

# Хешування строки. Хеш-функція

Котенко Андрій ФВЕ



# Вступ

- Хеш-функція – функція, що перетворює масив вхідних в бітову стрічку визначеної довжини (ціле число, стрічка).
- Тобто можна вважати це присвоєнням унікального значення: хеш-кода, хеш-суми, хеш.

# Пошук елементів

- Ототожнюючи хеш з індексом масива, можна проводити швидкий пошук (  $O(1)$  ).
- Умови на використання хеш-функцію:
  - легко розраховувана
  - рівномірний розподіл хешів
  - детермінованість

# Хеш-стрічки

- Візьмо хеш-функцію як остачу від суми ASCII значень:

$\text{hash} = \text{Sum}(\text{for } i \text{ in } \text{len}: \text{ASCIIval}(\text{str}(i)) ) \bmod N$

щоб уникнути спільних дільників візьмо  $N$  простим

Index				
0				
1				
2	abcdef	bcdefa	cdefab	defabc
3				
4				
-				
-				
-				
-				

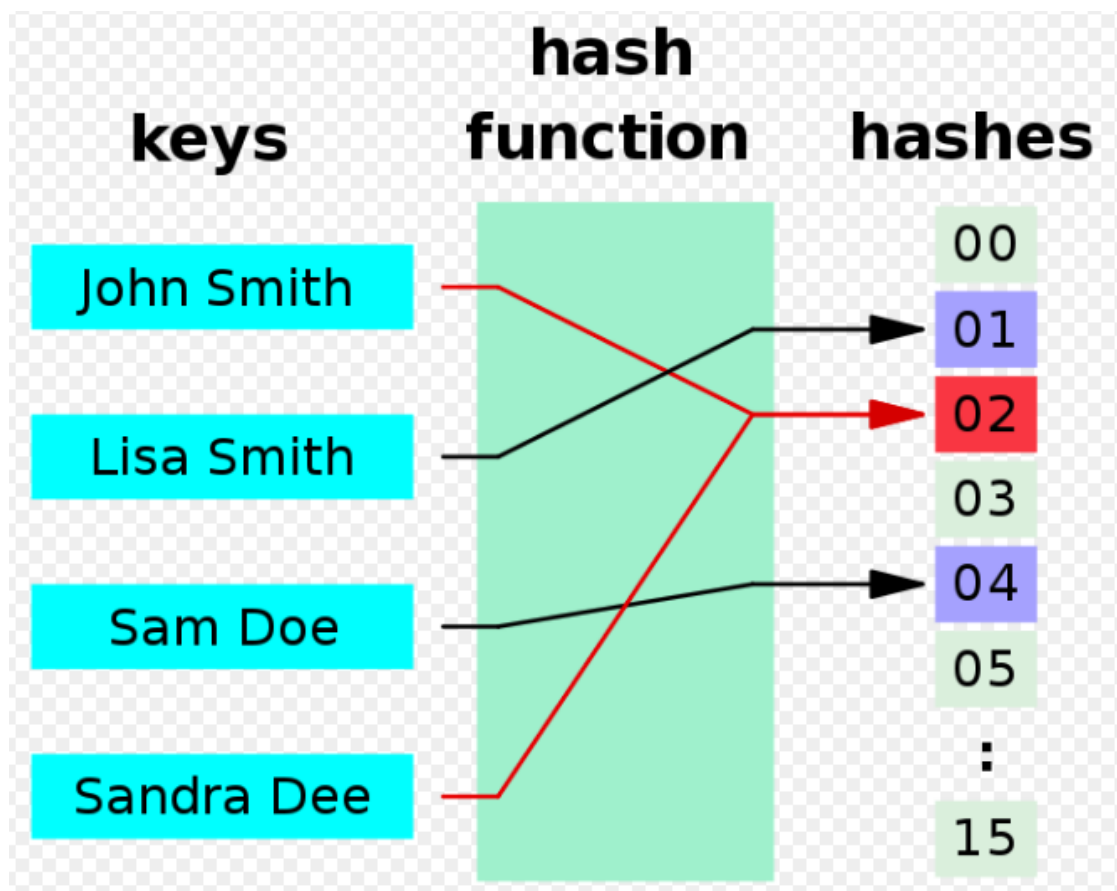
# Хеш стрічки

- Модифікуємо хеш-функцію беручи до уваги положення літери:  
hash =  
Sum( for i in len:  
     $i * \text{ASCIIval}(\text{str}(i)) \bmod N$

