Datové struktury 1

7. Přednáška 18.11.

Pavel Veselý

Plán: Zase hešování

Řešení kolizí:

- K čemu je mít dvě volby *(the power of two choices)*
  - Použití dvou hešovacích funkcí na separované řetězce
  - **Kukačkové hešování**
- Otevřená adresace = 1 prvek v buňce
  - **Lineární přidávání** a jeho (zjednodušená) analýza

Hešovací tabulka

- Reprezentace množiny klíčů (nebo slovníku)
  - Operace `Find, Insert, Delete`
- Tabulka ≈ pole přihrádek velikosti m
- Kolize = dva prvky skončí ve stejné přihrádce

Hlavní části hešovací tabulky:

1. Hešovací funkce
2. Řešení kolizí
   - Separované řetězce – spojový seznam (či dynamické pole) v každé buňce
   - **Dnes lepší…**
3. Dynamizace: udržujeme určité maximální (a minimální) zaplnění
   - Přehešujeme do nové tabulky – podobně jako dynamické pole

Co chceme od rodiny hešovacích funkcí $\mathcal{H}$?

- Univerzalita: malá pravděpodobnost kolize
  - $\forall x_1, x_2 \in \mathcal{U}, x_1 \neq x_2 \colon \Pr_{h \in \mathcal{H}}[h(x_1) = h(x_2)] \leq c/m$

- k-nezávislost: každé k prvků se hešuje (skoro) nezávisle
  - Existuje $c > 0 \colon \forall x_1, \ldots, x_k \in \mathcal{U}, \text{různé}, \forall y_1, \ldots, y_k \in [m]$:
    $$\Pr_{h \in \mathcal{H}}[h(x_1) = y_1 \;\&\; \cdots \;\&\; h(x_k) = y_k] \leq c/m^k$$
- Příklad k-nezávislé rodiny: náhodný polynom stupně $k - 1$ nad $\mathbb{Z}_p$ modulo m
- Uniformita = 1-nezávislost
- Cvičení:
  - 2-nezávislost implikuje univerzalitu (ale ne obráceně)
  - k-nezávislost implikuje $(k-1)$-nezávislost (ale ne obráceně)

Tabelace (tabulkové hešování, Zobristovo)

- $[2^w] \to [2^\ell]$ pro $w = k \cdot t$
- Rozdělíme klíč na k částí $x^{(1)}, x^{(2)}, \dots, x^{(k)}$ o t bitech
  - Každá část hešována plně náhodnou fcí
  - Tedy máme k tabulek T_i pro funkce $[2^t] \to [2^\ell]$
    - Prostor celkově $k \cdot 2^t \cdot \ell$ bitů
  - Heše pro jednotlivé části vyxorujeme, tedy

$$\mathrm{T}_1\left(x^{(1)}\right) \oplus \mathrm{T}_2\left(x^{(2)}\right) \oplus \cdots \oplus \mathrm{T}_\mathrm{k}\left(x^{(k)}\right)$$

- Tabelace je 3-nezávislá
  - Cvičení: 2-nezávislost
  - ale ne 4-nezávislá, pokud máme alespoň dvě tabulky
- Pozn.: Tabelace má mnohé jiné silné vlastnosti…

Síla dvou voleb *(the power of two choices)*

- Nejplnější přihrádka má $\Theta\left(\frac{\log(n)}{\log\log(n)}\right)$ prvků
  - s velkou pravděpodobností pro plně náhodnou hešovací funkci
- Použijeme dvě nezávislé hešovací funkce h_1, h_2
  - Prvek x dáme do méně plné z přihrádek $h_1(x)$ a $h_2(x)$
  - Nyní bude mít nejplnější přihrádka $\Theta(\log\log(n))$ prvků
    - Opět s velkou pravděpodobností pro plně náhodnou hešovací funkci
  - `Find, Insert, Delete` mají pořád očekávanou složitost $O(1)$ pro $m = \Omega(n)$

Kukačkové hešování

- Dvě nezávislé hešovací funkce h_1, h_2
- V přihrádce pouze jeden prvek
- x může být pouze v $h_1(x)$ nebo $h_2(x)$
  - `Find` a `Delete` mají pořád složitost $O(1)$ vždy
- `Insert:` prvky se vyhazují jako kukačky ☺
  - Ale jen omezený počet kroků – jinak přehešujeme s novými h_1, h_2

Věta: Pokud:

- $m \geq (2 + \varepsilon) \cdot n$ (pro konstantní $\varepsilon > 0$),
- h_1, h_2 jsou $\lceil 6 \cdot \log_2 n \rceil$-nezávislé nebo tabelace a
- Timeout při Insertu je $\lceil 6 \cdot \log_2 n \rceil$ (nebo větší),

Pak `Insert` má složitost $O(1)$ ve střední hodnotě a
střední hodnota počtu přehešování při vkládání n prvků je $O(1)$

Otevřená adresace

- V přihrádce pouze jeden prvek
- Každý prvek x má vyhledávací posloupnost = permutace přihrádek
  - $h(x, i)$ = i-tý prvek posloupnosti, $i = 0, 1, 2, \ldots, m - 1$
  - `Insert(`x`):` Prvek přidáme na první volnou buňku v posloupnosti
  - `Find(`x`):` Projdeme posloupnost, dokud nenarazíme na x nebo prázdnou buňku
  - `Delete(`x`):` Pokud najdeme x, buňku označíme za prázdnou 
    - Pokud je pomníčků příliš mnoho, tabulku přehešujeme
- **Lineární přidávání** – „jdeme na následující přihrádku“
  - $h(x, i) = h(x) + i \bmod m$

