Algoritmy a grafy 1 (BI-AG1), Cvičení č. 7

Randomizace, hešování

Paralelka 104, Úterý 16:15-17:45

Cvičící: Šimon Lomič
lomicsim@fit.cvut.cz

Informace: lomicsim.github.io

Fakulta informačních technologií
České vysoké učení technické v Praze
https://courses.fit.cvut.cz/BI-AG1

(Verze dokumentu: 27. 11. 2018 15:12)

7.1 Generování náhodných čísel

Mějme instrukci `randombit()`, která vrací hodnoty $\{0,1\}$ s rovnoměrnou pravděpodobností nezávisle na výsledku předchozích volání.

(a) Jak za pomocí této instrukce vygenerovat náhodné číslo z intervalu $[0, 2^k)$ pro zadané $k > 0$.

(b) Jak vygenerovat náhodné číslo z intervalu $[0, n), n \in \mathbb{N}$.

- Jaký je maximální počet volání `randombit()`?

7.2 Pravděpodobnostní prostor

- **Diskrétní pravděpodobnostní prostor** je dvojice $(\Omega, \mathbf{P})$, kde
  - Ω je konečná nebo spočetně nekonečná množina **elementárních jevů**.
  - $\mathbf{P} : \Omega \to [0, 1]$ je funkce, která elementárním jevům přiřazuje jejich **pravděpodobnosti** taková, že součet pravděpodobností všech elementárních jevů je roven 1, čili $\sum_{\omega \in \Omega} \mathbf{P}(\omega) = 1$.
- **Jev** je nějaká množina $A \subseteq \Omega$ elementárních jevů.
- **Pravděpodobnost jevu** A je: $\mathbf{P}(A) = \sum_{\omega \in A} \mathbf{P}(\omega)$.

Cvičení: Mějme klasickou šestistěnnou kostku. Hodíme s ní, a pokud padne liché číslo, hodíme ještě jednou

(a) Určete pravděpodobnost, že padne číslo 2.

(b) V jakém pravděpodobnostním prostoru (Ω, P) se tento experiment odehrává?

(c) Uveďte příklady jevů $A \subseteq \Omega$ a určete jejich pravděpodobnost.

7.3 Náhodná veličina, střední hodnota

- **Náhodná veličina** (proměnná) je funkce $X : \Omega \to \mathbb{R}$. Přiřazuje tedy každému elementárnímu jevu $\omega \in \Omega$ nějakou číselnou hodnotu $X(\omega)$.
- **Střední hodnota** $\mathbf{E}[X] = \sum_{x_i} x_i \cdot \mathbf{P}[X = x_i]$ (průměr všech možných hodnot veličiny X vážený jejich pravděpodobnostmi).
- **Čekáme-li na náhodný jev**, který nastane s pravděpodobností p, dočkáme se ho ve střední hodnotě v $1/p$-tém pokusu.

Cvičení:

(a) *Mějme klasickou šestistěnnou kostku. Hodíme s ní, a pokud padne liché číslo, hodíme ještě jednou.*

Určete střední hodnotu čísla, které padne na kostce.

(b) Určete střední hodnotu náhodné veličiny $X =$ počet použití instrukce `randombit()`, při generování náhodného čísla z intervalu $[0, n)$.

7.4 Pravděpodobnost, čekání na jev

(a) Popište algoritmus, který v lineárním čase vygeneruje náhodnou permutaci množiny $\{1, 2, \ldots, n\}$.

(b) Mějme velmi velký soubor (nevejde se do paměti). Navrhněte algoritmus, který vypíše rovnoměrně náhodný řádek a projde celý soubor pouze jednou (dopředu neznáme počet řádků).
   - (Za domácí úkol vyberte rovnoměrně náhodnou k-tici.)

(c) Uvažujte následující experiment: mějme n přihrádek a postupně do nich umisťujme čísla $\{1, 2, \ldots, n\}$ tak, že vždy vybereme rovnoměrně náhodně přihrádku a pokud v ní již něco je, vybíráme znovu. Určete střední počet pokusů volby přihrádky.

