Hashowanie (ciąg dalszy)

- metoda przeulekania (listy elementów)
- adresowanie otwarte
- podwójne hashowanie

metoda liniowa
metoda kwadratowa $\rightarrow$ m różnych permutacji
(ciąg adresów zależy tylko od $h(e)$)

podwójne hashowanie:

$\rightarrow m^2$ permutacji

Zał.

(dper) Każda permutacja liczb $\underline{m}$ jest z jednakowym prawdopodobieństwem ciągiem kontrolnym.

Tw

(dper) $\Rightarrow$ $\alpha = \frac{n}{m} < 1$ $\Rightarrow$ oczekiwana liczba prób wyszukiwania klucza, którego nie ma, jest $\leq \frac{1}{1-\alpha}$

D-d

Niech p_i – ppd, że zadanie wykonanych i prób.

Chcemy obliczyć $\sum_{i=1}^{m} i \cdot p_i$

Niech q_i – ppd., że wykonamy co najmniej i prób

$$p_i = q_i - q_{i+1}$$

$$\sum_{i=1}^{m} i \cdot p_i = \sum_{i=1}^{m} i(q_i - q_{i+1}) \leq \sum_{i=1}^{\infty} i(q_i - q_{i+1}) = \sum_{i=1}^{m} q_i$$

$$q_1 = 1 \qquad q_2 = \frac{n}{m} \qquad q_3 = \frac{n}{m} \cdot \frac{n-1}{m-1}$$

$$q_i = q_{i-1} \cdot \frac{n-i+2}{m-i+2}$$

$$q_i \leq \left(\frac{n}{m}\right)^{i-1}$$

$$\sum q_i \leq \sum_{i=0}^{\infty} \alpha^i = \frac{1}{1-\alpha}$$

Tw.

(dper) $\Rightarrow$ $\alpha = \frac{n}{m} < 1$ $\Rightarrow$ oczekiwana liczba prób udanego wyszukiwania jest $\leq \frac{1}{\alpha} \ln \frac{1}{1-\alpha}$

Uniwersalne rodziny funkcji hashujących

($\rightarrow$ hashowanie uniwersalne)

Def

Niech H – rodzina funkcji $U \to \underline{m}$

H nazywamy uniwersalną, jeśli

$$\forall k_1, k_2 \in U, k_1 \neq k_2 . \ |\{h \in H \mid h(x) = h(y)\}| \leq \frac{|H|}{m}$$

Przykłady:

1) Niech r: $|U| < m^{r+1}$

m – pierwsza

$\forall a \in [0 .. m^{r+1} - 1]$ definiujem h_a:

$$h_a(x) = \left(\sum_{i=0}^{r} a_i x_i\right) \bmod m$$

x_i – cyfry x w rozwinięciu m-arnym

a_i – cyfry a —— " ——

2) Niech p – l. pierwsza : $p > |U|$

$\forall a \in \mathbb{Z}_p^*$ $b \in \mathbb{Z}_p$ definiujemy $h_{a,b}$:

$$h_{a,b}(x) = ((ax + b) \bmod p) \bmod m$$

Fakt

Rodzina $H_{p,m} = \{ h_{a,b} \mid a \in \mathbb{Z}_p^* \ b \in \mathbb{Z}_p \}$ jest uniwersalna.

