10. cvičení

Datové struktury I, 6. 12. 2022 `https://iuuk.mff.cuni.cz/~chmel/2223/ds1/`

Věta. Tabulkové hashování je 3-nezávislé.

Úloha 1 (Tuhle větu si dokážeme)
Dokažte předcházející větu s následujícím postupem. Mějme $a, b, c \in \mathbb{Z}_2^\ell, x \neq y \neq z \neq x \in \mathbb{Z}_2^w$, a používejme tabulkové hashování s d částmi. Pak chceme ukázat, že $\Pr_{h \in \mathcal{H}}[h(x) = a \wedge h(y) = b \wedge h(z) = c] \leq \frac{1}{m^3}$.
(a) 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.
(b) Prvně prozkoumejme případ, 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 $h(x) = a, h(y) = b, h(z) = c$?
(c) 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?
(d) Uvědomte si, že toto stačí.

Lemma. Nechť $\mathcal{F}$ je c-univerzální rodina funkcí $f : \mathcal{U} \to [r]$, $\mathcal{G}$ je $(2, d)$-nezávislá rodina funkcí $g : [r] \to [m]$. Pak $\mathcal{H} = \mathcal{F} \circ \mathcal{G} = \{f \circ g : f \in \mathcal{F}, g \in \mathcal{G}\}$ je $(2, c')$-nezávislá, kde $c' = d \cdot \left(\frac{cm}{r} + 1\right)$.

Důkaz. Mějme $x_1 \neq x_2, i_1, i_2 \in [m]$ a uvážíme jevy M - match: $h(x_1) = g(f(x_1)) = i_1, h(x_2) = i_2$ a C - kolize: $f(x_1) = f(x_2)$. Pak $\Pr[M] = \Pr[M \wedge \neg C] + \Pr[M \wedge C] = \Pr[M|\neg C] \cdot \Pr[\neg C] + \Pr[M|C] \cdot \Pr[C] \leq \frac{d}{m^2} \cdot 1 + \frac{d}{m} \cdot \frac{c}{r} = \frac{d \cdot \left(\frac{cm}{r}+1\right)}{m^2}$. ⊞

Tvrzení. Pro prvočíslo p a délku vektoru d definujeme třídu hashovacích funkcí $\mathcal{R} = \{h_a : a \in \mathbb{Z}_p\}$, kde $h_a(x) = \sum_{i=0}^{d-1} x_{i+1} a^i$. (Bereme $x \in \mathbb{Z}_p^d$, indexujeme od jedničky.) Pak $\mathcal{R}$ je d-univerzální, a $\mathcal{R} \circ \mathcal{L}$ je $(2, 5/2)$-nezávislý systém, máme-li $p \geq 4dm$, kde $\mathcal{L} = \{h_{a,b}(x) = (ax + b \mod p) \mod m : a, b \in \mathbb{Z}_p\}$.

Úloha 2 (I tohle si dokážeme)
Dokažte předcházející tvrzení. Začneme s univerzalitou. Chceme tedy dokázat, že pro každá dvě slova $x \neq y \in \mathbb{Z}_p^d$ je $\Pr_a[\sum_{i=0}^{d-1} x_{i+1} a^i = \sum_{i=0}^{d-1} y_{i+1} a^i] \leq \frac{d}{p}$.
Hint: Základní věta algebry říká, že každý polynom stupně d má nejvýše d kořenů – zkuste tam nějaký najít
Dále použijeme lemma, které jsme si dokázali společně na začátku hodiny, a fakt z přednášky, že $\mathcal{L}$ je (2,2)-nezávislý systém pro $p \geq 4m$. Tím jsme už vše dokázali.

Úloha 3 (Univerzální modulo může rozbít univerzalitu)
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 $m > 1$ 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.

Řešení
Uvažme $\mathcal{H}_1$ z předchozí úlohy - pak $\mathcal{H}_1 \mod m$ nemůže být c-univerzální, protože dokud $m < |\mathcal{U}|$, pak prvky 1 a $m + 1$ se vždycky zobrazí na tentýž prvek 1.

Úloha 4 (Vyhledávání jehly v textu)
Vymyslete algoritmus na nalezení všech výskytů podřetězce x délky n v textu T délky m pomocí hashování, který běží v průměrném čase (tj. ve střední hodnotě) $\mathcal{O}(n + m + k \cdot n)$, kde k je počet výskytů x v T.

Řešení
Rabin-Karp s Rolling hashem, čas: máme m času na projití řetězce, v každém kroku uděláme konstantní úpravu hashe a zkontrolujeme, že není stejný. Pokud je hash stejný, zkontrolujeme celý string. Protože máme hashování do nějakého $\mathbb{Z}_p$, pravděpodobnost kolize je d/p, a tedy celkem máme ve střední hodnotě asi $(k + md/p)$ kolizí. Pokud ale zvolíme $p > m \cdot d$ (nebo obecně stačí $p \in \Omega(m \cdot d)$), máme konstantně mnoho falešných kolizí ve střední hodnotě.