7.5 Hešovací tabulka

- **Hešovací tabulka** pro univerzum klíčů $\mathcal{U}$ je množina **přihrádek** $\mathcal{P} = \{0, \ldots, m-1\}$ $(m \ll |\mathcal{U}|)$ a **hešovací funkce** $h : \mathcal{U} \to \mathcal{P}$,
  - Prvek s klíčem $k \in \mathcal{U}$ umístíme do přihrádky $h(k)$.
  - **Kolize** dvou různých klíčů $k_1, k_2 \in \mathcal{U}$ nastane když $h(k_1) = h(k_2)$ (padnou do stejné přihrádky).
  - Při návrhu hešovací tabulky minimalizujeme:
    - ⋆ Čas nutný ke spočtení $h(k)$.
    - ⋆ Velikost tabulky m – pro n prvků ideálně $m = \mathcal{O}(n)$.
    - ⋆ Počet kolizí: aby tabulka fungovala v průměrném případě dobře, h musí rozdělovat $\mathcal{U}$ do přihrádek **rovnoměrně**.

Cvičení:

(a) Jmenujte příklad kdy se vyplatí použít hešovací tabulku oproti poli / bitsetu / vyhledávacímu stromu.

(b) Uvažujme univerzum $\mathcal{U} = \mathbb{N}$. Jaké nevýhody má hešovací funkce $h(k) = k \mod m$?

7.6 Hešování

Nalezněte kolizní páry pro následující hešovací funkce:

(a) $h(k) = (\sum_{i=1}^{n} 12k_i + 37) \mod 13$ na univerzu $\mathcal{U} = \{0, 1\}^n$.

(b) $h(k) = (5k + 7) \mod 13$ na univerzu $\mathcal{U} = \mathbb{N}$.

(c) $h(k) = (\sum_{i=1}^{n} k_i + 3) \mod 13$ na univerzu $\mathcal{U} = \mathbb{N}^n$.

(d) $h(k) = (\prod_{i=1}^{n} i \cdot k_i + 1) \mod 13$ na univerzu $\mathcal{U} = \mathbb{N}^n$.

(e) $h(k) = ((k + 2) \mod 3 + k) \mod 5$ na univerzu $\mathcal{U} = \mathbb{N}$.

(f) Pro hešovací funkci $h(k) = (k \cdot n) \mod m$, kde n, m jsou nesoudělná čísla dokažte, že při vkládání klíčů $0, 1, \ldots, m-1$ do hešovací tabulky velikosti m nedojde k žádné kolizi. **(0.5 b)**

(g) Vložte následující klíče do hešovací tabulky velikosti 5 s hešovací funkcí řetězců: $h(s) = \sum_{i=1}^{size(s)} s[i] \cdot 3^i \mod 5:$

- abba, baba, b, bab

Pro jednoduchost mapujme znaky na čísla: $a = 1, b = 2 \ldots$

7.7 Hešování s otevřenou adresací

- Do každé přihrádky se vejde jen **jeden prvek** (pole $A[0], \ldots, A[m-1]$).
- Hešovací funkce $h(k, i)$ určuje přihrádku klíče k při i-tém pokusu o vložení (postupně zkoušíme $i = 0, 1, \ldots, m-1$). Určuje tzv. **vyhledávací posloupnost** pro každý klíč.

Cvičení: Uvažujte následující hešovací funkci s otevřenou adresací:

$$h(x, i) = (x + i \cdot f(x)) \bmod 7, \qquad \text{kde } f(x) = 2 + (x \bmod 3).$$

Postupně vkládejte klíče 1,4,6,8,11.

7.8 Domácí úkol

(a) Mějme velmi velký soubor (nevejde se do paměti). Navrhněte algoritmus, který vybere náhodnou k-tici řádků a projde celý soubor pouze jednou (každá k-tice musí mít stejnou pravděpodobnost). **(0.5 b)**

(b) Mějme zakořeněný strom T o n vrcholech. Navrhněte lineární algoritmus, který vypíše délku nejdelší cesty v T. **(0.5 b)**

Úkol odevzdejte na příštím cvičení (případně e-mailem).