1. Speciální datové struktury

FIXME: Úvod.

1.1. Hešování

Lidé už dávno zjistili, že práci s velkým množstvím věcí si lze usnadnit tím, že je rozdělíme do několika menších skupin a každou zpracujeme zvlášť. Příklady najdeme všude kolem sebe: Slovník spisovného jazyka českého má díly A až M, N až Q, R až U a V až Ž. Katastrální úřady mají svou působnost vymezenu územím na mapě. Padne-li v Paříži smog, smí v některé dny do centra jezdit jenom auta se sudými registračními čísly, v jiné dny ta s lichými.

Informatici si tuto myšlenku také oblíbili a pod názvem *hešování* ji často používají k uchovávání dat.

Mějme nějaké universum $\mathcal{U}$ možných hodnot, konečnou množinu přihrádek $\mathcal{P} = \{0, \ldots, p-1\}$ a *hešovací funkci,* což bude nějaká funkce $h : \mathcal{U} \to \mathcal{P}$, která každému prvku universa přidělí jednu přihrádku. Chceme-li uložit množinu prvků $X \subset \mathcal{U}$, rozstrkáme její prvky do přihrádek: prvek $x \in X$ umístíme do přihrádky $h(x)$. Budeme-li pak hledat nějaký prvek $u \in \mathcal{U}$, víme, že nemůže být jinde než v přihrádce $h(u)$.

Podívejme se na příklad: Universum všech celých čísel budeme rozdělovat do 10 přihrádek podle poslední číslice. Jako hešovací funkci tedy použijeme $h(x) = x \bmod 10$. Zkusíme uložit několik slavných letopočtů naší historie: 1212, 935, 1918, 1948, 1968, 1989:

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
		1212			935			1918 1948 1968	1989

Hledáme-li rok 2015, víme, že se musí nacházet v přihrádce 5. Tam je ovšem pouze 935, takže hned odpovíme zamítavě. Hledání roku 2016 je dokonce ještě rychlejší: přihrádka 6 je prázdná. Zato hledáme-li rok 1618, musíme prozkoumat hned 3 hodnoty.

Uvažujme obecněji: kdykoliv máme nějakou hešovací funkci, můžeme si pořídit pole p přihrádek, v každé pak „řetízek“ – spojový seznam hodnot. Tato jednoduchá datová struktura je jednou z možných forem *hešovací tabulky.*

Jakou má hešovací tabulka časovou složitost? Hledání, vkládání i mazání sestává z výpočtu hešovací funkce a projití řetízku v příslušné přihrádce. Pokud bychom uvažovali „ideální hešovací funkci“, kterou lze spočítat v konstantním čase a která zadanou n-prvkovou množinu rozprostře mezi p přihrádek dokonale rovnoměrně, budou mít všechny řetízky n/p prvků. Zvolíme-li navíc počet přihrádek $p = \Theta(n)$, vyjde konstantní délka řetízku, a tím pádem i časová složitost operací.

Praktické hešovací funkce

FIXME

Přehešovávání

FIXME

Následuje zápis z dávné přednášky, který bude postupně zapracován do textu kapitoly.

Mějme universum U (jeho velikost označíme u), množinu P přihrádek ($p = |P|$) a nějakou funkci $h : U \to P$, které budeme říkat *hashovací funkce.*

Datová struktura bude fungovat takto: když prvek vkládáme, spočteme hashovací funkci a vložíme prvek do příslušné přihrádky (přihrádky budeme reprezentovat jako pole seznamů). Pokud chceme prvek vyhledat nebo smazat, opět vyhodnotíme hashovací funkci a dozvíme se, ve které jediné přihrádce ho dává smysl hledat.

Budeme-li předpokládat, že výpočet funkce h trvá $\mathcal{O}(1)$, bude vkládání pracovat v konstantním čase a ostatní operace v čase lineárním s počtem prvků v dané přihrádce. Pokud se hashovací funkce bude chovat „rozumně náhodně“, můžeme očekávat, že po vložení n prvků jich bude v každé přihrádce přibližně n/p, takže při volbě $p \approx n$ můžeme získat konstantní časovou složitost operací. (Volit $p \gg n$ nemá smysl, protože pak bychom inicializací pole trávili příliš mnoho času.)

