Hašování

Co je hashovací funkce? Čím se liší od nějaké funkce? Co požadujeme ohledně času a prostoru?

Rodina hashovacích funkcí $H = \{h : \mathcal{U} \to [m]\}$ je *c-univerzální* pokud
$\forall x \neq y \in \mathcal{U} : Pr_{h\in H}[h(x) = h(y)] \leq \frac{c}{m}$
Rodina je *k-nezávislá* pokud
$\forall x_1, \ldots, x_k \in \mathcal{U}, \forall t_1, \ldots, t_k : Pr_{h\in H}[\forall i : h(x_i) = t_i] \leq \frac{1}{m^k}$
(disclaimer: terminologie v různých zdrojích se liší)

- Použijme funkci z 2-univerzální rodiny $H : \mathcal{U} \to [m]$ na celkem n prvků. Ukažte, že pokud $m \geq 2n^2$, tak pravděpodobně nenastanou žádné kolize.
- Rozhodněte zda je 1-univerzální rodina je 2-nezávislá a naopak.
- Máme univerzální rodinu hashovacích funkcí $\mathcal{U} \to [2^k]$. U kolika (binárních zápisů) hashů můžeme očekávat, že budou končit na alespoň i nul pokud hashujeme m prvků?
- Vkládejme prvky do pole, v případě kolize zkoušíme následující buňku dokud nenajdeme prázdnou. V čem je lepší se posouvat místo o jednu pozici o c pozic?

Párek rodin hashovacích funkcí

a) $\mathcal{U} = \mathbb{Z}_p^d; H = \{h_v(w) =< v, w >: v \in \mathcal{U}\}$

b) $\mathcal{U} = \mathbb{Z}_p^d; H = \{h_c(w) = \sum w_i * c^i \bmod p : c \in \mathbb{Z}_p\}$

- Rozhodněte, zda jsou tyto rodiny 2-nezávislé.
- Mějme dáno n vektorů. Rozhodněte v čase $\mathcal{O}_{\mathbb{E}}(n)$ zda jsou na vstupu nějaké stejné vektory. Předpokládejme, že si umíme pořídit pole, které není třeba inicializovat (černá magie).
- Chceme datovou strukturu, která umí uchovávat zhruba m čísel, a která podporuje Insert, Delete a umí pro číslo x rozhodnout zda jsou ve struktuře dva prvky se součtem x. Je potřeba zvolit vhodnou rodinu funkcí, kde umíme hash součtu vypočítat z hashů sčítanců. Porovnejte se strukturou na principu binánrích stromů. Co když naše prvky jsou vektory?
- Máme dlouhou posloupnost čísel obsahující celkem m prvků, ale pouze n z nich jsou různé. Chtěli bychom přibližně určit n, ale máme k dispozici asymptoticky méně než n paměti. Příklad: paměť $= O(m^{1/10})$, $n = m^{1/2}$. (velmi těžké)

Domácí úkol

Vezměme rodinu hashovacích funkcí b) ze cvičení, tedy:

pro $\mathcal{U} = \mathbb{Z}_p^k$ (univerzum k-dimenzionálních vektorů nad $\mathbb{Z}_p$)

$H = \{h_c(w) = \sum_{i=0}^{k-1} w_i * c^i \bmod p : c \in \mathbb{Z}_p\}$

(polynom c s koeficienty w, počítáno mod p, c vybíráme náhodně)

Předpokládejte, že je 2-nezávislá a 1-univerzální. Ukažte, že pokud známe hodnotu $h_c(a_1, a_2, \dots, a_k)$, tak umíme rychle spočítat $h_c(a_2, a_3, \dots, a_k, a_{k+1})$ v konstantním čase. Pozn: pokud sčítáme, násobíme a chceme výsledek modulit, můžeme ekvivalentně v každém mezikroku modulit všechny mezivýsledky.

Využijte této znalosti a ukažte, že pokud dostaneme posloupnost $A = a_1, a_2, \dots, a_m$ délky m a vzor $V = v_1, v_2, \dots, v_k$ délky $k < m$, tak umíme v čase $f(m,k) = \mathcal{O}_{\mathbb{E}}(m)$ rozhodnout, zda je vzor V obsažen v A jako souvislá popdposloupnost. Všimněte si, že se jedná o analýzu v průměrném případě, a složitost vůbec nezávisí na k, díky $k < m$ tedy může být složitost $\mathcal{O}_{\mathbb{E}}(m + c_0 k)$, nikoliv však $\mathcal{O}_{\mathbb{E}}(mk)$ (to by šlo triviálně hrubou silou).

Bude třeba vhodně nastavit velikost p (chceme odhad velikosti, neřešíme konkrétní hodnotu nebo jak p najít). Využijte toho, že pro n-prvkovou množinu N, dané p a libovolnou hodnotu s (hash nějakého prvku x) můžeme odhadnout, kolik prvků bude mít hash s následovně:

$\mathbb{E}_c[\#elements] = \Sigma_{i \in N} Pr_c[h_c(i) = s] = \Sigma_{i \in N} Pr_c[h_c(i) = h_c(x)] \leq \frac{m}{p}$

Hint: Pro efektivní výpočet si stačí pamatovat jeden mezivýsledek násobení.

Hint: Není třeba používat žádné pole.

Hint: Pokud dvě posloupnosti mají stejný hash, jak dlouho (v nejhorším případě) nám trvá zjistit, zda jsou stejné? Kolik času ve střední hodnotě zaberou všechny takové kontroly?