

# Haszowanie

- **złożoność obliczeniowa:**  $O(h + 1)/O(h + 1 + n) \leftarrow h$  to złożoność funkcji haszującej
- haszowanie - tablicowanie za pomocą funkcji haszującej, różniące się tym, że zamiast umieszczania wartości w indeksie odpowiadającej liczbie to umieszczamy w wartość o indeksie będącym wyjściem funkcji haszującej dla tej wartości
  - w przypadku kolizji haszowania (nieunikniona) przeszukujemy liniowo tablicę w poszukiwanie następnego wolnego miejsca  $\rightarrow$  stąd pesymistyczna złożoność to  $O(h + n)$
- maskowanie bitowe - haszowanie, tyle że wynikiem funkcji haszującej jest liczba w systemie binarnym (dwójkowym), gdzie 1 oznacza, że  $i$ -ta liczba należy do podzbioru, a 0 nie  $\rightarrow$  patrz generowanie podzbiorów

## Algorytm wstawiania łańcucha do tablicy haszowanej

### Wejście:

$s$	łańcuch do umieszczenia w tablicy haszowanej
$T$	tablica łańcuchów
$n$	rozmiar tablicy, $n \in N$

### Wyjście:

Tablica  $T$  z wstawionym łańcuchem  $s$ , jeśli było dla niego wolne miejsce.

### Zmienne pomocnicze:

$i$	indeks, $i \in N$
$h$	przechowuje wartość haszu, $h \in N$
$hf(x, n)$	oblicza indeks w $T$ na podstawie łańcucha $x$ i liczby komórek $n$ .

### Lista kroków dla wersji z duplikatami:

K01:	$h \leftarrow hf(s, n)$	obliczamy indeks początkowy
------	-------------------------	-----------------------------

K02:	$i \leftarrow h$	przeszukiwanie tablicy rozpoczynamy od wyliczonego indeksu
K03:	<b>Jeśli</b> $T[i] = ""$ , <b>to</b> $T[i] \leftarrow s$ <b>zakończ</b>	w puste miejsce zapisujemy łańcuch
K04:	$i \leftarrow (i + 1) \bmod n$	przechodzimy do następnej komórki
K05:	<b>Jeśli</b> $i = h$ , <b>to zakończ</b>	Jeśli wróciliśmy w to samo miejsce, to kończymy
K06:	<b>Idź do</b> kroku K03	inaczej kontynuujemy pętlę

### Lista kroków dla wersji bez duplikatów:

K01:	$h \leftarrow hf(s, n)$	obliczamy indeks początkowy
K02:	$i \leftarrow h$	przeszukiwanie tablicy rozpoczynamy od wyliczonego indeksu
K03:	<b>Jeśli</b> $T[i] = ""$ , <b>to</b> $T[i] \leftarrow s$ <b>zakończ</b>	w puste miejsce zapisujemy łańcuch
K04:	<b>Jeśli</b> $T[i] = s$ , <b>to zakończ</b>	jeśli w tablicy już jest łańcuch $s$ , to kończymy bez wstawiania
K05:	$i \leftarrow (i + 1) \bmod n$	przechodzimy do następnej komórki
K06:	<b>Jeśli</b> $i = h$ , <b>to zakończ</b>	Jeśli wróciliśmy w to samo miejsce, to kończymy
K07:	<b>Idź do</b> kroku K03	inaczej kontynuujemy pętlę

## Algorytm wyszukiwania łańcucha w tablicy haszowanej

### Wejście:

$s$	łańcuch do wyszukania w tablicy haszowanej
$T$	tablica łańcuchów
$n$	rozmiar tablicy, $n \in \mathbb{N}$

### Wyjście:

Indeks łańcucha w  $T$  lub -1, jeśli łańcucha nie ma w  $T$ .

### Zmienne pomocnicze:

$i$	indeks, $i \in \mathbb{N}$
$h$	przechowuje wartość haszu, $h \in \mathbb{N}$

$hf(x, n)$	oblicza indeks w $T$ na podstawie łańcucha $x$ i liczby komórek $n$ .
------------	---

### **Lista kroków:**

K01:	$h \leftarrow hf(s, n)$	obliczamy indeks początkowy
K02:	$i \leftarrow h$	przeszukiwanie tablicy rozpoczynamy od wyliczonego indeksu
K03:	<b>Jeśli <math>T[i] = ""</math>, to zakończ z wynikiem -1</b>	brak łańcucha
K04:	<b>Jeśli <math>T[i] = s</math>, to zakończ z wynikiem <math>i</math></b>	łańcuch odnaleziony
K05:	$i \leftarrow (i + 1) \bmod n$	przechodzimy na następną pozycję
K06:	<b>Jeśli <math>i = h</math>, to zakończ z wynikiem -1</b>	powrót na pozycję wyjściową, czyli brak łańcucha
K07:	<b>Idź do kroku K03</b>	inaczej kontynuujemy pętlę