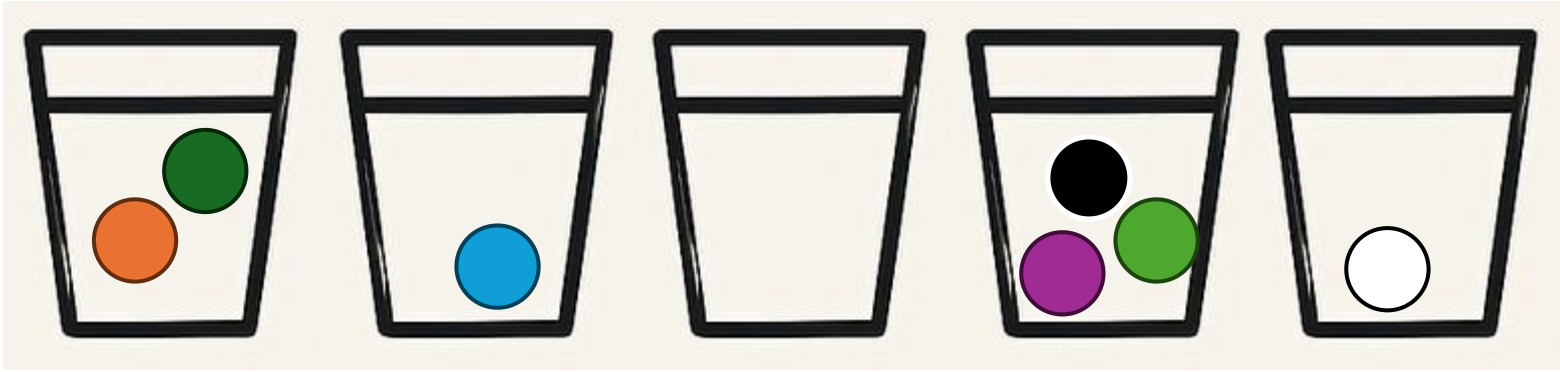
Datové struktury 1

6. Přednáška 11.11.

Pavel Veselý

Plán: Hešování



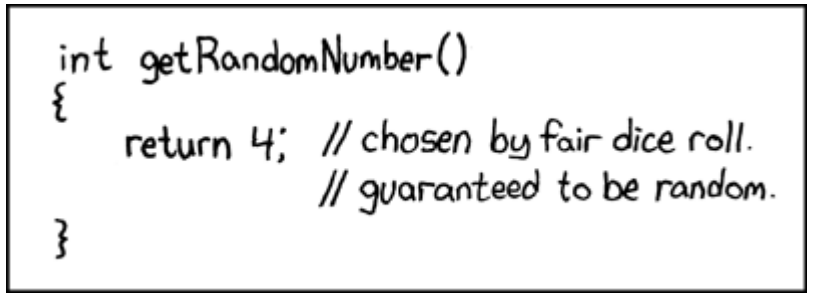
- Házení míčků do košů
- Narozeninový paradox
- Hešovací funkce:
  - Univerzální = malá pravděpodobnost kolize dvou prvků
  - k-nezávislé = k prvků se hešuje nezávisle
- Příště: řešení kolizí

Hešování

- Prvkům z univerza $\mathcal{U}$ přiřadíme „náhodná“ čísla
  - Nebo alespoň „náhodně vypadající“

Hešovací funkce: $h: \mathcal{U} \rightarrow [m]$

- Příklad:
  - $h(x) = a \cdot x \mod m$, kde a je nesoudělné s m



Aplikace:

- **Hešovací tabulka**
- **Bloom filtry**
- Vzorkování
- ....