Słownik statyczny

operacje: • create
• find

$k_1, \dots, k_n$ – znane klucze

Cele: • pamięć $O(n)$

• czas find $O(1)$

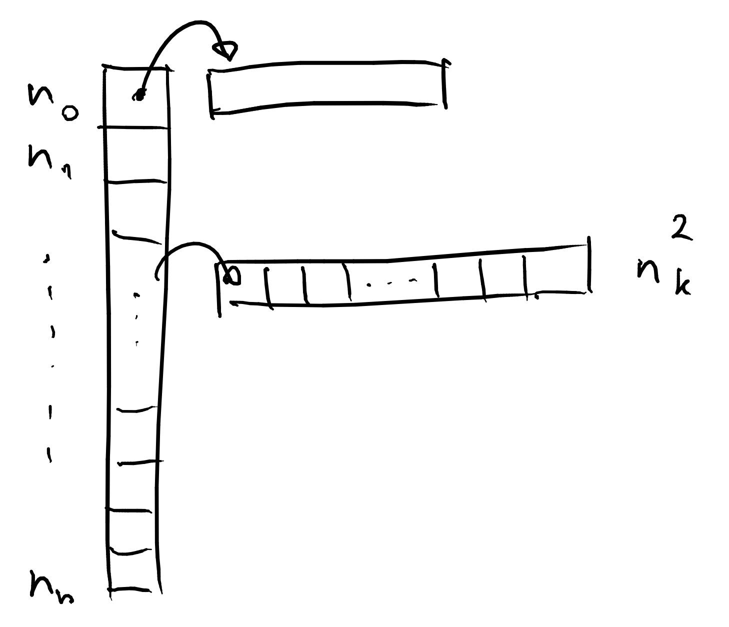
• czas utworzenia: oczekiwany sensownie mały
(tworzenie zrandomizowane)

Konstrukcja (Fredman, Komlos, Szomerédi)

Idea

żeby n kluczy nie konfliktowało ze sobą: $\underline{n^2 \text{ slotów}}$
(paradoks urodzin)

za dużo



Hashowanie dwupoziomowe

• na pierwszym poziomie funkcja hashująca rozrzuca klucze tak, żeby $\sum n_i^2 = O(n)$

n_i – # kluczy z kubełka i

• drugi poziom:
  • kubełek i: tablica rozmiaru n_i^2, i-ta funkcja hashująca rozlokowuje klucze bezkolizyjnie

Lemat (nierówność Markowa)

Zał.: X – zm. losowa, $t>0$

$$\Pr[|X| \geq t] \leq \frac{E[|X|]}{t}$$

Fakt

Z ppd $\geq \frac{1}{2}$ funkcja wybrana losowo z rodziny uniwersalnej umieszcza bezkonfliktowo $n=\sqrt{m}$ kluczy w tablicy m-elementowej

$k_1, \dots, k_n$ $\quad n=\sqrt{m}$

$$X_{ij} = \begin{cases} 1 & k_i \text{ konfliktuje z } k_j \\ 0 & \text{w p.p.} \end{cases}$$

$$X = \sum_{1 \leq i \leq j \leq n} X_{ij} \qquad \Pr[X_{ij}=1] \leq \frac{1}{m} \quad \text{– rodz. uniwersalna}$$

$$E[X] = \sum_{1 \leq i \leq j \leq n} E[X_{ij}] = \binom{n}{2} \cdot \frac{1}{m} = \frac{n^2}{2m} = \frac{1}{2}$$

Fakt

Z ppd $\geq \frac{1}{2}$ losowo wybrana funkcja z rodziny uniwersalnej rozrzuca n kluczy do tablicy m elementowej tak, że zachodzi:

$$\sum_{i=0}^{m-1} n_i^2 < 4n$$

n_i – # kluczy w i-tym kubełku

Chcemy obliczyć $E[\sum_{i=0}^{n-1} n_i^2]$

Potrafimy: $E[\sum_{i=0}^{n-1} n_i] = n$

$$E\left[\sum_{i=0}^{n-1} \binom{n_i}{2}\right]$$

$\binom{n_i}{2}$ – # kolizji w i. kubełku

$\sum \binom{n_i}{2}$ – suma wszystkich kolizji

$$X = \sum X_{ij}$$

$$X_{ij} = \begin{cases} 1 & h(k_i) = h(k_j) \\ 0 & \end{cases}$$

$$E[X] = \binom{n}{2} \frac{1}{n} = \frac{n-1}{2}$$

$$E\left[\sum \binom{n_i}{2}\right] = E(X) = \frac{n-1}{2}$$

$$\dots \; E\left(\sum n_i^2\right) < 2n$$

$$\Pr\left[\sum n_i^2 \geq 4n\right] \overset{?}{\leq} \frac{2n}{4n} = \frac{1}{2}$$

Jaka jest oczekiwana liczba prób funkcji 2. poziomu?