

8. CVIČENÍ Z DATOVÝCH STRUKTUR 1, ZS24/25

Hurá, hešování!!! Univerzální, k -nezávislé, tabulační a možná i perfektní.

1. *Univerzální a nezávislá rozcvička z definic a formálních důkazů.*

a) Dokažte, že $(2, c)$ -nezávislost implikuje c -univerzalitu.

b) Dokažte, že (k, c) -nezávislost implikuje $(k - 1, c)$ -nezávislost.

(Pokud si myslíte, že už jste to někde viděli, tak máte pravdu ... nicméně procvičení nikdy neuškodí :-)

2. *Varianty lineární kongruence (poly mod prime).* Na přednášce byl důkaz 2-nezávislosti systému $\mathcal{H}_{\text{lin}} := \{h_{a,b} \mid a, b \in [p]\}$ pro prvočíslo p , kde $h_{a,b}(x) := ((ax + b) \bmod p) \bmod m$.

a) Zdůvodněte, že \mathcal{H}_{lin} není 3-nezávislá.

b) Co kdybychom měli vždy $b = 0$, tedy volili náhodně jen a ? Bude výsledná rodina c -univerzální? Bude 2-nezávislá?

3. *Vyloženě praktické systémy.* Uvažme systém funkcí $\mathcal{H}_1 = \{\text{id}\}$, který obsahuje jedinou funkci id (identitu), jež zobrazí x na x . Je \mathcal{H}_1 c -univerzální pro nějaké c ? Je \mathcal{H}_1 (k, c) -nezávislý pro nějaká k a c ?

Dále uvažme systém $\mathcal{H}_2 = \{h_a(x) = a : a \in [m]\}$. Dokažte, že tento systém je $(1,1)$ -nezávislý. Dále ukažte, že \mathcal{H}_2 není $(2, c)$ -nezávislý ani c -univerzální pro žádné c .

4. *Hlavní chod: Tabulační (tabulkové) hešování.* Připomeňte si, že tabulační hešování je 2-nezávislé a *není* 4-nezávislé, pokud používáme alespoň dvě tabulky. Cílem je dokázat 3-nezávislost.

Půjdeme na to následující způsobem: Mějme $a, b, c \in \mathbb{Z}_2^\ell, x \neq y \neq z \neq x \in \mathbb{Z}_2^w$, a použijeme tabulkové hashování s d částmi. Pak chceme ukázat, že $\Pr_{h \in \mathcal{H}}[h(x) = a \wedge h(y) = b \wedge h(z) = c] \leq \frac{1}{m^3}$.

i) Prvně si uvědomme, že pokud máme jen jednu část, a tedy jednu tabulku, tvrzení je triviální.

Dále mějme alespoň dvě části. Protože x, y, z jsou různé, musí se (po dvou) lišit alespoň v jedné části.

ii) Začneme s případem, kdy existuje část i , že x^i, y^i, z^i jsou všechny různé. Mějme jakkoliv zvolené ostatní tabulky, kromě tabulky T_i . S jakou pravděpodobností můžeme zvolit funkci pro tabulku T_i tak, že $h(x) = a, h(y) = b, h(z) = c$?

iii) Jinak existují (BÚNO) části i, j takové, že $z^i = x^i \neq y^i$ a $y^j = x^j \neq z^j$. Potom máme následující soustavu rovnic, kde v_x, v_y, v_z jsou vyXORované výsledky z ostatních tabulek:

$$\begin{aligned}T_i[x^i] \oplus T_j[x^j] \oplus v_x &= a \\T_i[y^i] \oplus T_j[y^j] \oplus v_y &= b \\T_i[z^i] \oplus T_j[z^j] \oplus v_z &= c\end{aligned}$$

Opět si představme, že v_x, v_y, v_z už známe. S jakou pravděpodobností budou náhodně volené tabulky T_i, T_j splňovat tuto soustavu rovnic?

iv) Uvědomte si, že toto stačí.

5. *Pokud vás ještě baví definice: Modulo univerzálního systému nemusí být univerzální.* Ukažte, že pokud máme univerzální systém hashovacích funkcí \mathcal{H} , pak systém \mathcal{H}' , kde ke každé fci navíc přidáme modulo m , už nemusí být univerzální. Formálně: Dokažte, že pro každé c a $m > c$ existuje univerzum \mathcal{U} a systém \mathcal{H} z \mathcal{U} do \mathcal{U} tak, že \mathcal{H} je univerzální, ale \mathcal{H}' už není c -univerzální.