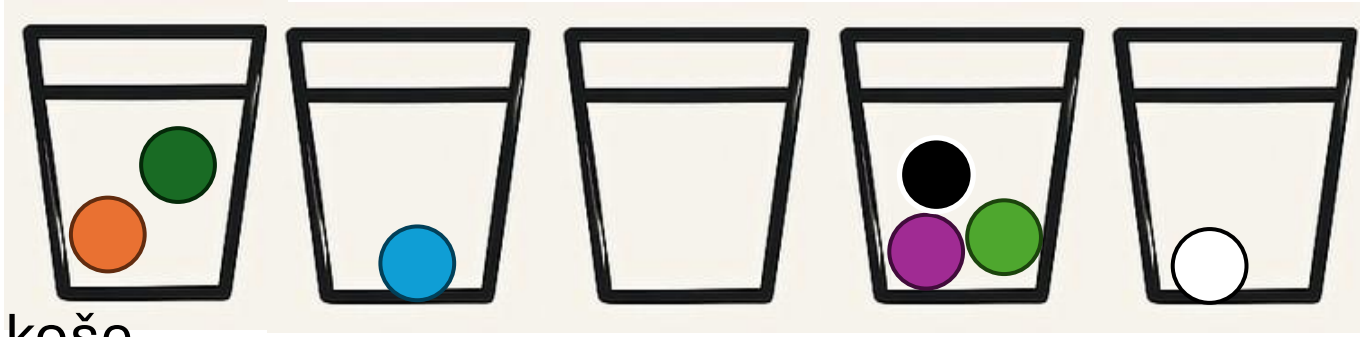
Hešovací tabulka

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

Hlavní části hešovací tabulky:

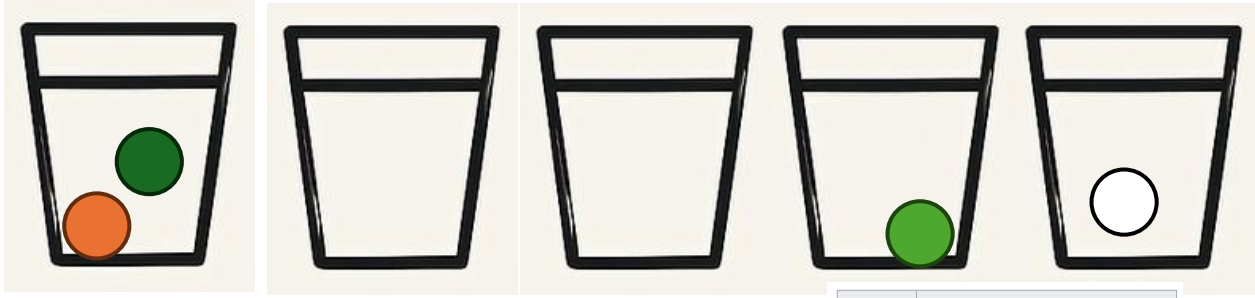
1. Hešovací funkce
2. Řešení kolizí
   - Separované řetězce – spojový seznam (či dynamické pole) v každé buňce
   - Příště lepší…
3. Dynamizace: udržujeme určité maximální (a minimální) zaplnění
   - Přehešujeme do nové tabulky – podobně jako dynamické pole

Hešování jako házení míčků do košů



- n míčků a m košů
  - Každý míček nezávisle hozen do koše
  - Jak to souvisí s hešováním?
- Opakování pravděpodobnosti:
  - Co jsou elementární jevy Ω?
  - Co je náhodná veličina? Co je její střední hodnota?
- Pro házení míčků:
  - Střední hodnota # míčků v libovolném koši je $\frac{n}{m}$
  - S velkou pravděpodobností bude mít nejplnější koš $\Theta\left(\frac{\log(n)}{\log\log(n)}\right)$ míčků
    - Důkaz Černovovou nerovností vynechán

Kdy nenastanou žádné kolize?