Otevřená adresace: lineární přidávání

- V přihrádce pouze jeden prvek
- Vyhledávací posloupnost $h(x, i) = (h(x) + i) \bmod m$

➕ Dobré chování ke cache CPU (sekvenční průchod)

➖ Tvoří se úseky obsazených buněk

Známé teoretické vlastnosti: nechť $m \geq (1 + \varepsilon) \cdot n$, pak střední hodnota složitosti operací `Find`, `Insert`, `Delete` je omezena:

- $O(\varepsilon^{-2})$ pro plně náhodnou hešovací funkci, 5-nezávislou, nebo **tabelaci**
- $\Omega(\log n)$ pro jistou 4-nezávislou rodinu
- $\Omega(\sqrt{n})$ pro jistou 2-nezávislou
- Ref.: [Knuth ‘63], [Pătraşcu & Thorup ‘12]

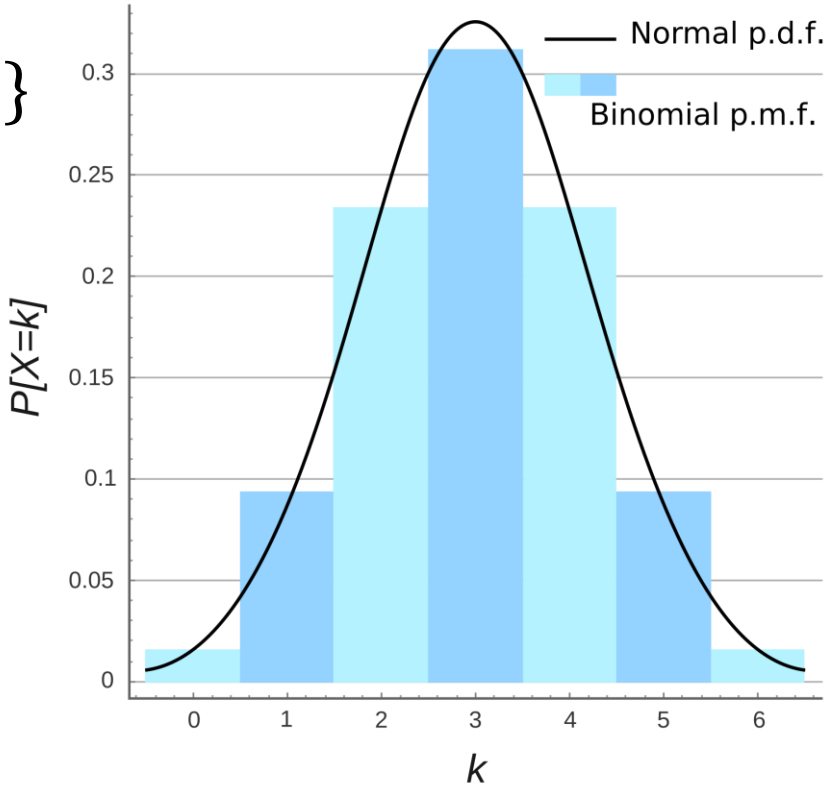
Vsuvka: Černovova nerovnost

- Házíme nezávislými mincemi $X_1, X_2, \ldots, X_n \in \{0,1\}$
  - Mají různé pravděpodobnosti $\Pr[X_i = 1]$
- $\Pr[X_i = 1]$ stejné → binomická distribuce

Znění jedné varianty Černovovy nerovnosti:

- Nechť $X_1, X_2, \ldots, X_n \in \{0, 1\}$ jsou nezávislé náhodné veličiny
- Součet $X = X_1 + X_2 + \ldots + X_n$
  - má střední hodnotu $\mu = E[X]$
- Pak pro všechna $c > 0$:

$$\Pr[X \geq c \cdot \mu] \leq \left(\frac{e^{\delta}}{c^{c}}\right)^{\mu}$$



By Cflm001. Derived from File:BinDistApprox large.png by Xiao Fei, released under GFDL/CC-BY-SA-3.0., CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=7689435

Otevřená adresace: další varianty

- V přihrádce pouze jeden prvek
- Každý prvek x má vyhledávací posloupnost = permutace přihrádek
  - $h(x, i)$ = i-tý prvek posloupnosti, $i = 0, 1, 2, \ldots, m - 1$
- **Lineární přidávání** – „jdeme na následující přihrádku“
  - $h(x, i) = h(x) + i \bmod m$
- **Dvojité hešování –** skáčeme o náhodný počet kroků
  - m je prvočíslo a h, g hešovací funkce do $[m] = \{0, \ldots, m - 1\}$
  - $h(x, i) = h(x) + i \cdot g(x) \bmod m$

Příště

- Zase hešování ☺
  - Bloom filtry
    - datové struktury menší než data, které reprezentují množinu až na malou pravděpodobnost chyby
  - Hešování vektorů a řetězců
- Možná začneme DS pro řetězce