


Haszowanie 

# Funkcja Haszująca

Jest to taka funkcja  $h$ , która mapuje elementy z uniwersum  $U$  w  $\mathbb{N}_{<m}$ , gdzie  $m$  jest rozmiarem danej tablicy haszującej.

Daje dobre efekty gdy jest liczbą pierwszą!

Zadaniem tej funkcji jest zapisywanie elementów w odpowiednich miejscach tablicy tak, by znalezienie tego elementu było szybkie i szanse kolizji były jak najmniejsze.

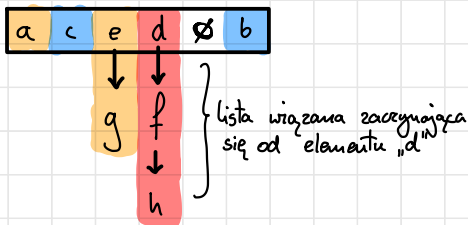
Kolizja to zjawisko gdy funkcja haszująca wskaże nam miejsce w tablicy, w którym już został wcześniej wstawiony element.

Prawdopodobieństwo kolizji dobrej funkcji haszującej definiujemy na  $\frac{1}{m}$ .

## Co robimy gdy trafimy na kolizję?

### Listowanie (Separate Chaining)

Do elementów w których wystąpiła kolizja dotężamy nową tworząc listę wiążaną.



Średni koszt wynosi:  $\Theta(1 + \frac{m}{n})$

$\alpha = \frac{m}{n}$  nazywany współczynnikiem wypełnienia gdzie:

$m$  - liczba elementów umieszczonych w tablicy  
 $n$  - rozmiar tablicy

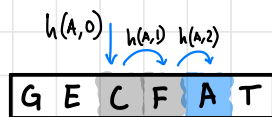
Dlatego też warto jest zwiększać rozmiar tablicy ( $n$ ) za każdym razem gdy współczynnik wypełnienia  $\alpha$  przekroczy jakiś próg np.  $\frac{2}{3}$ , żeby utrzymywać wydajność struktury

## Adresowanie Otwarte

Teraz gdy następuje kolizja - szukamy następnego wolnego miejsca.

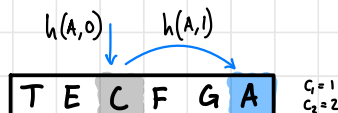
- Metoda liniowa

$$h(x, i) = (h'(x) + i) \bmod n$$



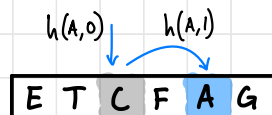
- Metoda kwadratowa

$$h(x, i) = (h'(x) + c_1 i^2 + c_2 i) \bmod n$$



- Podwójne haszowanie

$$h(x, i) = (h'(x) + i \cdot h''(x)) \bmod n$$



Metoda liniowa jak i podwójne haszowanie mają tendencję do lepszych rezultatów, ponieważ zapisują dane bliżej siebie w pamięci, co powoduje wydajniejsze wykorzystanie zasobów pamięci podręcznej.

Przy założeniu, że  $\alpha = \frac{m}{m} < 1$  oczekiwana liczba kolizji pod rząd jest  $\leq \frac{1}{1-\alpha}$ .

np. dla  $\frac{1}{2}$  wypełnionej tablicy mamy  $\frac{1}{1-\frac{1}{2}} = \frac{1}{\frac{1}{2}} = 2$

dla  $\frac{9}{10}$  wypełnionej tablicy mamy  $\frac{1}{1-\frac{9}{10}} = 10$

## Uniwersalne rodziny funkcji haszujących

Wyobraźmy sobie, że zamiast jednej funkcji haszującej mamy zbiór  $H$  funkcji takich, że prawdopodobieństwo kolizji ze sobą jest nie większe niż  $\frac{1}{m}$ .

$$|\{h \in H: h(x) = h(y)\}| \leq \frac{|H|}{m}$$

Przykład takiej rodziny:

Gwiazdka oznacza, że nie uwzględniamy zera  
Zbiór liczb całkowitych modulo  $p$

Dla dowolnych  $a \in \mathbb{Z}_p^*$ ,  $b \in \mathbb{Z}_p$  zdefiniujmy funkcję haszującą:

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

Wtedy rodzina  $H_{a,b}$ :

$$H_{a,b} = \{h_{a,b}: a \in \mathbb{Z}_p^* \wedge b \in \mathbb{Z}_p\}$$

Oczekiwana liczba kluczy podczas wstawiania  $n$  kluczy do tablicy hashującej rozmiar  $m$  gdy wykorzystujemy losowe funkcje hashujące z uniwersalnej rodziny funkcji hashujących:

$$E[X] = \binom{m}{2} \cdot \frac{1}{m}$$

Wartość oczekiwana      liczba par kolizji      ↑      prawdopodobieństwo kolizji

W przypadku gdy  $m = n^2$  takie prawdopodobieństwo  $\leq \frac{1}{2}$

$$E[X] = \binom{m}{2} \frac{1}{m^2} = \frac{m^2 - m}{2} \cdot \frac{1}{m^2} = \frac{m^2 - m}{2m^2} \leq \frac{1}{2}$$

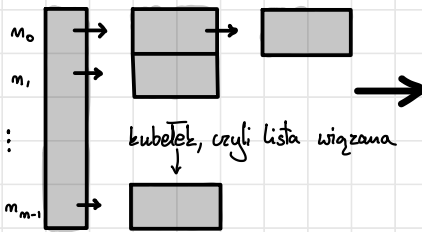
# Słownik Statyczny



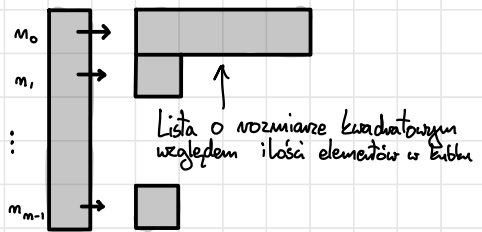
- Tylko wyszukiujemy dane (nie ma **insert** ani **delete**)

## Algorytm

1. Stosujemy podwójne haszowanie wylosowane z rodziny funkcji haszujących  $H$ .
2. Rozrzucamy elementy do  $m$  kubków.
3. Rozrzucenie jest poprawne wtedy gdy  $\sum m_i^2 < 4m$ , gdzie  $m_i$  oznacza liczbę kluczy wrzuconych do  $i$ -tego kubka.
4. Jeśli rozrzucenie jest niepoprawne - idź do 1.
5. Wylosuj funkcję haszującą dla każdego kubka i zastosuj ją by hashować każdy kubek do tablicy  $m_i^2$ -elementowej, gdzie  $m_i$  to rozmiar  $i$ -tego kubka.



Kroki 1-4



Krok 5

## Oczekiwany czas budowy słownika statycznego

$$\begin{aligned} E\left[\sum_{i=1}^m m_i^2\right] &= E\left[\sum m_i(m_i - 1) + m_i\right] = 2E\left[\sum \frac{m_i(m_i - 1)}{2}\right] + E\left[\sum m_i\right] = \\ &= m + 2E\left[\sum \binom{m_i}{2}\right] = m + m - 1 = 2m - 1 \leq 2m \end{aligned}$$

Wstawiliśmy  $m$  kluczy

liczba wszystkich kolizji

$$E\left[\sum \binom{m_i}{2}\right] = \frac{1}{m} \binom{m}{2} = \frac{m^2 - m}{2} \cdot \frac{1}{m} = \frac{m^2 - m}{2m} = \frac{m-1}{2}$$

pierwszy poziom ma rozmiar  $m$

Zastosujemy nierówność Markowa:

$$P(x \geq t) \leq \frac{E[x]}{t}$$

$$P(\sum m_i^2 \geq 4m) \leq \frac{2m}{4m} = \frac{1}{2} \Rightarrow P(\sum m_i \leq 4m) \geq \frac{1}{2}$$

↑

Zatem oczekujemy dwóch prób, żeby wylosować funkcję hashującą spełniającą ten warunek poprawnego słownika statycznego.