Hashování řetězců/vektorů

- chceme hashovat $\vec{x} = (x_0, \dots, x_{d-1}) \in \mathcal{U}^d$

Konstrukce hash. fce z polynomu.

- vybereme prvočíslo $p \geq |\mathcal{U}|$
- vybereme náhodné $a \in [p]$
- $h_a(x) = \left(\sum_{i=0}^{d-1} x_i a^i \right) \bmod p$

Věta: Systém $\mathcal{H} = \{ h_a \mid a \in [p] \}$ je d-universální.

Dk:

- uvažme $\vec{x} \neq \vec{y} \in \mathcal{U}^d$
- $\sum_{i=0}^{d-1} x_i a^i = \sum_{i=0}^{d-1} y_i a^i \iff \sum_{i=0}^{d-1} (x_i - y_i) a^i = 0 \iff a$ je kořen polynomu $h(a) = \sum_{i=0}^{d-1} (x_i - y_i) a^i$
  - zde jsou $x_0 - y_0, x_1 - y_1, \dots, x_{d-1} - y_{d-1}$ koeficienty polynomu
- základní věta algebry: Nenulový polynom stupně d má $\leq d$ kořenů

$\rightarrow$ pro $\leq d$ hodnot a se polynom $h(a)$ vyhodnotí na 0

$\rightarrow \Pr[h_a(x) = h_a(y)] \leq \frac{d}{p} \Rightarrow d$-universalita $\square$

Jak rychle lze vyhodnotit $h_a(x)$?

- $x_0 a^0 + x_1 a^1 + x_2 a^2 + \dots + x_{d-1} a^{d-1} = x_0 + a\left(x_1 + x_2 a^1 + x_3 a^2 + \dots + x_{d-1} a^{d-2}\right) =$

$$= x_0 + a\left(x_1 + a\left(x_2 + x_3 a + \dots\right)\right).$$

$\rightsquigarrow$ d násobení a sčítání.

- tohle je tzv. Hornerovo schéma

Požadavek $p \geq |\mathcal{U}|$

- ... je celkem drsný
- můžeme zredukovat p na nějaké m jak jsme zvyklí?
- ano, když výslednou hodnotu $h_a(\vec{x})$ hashujeme něčím

  2-nezávislým (např. $ax+b \bmod m$)

<u>Lem:</u> Nechť $\mathcal{F}$ je c-universální hash. systém z $\mathcal{U}$ do $[r]$.

Nechť $\mathcal{G}$ je $(2,d)$-nezávislý hash. systém z $[r]$ do $[m]$.

Pak jejich složení $\mathcal{H} = \mathcal{F} \circ \mathcal{G}$ (nejdříve aplikujeme $\mathcal{F}$, pak $\mathcal{G}$)

je $(2,c')$-nezávislý pro $c' = \left(\frac{cm}{r} + 1\right) d$.

- pro hashování polynomy: pokud $p \geq 4dm$, pak složení polynomů

  s $ax+b \bmod m$ je $\left(2, \frac{5}{2}\right)$-nezávislé.

Dk lemmatu:

- co chceme? Pro $x \neq y \in \mathcal{U}$, $i, j \in [m]$ chceme

$$\Pr_{\substack{f \in_R \mathcal{F} \\ g \in_R \mathcal{G}}} \left[g(f(x)) = i \wedge g(f(y)) = j \right] \le \frac{c'}{m^2}.$$

- $g \in \mathcal{G}$ je 2-nezávislý, hotovo z definice?
- ne, $f(x)$ není zaručeně různé od $f(y)$, abychom mohli aplikovat nezávislost :(
- ale $f \in \mathcal{F}$ je c-universální, takže jsou různé skoro vždy :)

- ozn:
  - M: jev $g(f(x)) = i \wedge g(f(y)) = j$
  - C: jev $f(x) = f(y)$
- $$\Pr[M] = \Pr[M \wedge \bar{C}] + \Pr[M \wedge C] = \Pr[M \mid \bar{C}] \Pr[\bar{C}] + \Pr[M \mid C] \Pr[C]$$

- $\Pr[M \mid \bar{C}] \overset{\text{2-nezávislost}}{\Longrightarrow} \le \frac{d}{m^2}$
  - zde $f(x) \neq f(y)$
- $\Pr[\bar{C}] \le 1$ stačí, tohle je těch $\dots \underbrace{+1)d}$ ve znění

- $\Pr[M \mid C]$
  - $f(x) = f(y) =: \xi$ (došla písmenka, proto ξ)

- ptáme se na $M \cap C \iff g(x) = i \wedge g(x) = j$

- $i \neq j \Rightarrow \Pr[M \cap C] = 0$

  $i = j \Rightarrow \Pr[M \cap C] \leq \frac{d}{m}$ pokud uděláme union bound přes $x \in [0, \ldots, m-1]$.

- $\Pr[C] \leq \frac{c}{r}$ z c-universality $\mathcal{F}$.

$\rightarrow \Pr[M] \leq \frac{d}{m^2} \cdot 1 + \frac{d}{m} \cdot \frac{c}{r} = \frac{1}{m^2}\left(d + \frac{cdm}{r}\right) = \frac{d}{m^2}\boxed{\left(1 + \frac{cm}{r}\right)}$

c'

[!]

