

Jeśli  $a_0, \dots, a_{n-1}, b_0, \dots, b_{n-1} \in \mathbb{N}$ . Obliczenia możemy wykonywać w  $\mathbb{Z}_p$ ,  $p \in \mathbb{P}$ , gdzie  $p$  jest odpowiednio duże i w  $\mathbb{Z}_p$  jest  $n$ -ty pierwotny pierwiastek z jedności, który ma własność:

$$(*) \quad \forall_{\substack{k \geq 0 \\ n \nmid k}} \sum_{j=0}^{n-1} (\omega_n^k)^j = 0$$

FAKT: Niech  $n, n$  - potęgi liczby 2 ( $\neq 1$ )

$$m = 2^{n/2} + 1$$

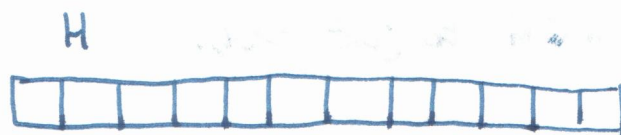
Wówczas  $n$  i  $n$  są odwracalne w pierścieniu reszt mod  $m$  oraz  $\omega$  jest  $n$ -tym pierwotnym pierwiastkiem z jedności spełniającym (\*).

## Haszowanie

$U$ -universum kluczy,  $S \subseteq U$

operacje porządane:

- insert
- delete
- search



$H[i] - \text{true iff } i \in S$

nierealne, gdy  $|U|$  - wielkie

$$m = O(|S|)$$

$n$

Chcemy mieć funkcję  $h: U \rightarrow \{0, \dots, m-1\}$   
 $\nwarrow$   $f$ -ga haszująca

Kolejna - synteza, w której  $\forall_{\substack{k, l \in S \\ k \neq l}} h(k) \neq h(l)$

Jaki redzić sobie z kłótniami:

Met. pamiętanie kluczy

- nawlekanie ( $H[i]$  - lista kluczy z  $S$ , dla których  $h$  przyjmuje wartości  $i$ )
- adresowanie otwarte

Chcieli byśmy, by  $h$  rozmieszczała klucze równomiernie.

Co by było, gdyby  $h$  rozmieszczała losowo? - analogia ze schematami urnowymi?

Jakie byłaby oceniana długość listy (np. listy  $H[0]$ )?  $\phi: O(n/m)$

$X_i = \begin{cases} 1, & \text{jeśli klucz upada do listy } 0\text{-tej} \\ 0, & \text{w.p.p.} \end{cases}$

$X_1, X_2, \dots, X_n$

$X = \sum_{i=1}^n X_i$  - długość listy  $H[0]$ ,  $E[X] = \sum_{i=1}^n E[X_i] = \frac{n}{m}$

Zatem jeśli  $n \leq m$  to jest  $O(1)$ .



Jakie jest oceniana wartość długości najdłuższej listy?

Gdy  $n = m$ , to  $\frac{\log n}{\log(\log(n))}$

Użyjemy f-gi pseudolosowych:

- deterministyczne
- „zachowują się” podobnie do losowych
- szybko ( $h$  w  $O(1)$ ) wyliczalne

## Przykłady takich f-gi:

- $h(k) = k \bmod m$

OSTROŻNOŚĆ W WYBORZE ~~m~~ m; zalecane - m - liczba pierwsza oddzielona od potęgi 2

- $h(k) = \lfloor m(k \cdot A - \lfloor k \cdot A \rfloor) \rfloor$ , gdzie A - ustalona liczba z (0,1)

dobry A:  $A = \frac{\sqrt{5} - 1}{2}$ , tu m może być potęgą 2-ki

## Adresowanie otwarte:

- klucze pamiętam w tablicy H

- k chcemy zapamiętać w  $H[h(k)]$ , a jeśli jest konflikt to stosujemy jakąś strategię znajdowania wolnej lokalizacji klasyczne strategii:

- metoda liniowa

$$h(k, i) = (h'(k) + i) \bmod m, \quad i \text{ ozn. nr próby}$$

Najpierw k próbujemy wstawić w  $h(k, 0)$ , potem  $h(k, 1) \dots h(k, i)$

- metoda kwadratowa

$$h(k, i) = (h'(k) + c_1 i + c_2 i^2) \bmod m$$

$c_1, c_2$  - pewne stałe dobrane tak, by zachodził

wzorek (dop):  $h(k, 0), h(k, 1), \dots, h(k, m-1)$  - permutacja

liczb  $0, \dots, m-1$

- podwójne haszowanie

$$h(k, i) = (h_1(k) + i h_2(k)) \bmod m, \quad h_1, h_2 \text{ - f-gi haszujące}$$

$\forall k \quad h_2(k)$  - względnie pierwsze z m



met. liniowa:

$m$  permutacji wg których są próbowane lokalizacje

met. kwadratowa:

$m$  permutacji

met. harmoniczna podw:

$m^2$  permutacji

Jaki dobierać  $m$ ?

Gdy  $n \gg m$ , to robimy przeszerzenie o dwie razy większą tablicę

Adresowanie otwarte c.d.

24.05.2018

FAKT Przy założeniu  $(dper)^*$  i  $\alpha = \frac{n}{m} < 1$  <sup>nsp. zapamiętanie tablicy</sup> liniowa linie przy wyszukaniu przy wyodrębnieniu klucza zachowanym fragmentem jest  $< 1 - \alpha$

\* (~~klucza~~  $dper$ ) - która permutacja linb  $0, \dots, m-1$  jest jednolito ppb jako ciąg  $h(k, 0), h(k, 1), \dots, h(k, m-1)$

Niech  $p_i$  - ppb wykonanie <sup>niewdanych</sup> i prób

$$Szukamy (x) = \sum i p_i$$

Niech  $q_i$  - ppb wykonanie co najmniej <sup>niewdanych</sup> i prób

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

$$(x) = \sum_i i (q_i - q_{i+1}) = \sum_i q_i$$

$$q_1 = \frac{n}{m}, \quad q_2 = \frac{n}{m} \left( \frac{n-1}{m-1} \right) \leq \left( \frac{n}{m} \right)^2, \quad \dots, \quad q_i \leq \left( \frac{n}{m} \right)^i = \alpha^i$$

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

## FAKT

Dla wyliczenia zalogowanego powrotu mamy  $\leq \frac{1}{\alpha} \ln\left(\frac{1}{1-\alpha}\right) + \frac{1}{\alpha}$

## Def

Niech  $H$  będzie rodziną f-gi hashujących z  $U$  do  $\{0, \dots, m-1\}$ . Rodzinę  $H$  nazywamy uniwersalną jeśli:

$$\forall_{\substack{x, y \in U \\ x \neq y}} |\{h \in H : h(x) = h(y)\}| \leq \frac{|H|}{m} \quad (\text{innymi słowy jest mała})$$

## Przykłady rodzin uniwersalnych:

- Niech  $m$ , t.j.  $m \in \mathbb{P}$  oraz  $m^{r+1} > |U|$

$0 \leq a < m^{r+1}$  przedstawiamy w systemie  $m$ -owym  $(a_0, a_1, \dots, a_r)$   
to też jest w sys  $m$ -owym

definiujemy:

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

(dod, że to jest r. uniwersalna poniżej)

- Niech  $p \in \mathbb{P}$ ,  $p > |U|$ ,  $m$  - wielkość tablicy hashującej

$$\forall_{\substack{a \in \mathbb{Z}_p^* \\ b \in \mathbb{Z}_p}} \text{ Niech } h_{a,b}(x) = ((ax+b) \bmod p) \bmod m$$

## FAKT

Zbiór f-gi  $H_{p,m} = \{h_{a,b} : a \in \mathbb{Z}_p^*, b \in \mathbb{Z}_p\}$  jest rodziną uniwersalną

## D-d

$$h_{a,b}(x) = ((ax+b) \bmod p) \bmod m, \text{ niech } h'_{a,b} = (ax+b) \bmod p$$