6. *Bonus: Perfektní hešování dle Fredmana, Komlóse a Szemerédiho (FKS).* Máme danu n -prvkovou množinu S jako podmnožinu nějakého (obrovského) univerza \mathcal{U} , např. 64-bitové integery. Cílem je navrhnout pro S datovou strukturu velikosti $O(n)$, která zvládne pro zadaný dotaz x zjistit, jestli x náleží v S , v konstantním čase vždy. (V čem klasická hešovací tabulka velikosti $O(n)$ nesplňuje požadavky?)

- Připomeňte si narozeninový paradox, tedy nejmenší počet lidí s rovnoměrně náhodnými narozeninami takový, aby s pravděpodobností alespoň 50% měli dva stejné narozeniny.
- Hešování s úplně náhodnou (nebo c -univerzální) hešovací funkcí funguje podobně. Jak zhruba musíme mít velkou tabulku, aby nastala kolize s pravděpodobností méně jak 50%?
- Perfektní hešování vybudujeme takto: Pořídíme si hešovací funkci $h : \mathcal{U} \rightarrow [m]$ pro $m = O(n)$, kterou vybereme náhodně z c -univerzální rodiny. Pro každou přihrádku $i \in [m]$ zavedeme tabulku druhé úrovně, která bude dost velká, aby tam nenastala kolize s pravděpodobností alespoň 50% (pokud by kolize nastala, změníme hešovací funkci).
- Teď už stačí jen omezit celkovou velikost tabulek druhé úrovně ve střední hodnotě.

Užitečné definice

Definition 1 (c -univerzální systém fci) *Systém \mathcal{H} funkcí $h : \mathcal{U} \rightarrow [m]$ je c -univerzální pro $c > 0$, pokud pro všechna $x \neq y$ platí $\Pr_h[h(x) = h(y)] \leq \frac{c}{m}$. Systém \mathcal{H} je univerzální, pokud je c -univerzální pro nějaké $c > 0$.*

Definition 2 (k -nezávislý systém fci) *Systém \mathcal{H} funkcí $h : \mathcal{U} \rightarrow [m]$ je (k, c) -nezávislý pro nějaká $k \geq 1, c > 0$, pokud $\Pr_{h \in \mathcal{H}}[h(x_1) = a_1 \wedge \dots \wedge h(x_k) = a_k] \leq \frac{c}{m^k}$ pro libovolná x_1, \dots, x_k různá, a_1, \dots, a_k ne nutně různá.*

Systém \mathcal{H} je k -nezávislý, pokud je (k, c) -nezávislý pro nějakou nezávislou konstantu c .

Definition 3 (Tabulkové hashování) *Představme si, že chceme zahashovat n -bitové řetězky do m -bitových řetězků, kde $n = k \cdot \ell$. Řetězek $x \in \{0, 1\}^n$ pak rozložíme do k částí délky ℓ , které značíme x^i . Můžeme tedy psát $x = x^1 x^2 \dots x^k$. Pak generování naší hashovací funkce $h : \{0, 1\}^n \rightarrow \{0, 1\}^m$ vypadá tak, že vybereme uniformně náhodně k funkcí $T_i : \{0, 1\}^\ell \rightarrow \{0, 1\}^m$ (tyto reprezentujeme tabulkou, proto tabulkové hashování). Vyhodnocujeme pak $h(x) = \bigoplus_{i=1}^k T_i(x^i) = T_1(x^1) \oplus T_2(x^2) \oplus \dots \oplus T_k(x^k)$, kde \oplus značí XOR (po jednotlivých bitech).*

Theorem 1 (Markovova nerovnost) *Bud' X nezáporná náhodná veličina. Pak $\forall \varepsilon > 0$ platí $P[X \geq \varepsilon] \leq \frac{\mathbb{E}[X]}{\varepsilon}$. Ekvivalentně pro jakékoli $d \geq 1$, $P[X \geq d \cdot \mathbb{E}[X]] \leq \frac{1}{d}$.*