- tohle teda platí pro $c' = (c+1)d$
- g hashuje z $[r]$ do $[m]$

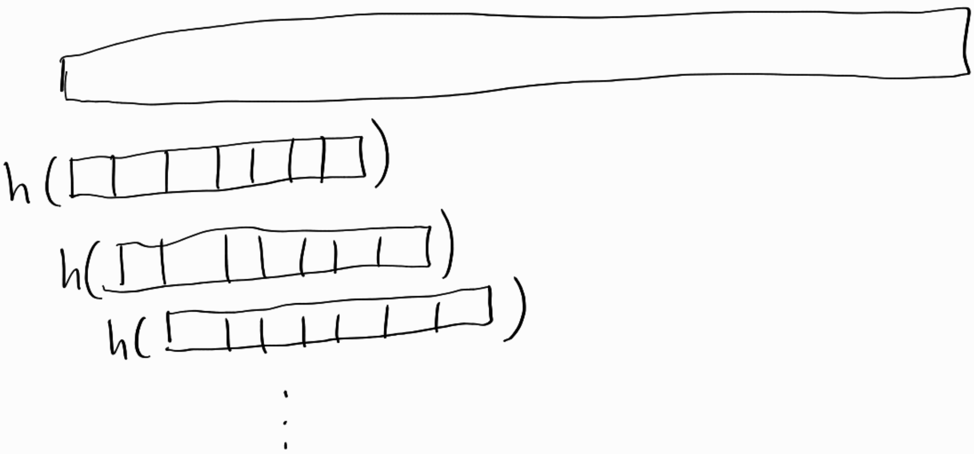
$\rightarrow r \geq m \Rightarrow \frac{m}{r} \leq 1$ a dosadíme za c'.

Rabin-Karp alg.

- chceme hledat podřetězce σ v textu S a máme „málo" prostoru
  - např. KMP, BM, Suffix Array/Tree využívají paměť $\Omega(|\sigma|)$
- zajímá nás $S[i : i+|\sigma|] = \sigma$ pro $\approx |S|$ různých i
- kontrola všech pozic je drahá $\rightarrow O(|S| \cdot |\sigma|)$
- budeme kontrolovat jen ty pozice, kde $h(S[i:i+|\sigma|]) = h(\sigma)$

pro vhodnou hash fci h.

- na první pohled je tohle ještě horší než naivní celý neboť musíme navíc hashovat, což stojí $\Omega(|\sigma|)$ času.
- ledaže bychom uměli počítat hash fci v „konst" čase :)
- S



- chceme každou další h kromě první počítat v konst. čase
- použijeme polynomiální systém z předchozích 4 stran
- máme $S[j]\,a^0 + S[j+1]\,a^1 + S[j+2]\,a^2 + \dots + S[j+|\sigma|-1]\,a^{\dots}$.
- chceme $S[j+1]\,a^0 + S[j+2]\,a^1 + \dots$
- tedy stačí

$$\left.\begin{aligned} &\text{odečíst } S[j]\,a^0, \\ &\text{vydělit } a \\ &\text{přičíst } S[j+|\sigma|]\,a^{|\sigma|-1}. \end{aligned}\right\} O(1) \text{ času}$$

a máme to

- alternativně

$$S[j]\,a^{|\sigma|-1} + S[j+1]\,a^{|\sigma|-2} + \dots + S[j+|\sigma|-1]\,a^0$$

a chceme

$$S[j+1]\,a^{|\sigma|-1} + \dots + S[j+|\sigma|-1]\,a^1 + S[j]\,a^0$$

- stačí odečíst $S[j]\,a^{|\sigma|-1}$,

  vynásobit a

  přičíst $S[j]\,a^0$

- pokud je dělení o hodně dražší než násobení, tak alternativa je rychlejší.
  - to je většinou pravda, zejména kvůli HW paralelizaci násobení
  - pro dělení existuje nějaký taky, ale paralelní
    - na starších Pentiumech (Intel) měl těžký bug a
    - Intel si ho zapatentoval :(

Analýza

- první hash stojí $O(|\sigma|)$ a každý další $O(1)$

$\rightarrow$ celkem $O(|S| + |\sigma|)$ … opt :)

- ozn N počet pozic i t.ž. $h(S[i : i+|\sigma|-1]) = h(\sigma)$
- celková složitost je

$$O(|S| + |\sigma| + |\sigma| \cdot N)$$

pokud stačí zahlásit první výskyt, jinak tam je ještě $\#$ výskytů $\cdot\, |\sigma|$

- kolik je N?
- polynomy jsou d-nezávislý hash systém

Odsud dolů to jde možná zlepšit, nestíhám ☹

$\Rightarrow$ pro $\tau \neq \sigma$ platí

$$\Pr[h(\sigma) = h(\tau)] \leq \frac{d}{p}$$

- $\mathbb{E}[N] \leq \frac{d}{p} |S|$
- zvolíme-li: $p \geq d|S|$, máme vyhráno.