- n míčků a m košů
- Pokud $m \gg n$, nastane kolize?
  - Pro jaký poměr m/n už kolize nenastane s pstí. $\geq 50\%$?
  - Odpověď: potřebujeme $m = \Omega(n^2)$ a zároveň to stačí
    - Tedy pro $m \gg n^2$ žádná kolize s velkou pstí
      a pro $m \ll n^2$ nastane kolize s velkou pstí

- Narozeninový „paradox“:
  - Pokud je v místnosti ≥ 23 lidí,
    mají dva stejné narozeniny s pstí. $\geq 50\%$

n	$p(n)$
1	0.0%
5	2.7%
10	11.7%
20	41.1%
23	50.7%
30	70.6%
40	89.1%
50	97.0%
60	99.4%
70	99.9%
75	99.97%
100	99.999 97%
200	$(100 - 2 \times 10^{-28})\%$
300	$(100 - 6 \times 10^{-80})\%$
350	$(100 - 3 \times 10^{-129})\%$
365	$(100 - 1.45 \times 10^{-155})\%$
≥ 366	100%

Hešovací funkce (aneb jak implementujeme házení)

- Plně náhodná (každý prvek z $\mathcal{U}$ zahešován uniformně a nezávisle na ostatních)
  - Potřebuje prostor $\Omega(\mathcal{U})$ na uložení ☹
- Pevná funkce
  - Lze na ní zaútočit – způsobit velký počet kolizí ☹
- Náhodný výběr z množiny funkcí
  - Např. funkce má několik parametrů volených náhodně
- Pro srovnání: kryptografické hešovací funkce
  - Např. MD5, rodina SHA, …
  - Mnohem silnější požadavky:
    - Těžké najít kolizi nebo funkci invertovat
  - … ale pomalejší

Input data is *not* random!
So good hash functions *must be* random!

Autor: Jeff Erickson

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

- Rychlé vyhodnocení + málo prostoru
  - Typicky výpočet v $O(1)$ z parametrů, které vygenerujeme náhodně na začátku
- Uniformita: $\forall x \in \mathcal{U}\ \forall y \in [m]: \Pr_{h \in \mathcal{H}}[h(x) = y] \leq c/m$
- Univerzalita: malá pravděpodobnost kolize
  - $\forall x_1, x_2 \in \mathcal{U}, x_1 \neq x_2: \Pr_{h \in \mathcal{H}}[h(x_1) = h(x_2)] \leq c/m$
  - Příklad: multiply-shift: w-bitová čísla na ℓ-bitová
    - $h_a(x) = \frac{a \cdot x \bmod 2^w}{2^{w-\ell}}$, kde a je liché
    - 2-univerzální (bez důkazu)

$c > 0$ je konstanta

```
#include <stdint.h> //defines uint64_t as unsigned 64-bit integer.
uint64_t hash(uint64_t x, uint64_t l, uint64_t a) {
  // hashes x universally into l<=64 bits using random odd seed a.
  return (a*x) >> (64-l);
}
```

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

- Rychlé vyhodnocení + málo prostoru
  - Typicky výpočet v $O(1)$ z parametrů, které vygenerujeme náhodně na začátku
- Uniformita: $\forall x \in \mathcal{U}\ \forall y \in [m]: \Pr_{h \in \mathcal{H}}[h(x) = y] \leq c/m$
- Univerzalita: malá pravděpodobnost kolize
  - $\forall x_1, x_2 \in \mathcal{U}, x_1 \neq x_2: \Pr_{h \in \mathcal{H}}[h(x_1) = h(x_2)] \leq c/m$

- 2-nezávislost: každé dva prvky se hešují (skoro) nezávisle
  - $\forall x_1, x_2 \in \mathcal{U}, x_1 \neq x_2\ , \forall y_1, y_2 \in [m]:$
    $\Pr_{h \in \mathcal{H}}[h(x_1) = y_1 \& h(x_2) = y_2] \leq c/m^2$
- Obecněji: k-nezávislost pro $k \geq 1$
- Příklad 2-nezávislé rodiny: lineární kongruence
- 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

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

- Tabelace je 3-nezávislá
  - ale ne 4-nezávislá, pokud máme alespoň dvě tabulky

Příště

- Zase hešování ☺
  - Řešení kolizí – tedy jak vypadají ty „koše“
    - S kukačkou
    - Lineárním přidáváním
    - Navíc také: *The power of two choices*