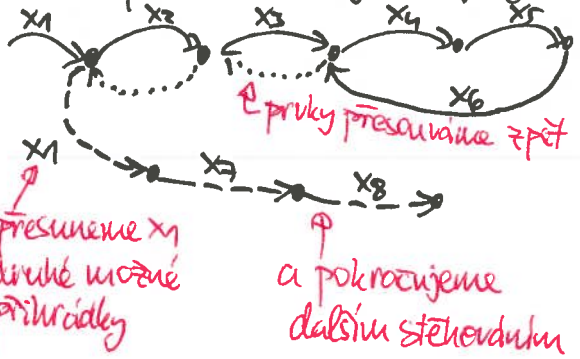
$\mathbb{E}[\#\text{přesunů}] = \sum_t \Pr[\#\text{přesunů} \ge t] \le \sum_t \frac{1}{2^t} \in O(1)$.

(pro $(6\log n, 1)$-nez. to platí přesně, jinak až na konst.)

$\Pr[\text{timeout}] \le \frac{1}{2^{6 \log n}} = n^{-6}$.

(český obecný trik: pro nezápornou celočíselnou veličinu X je $\mathbb{E}[X] = \sum_t \Pr[X \ge t]$ ... rovnou z definice $\mathbb{E}[X]$)

② Vystrčíme prvek x_1, který už byl jednou vystrčen:



ⓐ při 2. průchodu se už nic nezopakuje
- přitímeoutuje aspoň 1 typ hran se vyskytne aspoň $\frac{6\log n}{3} = 2\log n$ - krát
- podle ① se $\forall$ typ hran vyskytne v průměru $O(1)$-krát

$\Rightarrow \Pr[\text{timeout}] \leq 2^{-2\log n} \in O(n^{-2})$

ⓑ při 2. průchodu se také nějaký prvek zopakuje $\Rightarrow$ určitě nastane timeout

Počítejme konfigurace s t různými x_i a pro fixní x_1:

$$\#\text{konf.} \leq \underbrace{n^{t-1}}_{\text{volby ostatních } x_i} \cdot \underbrace{t}_{\text{po kolika kroc. skončí 1. cyklus}} \cdot \underbrace{t}_{\text{kam se 1. cyklus vrátí}} \cdot \underbrace{t}_{\text{kam se vrátí 2. cyklus}} \cdot \underbrace{m^{t-1}}_{\text{hodnoty heš-fcí ve vrcholech}} \in O(n^{t-1} m^{t-1} t^3)$$

Pr dané konfigurace je $\left(\frac{2}{m^2}\right)^t = \frac{2^t}{m^{2t}}$ ← 2 možné orientace hrany

Pr případu ⓑ pro konkrétní t $\in O\left(\frac{n^{t-1} m^{t-1} t^3 2^t}{m^{2t}}\right) = O\left(\frac{n^{t-1} 2^{2t-2} n^{t-1} t^3 2^t}{2^{4t} n^{2t}}\right) = O\left(\frac{t^3}{2^t n^2}\right)$ ← $m = 4n$

Přes všechna t: $\leq \sum_{t=2}^{\infty} O\left(\frac{t^3}{2^t n^2}\right) = O\left(\frac{1}{n^2} \cdot \underbrace{\sum_t \frac{t^3}{2^t}}_{\text{konverguje}}\right) = O\left(\frac{1}{n^2}\right)$

Celkem: Pokud nepřehešováváme, Insert stojí průměrně $O(1)$.
Přehešování nastane s pstí $\underbrace{O(1/n^2)}_{p}$ a způsobí n Insertů + průchod řetězcem délky $\Theta(\log n)$

Tedy $\mathbb{E}[T_{Ins}] =$

$$= p \cdot \left(\Theta(\log n) + n \cdot \mathbb{E}[T_{Ins}]\right) + (1-p) \cdot O(1)$$

… řešením rekurence je $O(1)$.

Shrnutí: Pro $(6\log n)$-nezávislý systém umíme Find $O(1)$ w.c.
Ins, Del $O(1)$ očekávaně amortizovaně

Dále se ví:
- 6-nezávislost obecně nestačí
- tabelační hešování stačí

umí se sestrojit takový, který hešuje v čase $O(1)$ s $O(n^\varepsilon)$ prostoru, ale je značně nepraktický [Siegelovy heš. funkce]