Tento přístup má ale samozřejmě své zádrhele: potřebujeme s prvky universa umět počítat (už si nevystačíme s porovnáváním), ale hlavně potřebujeme sehnat hashovací funkci, která se chová dostatečně rovnoměrně. Často se používají funkce, které se pro obvyklé vstupy chovají „pseudonáhodně“, třeba:

- $x \mapsto ax \bmod p$, pokud je universum číselné (pro nějakou konstantu a; nejlepší je, když a i p jsou prvočísla);
- $x_1, \ldots, x_n \mapsto \left(\sum_i C^i x_i\right) \bmod p$, pokud hashujeme řetězce (C a p opět nejlépe prvočíselná, navíc je-li ℓ obvyklá délka řetězce, mělo by být $C^\ell \gg p$).

Nicméně, ať už zvolíme jakoukoliv deterministickou funkci, vždy budou existovat nepříjemné vstupy, pro které skončí všechny prvky v téže přihrádce a operace budou mít lineární složitost namísto konstantní. Pomůžeme si snadno: vybereme hashovací funkci náhodně. Ne ze všech funkcí (ty bychom neuměli reprezentovat), nýbrž z vhodně zvolené třídy funkcí, které umíme snadno popisovat pomocí parametrů.

Definice: Systém funkcí S z U do P nazveme c-universální (pro nějaké $c \geq 1$), pokud pro všechny dvojice x, y navzájem různých prvků z U platí

$$\Pr_{h \in S}[h(x) = h(y)] \leq c/p.$$

(Kdybychom volili náhodně z úplně všech funkcí, vyšla by tato pravděpodobnost právě $1/p$ – c-universální systém je tedy nejvýše c-krát horší.)

Lemma: Buď h funkce náhodně vybraná z nějakého c-universálního systému. Nechť $x_1, \ldots, x_n$ jsou navzájem různé prvky universa vložené do struktury a x je nějaký prvek universa. Potom pro očekavaný počet prvků v téže přihrádce jako x platí:

$$\mathbb{E}[\#x : h(x) = h(x_i)] \leq cn/p.$$

Důkaz: Pro dané x definujeme indikátorové náhodné proměnné:

$$Z_i = \begin{cases} 1 & \text{když } h(x) = h(x_i) \\ 0 & \text{jinak} \end{cases}$$

Jinými slovy, Z_i říká, kolikrát padl prvek x_i do přihrádky $h(x)$, což je buď 0 nebo 1. Proto $Z = \sum_i Z_i$ a díky linearitě střední hodnoty je hledaná hodnota $\mathbb{E}[Z]$ rovna $\sum_i \mathbb{E}[Z_i]$. Přitom $\mathbb{E}[Z_i] = \Pr[Z_i = 1] \leq c/p$ podle definice c-universálního systému. Takže $\mathbb{E}[Z] \leq cn/p$. □

FIXME: Doplnit přehashovávání.

Zbývá dořešit, kde nějaký c-universální systém sehnat. Známých konstrukcí je vícero, zde si předvedeme jednu lineárně algebraickou.

Lemma: Předpokládejme, že p je prvočíslo, přihrádky jsou identifikované prvky konečného tělesa $\mathbb{Z}_p$ a universum U je vektorový prostor dimense d nad tělesem $\mathbb{Z}_p$, tedy $\mathbb{Z}_p^d$. Uvažujme systém funkcí $S = \{h_t \mid t \in \mathbb{Z}_p^d\}$, kde $h_t(x) := t \cdot x$ (skalární součin s vektorem s). Pak tento systém je 1-universální.

Důkaz: Nechť $x, y \in \mathbb{Z}_p^d$, $x \neq y$. Potom jistě existuje i, pro nějž $x_i \neq y_i$; bez újmy na obecnosti předpokládajme, že $i = d$. Nyní volíme t náhodně po složkách a počítáme pravděpodobnost kolize (rovnost modulo p značíme $\equiv$):

$$\begin{aligned} &\Pr_{t \in \mathbb{Z}_p^d}[h_t(x) \equiv h_t(y)] = \Pr[x \cdot t \equiv y \cdot t] = \Pr[(x - y)t \equiv 0] = \\ &= \Pr\left[\sum_{i=1}^{d} (x_i - y_i)t_i \equiv 0\right] = \Pr\left[(x_d - y_d)t_d \equiv -\sum_{i=1}^{d-1} (x_i - y_i)t_i\right]. \end{aligned}$$

Pokud už jsme $t_1, \ldots, t_{d-1}$ zvolili a nyní náhodně volíme t_d, nastane kolize pro právě jednu volbu (poslední výraz je linearní rovnice tvaru $ax = b$ pro nenulové a a ta má v libovolném tělese právě jedno řešení). Pravděpodobnost kolize je tedy nejvýše $1/p$, jak požaduje 1-universalita. □

Věta (Bertandův postulát): Pro libovolné $n \geq 1$ existuje prvočíslo p, které splňuje nerovnost $n < p \leq 2n$.