Index	
0	
1	
-	
-	
-	
11	defabc
12	
13	
14	cdefab
-	
-	
-	
-	
23	bcdefa
-	
-	
-	
38	abcdef
-	
-	

# Колізії

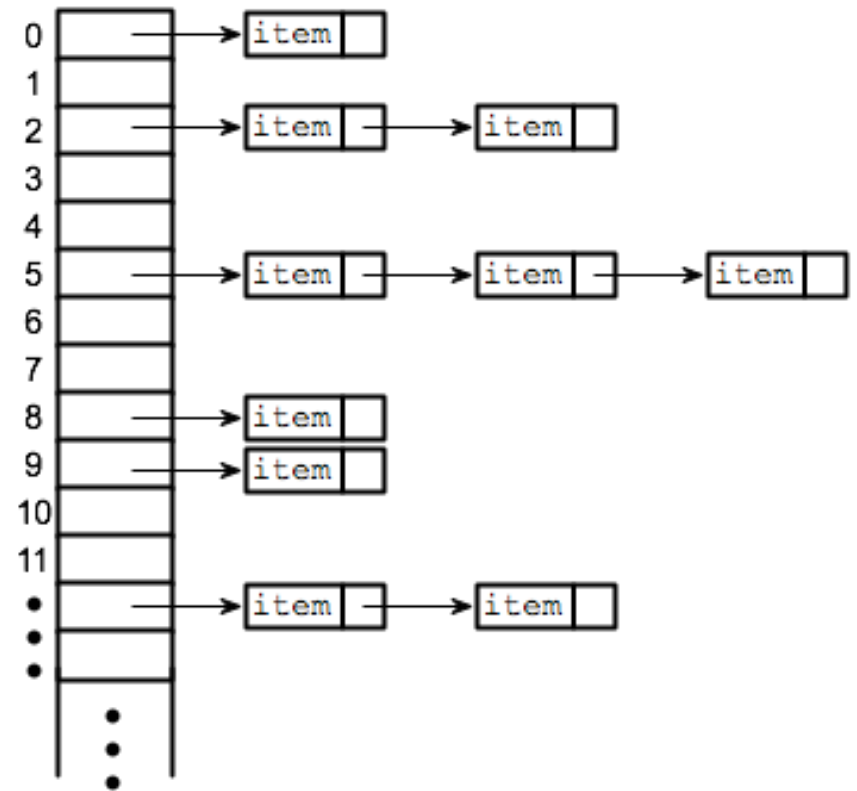
- Колізією називається присвоєння двом чи більше різним ключам одного хеша



# Розв'язання колізій

Метод ланцюгів  
(закрита адресація):

- Елементи з одним хешом об'єднуються в зв'язаний список.
- При кожній колізії елемент додається в початок списку.
- Пошук та видалення вимагаються лінійного пошуку по списку.



# Розв'язання колізій

Відкрита адресація

- В кожній комірці – вказівник на значення
- При колізії виконується пошук вільної комірки.

Наприклад, якщо комірка 6 зайнята, перевіряються 7,8,9 ... (один з методів)



# Поліноміальний хеш

Стрічка str:  $\text{strlen}(\text{str}) = n + 1$

$\text{str}[i]$  – код символу

$$\begin{aligned} \text{hash}(\text{str}[0:n]) &= s[0] + p * s[1] + p^2 * s[2] + \dots \\ &\dots + p^n * s[n] \end{aligned}$$

- Хеш однакових стрічок буде однаковим
- $\text{hash}(\text{str}[0:n]) = \text{hash}(\text{str}[0:k-1]) + p^k * \text{hash}(\text{str}[k:n])$

# Поліноміальний хеш

Можна порахувати хеши підстрічок:

```
hs[0] = s[0];  
for (int i = 1; i < n; i++)  
{  
    hs[i] = hs[i - 1] + pow[i] * s[i];  
}
```

# Поліноміальний хеш

Тоді хеш будь-якої підстрічки  
розраховуватиметься:

```
long Hash(long[] h, int left_ind, int right_ind)
{
    long result = h[right];
    if ( left_ind > 0) result -= h[left_ind - 1];
    return result;
}
```

# Застосування

- Порівняння підстрок:

$\text{Hash}(hs, L, R) * \text{pow}[X] == \text{Hash}(ht, X, Y) * \text{pow}[L]$

- Шукати підстроку в строці:

```
for (int i = 0; i + m <= n; i++)  
{  
    if (Hash(hs, i, i + m - 1) == ht * pow[i]) {гульня}  
}
```

де,  $ht$  – хеш шуканої підстрічки



**Дякую за увагу!**