Niech  $k, l$  dowolne różne liczby  $s = h'(k)$ ,  $t = h'(l)$

Spostereowanie:

$$\bullet \quad \underline{s \neq t}$$

gdyby

$$s = t, \text{ to}$$

$$s = (ak+b) \bmod p, \quad t = (al+b) \bmod p$$

$$0 = t - s = a(k-l) \bmod p, \text{ ale } p \nmid a, \text{ bo } a \in \{1, \dots, p-1\}$$

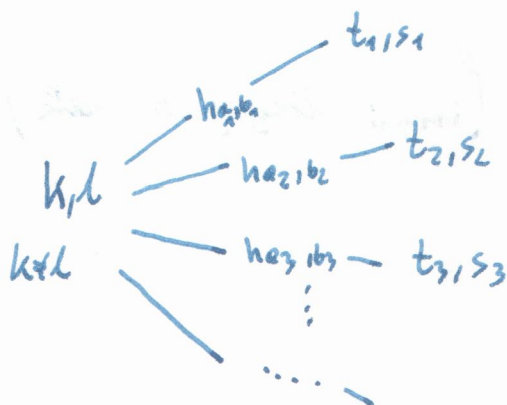
$$\text{oraz } p \nmid (k-l), \text{ bo } k \neq l \text{ i } p > |U|$$

- Korzystając z funkcji  $h_{a,b}$  przedstawiamy parę  $k, l$  na inną parę  $t, s$

$$t = (ak + b) \bmod p \quad s = (el + b) \bmod p$$

$$t - s = a(k - l) \bmod p \Rightarrow a = ((k - l)^{-1} \bmod p)(t - s) \bmod p$$

$$b = (t - ak) \bmod p$$



Różnych funkcji  $h_{a,b}$  jest  $(p-1)p$ . Różnych par jest  $p(p-1)$ , bo  $t \neq s$  i  $t, s \in \mathbb{Z}_p$ .  
tyle możliwości na a  
tyle na b

Istnieje więc bijekcja między funkcjami  $h_{a,b}$  a parami  $t, s$  (t. zn.  $t \neq s$ ).

- Zatem ppb kodzigi będący  $k, l$  przy haszowaniu jest równy ppb-stu wydoszowania pary  $s, t$  takiej, że  $t \equiv s \bmod m$  spośród wszystkich par  $t$  i  $s$  takich że  $t \neq s$ .

Dla danego (ale ustalonego)  $s$  liczb takich że  $t \equiv s \bmod m$  i  $t \neq s$  jest równa  $\lceil \frac{p}{m} \rceil - 1$ .

$$\lceil \frac{p}{m} \rceil - 1 \leq \frac{p-1+m}{m} - 1 = \frac{p-1}{m}$$

- Ponieważ różnych  $t$  dla danego  $s$  jest  $p-1$ , więc ppb, że  $s$  i  $t$  kodujące jest  $\leq \frac{1}{m}$ .



# Słownik statyczny

Klucze:  $k_1, \dots, k_m$  - znane

Operacje: FIND( $k_i$ )

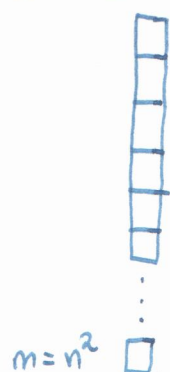
nie chcemy wykonywać DELETE, INSERT

Cel: stworzenie struktury (słownika):

- rozmiar  $O(n)$
- czas FIND  $O(1)$
- czas tworzenia struktury wielomianowy  
    ↑  
    Orelatywny

WTF? O CO TU  
CHODZI???

Idea: HASZOWANIE DOWPOZIOKOWE



$k_1, \dots, k_m$

$X = \# \text{kdzi}$

$X = \sum_{i \neq j} x_{ij}$

$X = \begin{cases} 1, & \text{jeśli } k_i \text{ koliduje z } k_j \\ 0, & \text{w p.p.} \end{cases}$

$$E[X] = \sum_{i \neq j} E[x_{ij}]$$

$$\binom{n}{2} \frac{1}{m} = \frac{n(n-1)}{2} \frac{1}{m}$$

teraz uźeć idea:

tablica pierwotna



rozmiar  $n_j^2$ , gdzie  $n_j$  to rozmiar kubelka j-tego

tablica ubrane