

Tablice z haszowaniem — wprowadzenie

W wielu przypadkach potrzebujemy struktury danych, która będzie umożliwiała przeprowadzanie tzw. „operacji słownikowych”, tj. operacji INSERT, SEARCH i DELETE. Tablice z haszowaniem to struktury, które efektywnie implementują słowniki:

- Oczekiwany czas wstawiania elementu do tablicy z haszowaniem wynosi $O(1)$ (pod pewnymi konkretnymi warunkami).
- W przypadku pesymistycznym możemy mieć jednak do czynienia ze złożonością $\Theta(n)$.

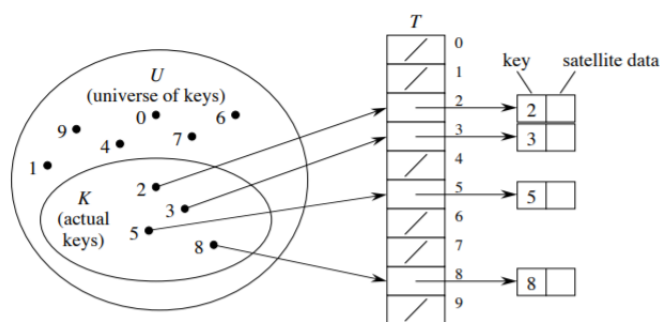
Tablice z haszowaniem są uogólnieniem „zwykłych” tablic:

- W przypadku zwykłych tablic element o kluczu k jest przechowywany na k -tej pozycji.
- A zatem mając klucz k , identyfikowany przez niego element znajdziemy na pozycji k . Taką metodę przechowywania elementów nazywamy *adresowaniem bezpośrednim*.
- Adresowanie bezpośrednie możemy wykorzystać jeżeli możemy sobie pozwolić na alokację tablicy, której rozmiar odpowiada rozmiarowi uniwersum kluczy.

Tablice z haszowaniem są rozwiązaniem, w przypadku, w którym nie możemy (lub nie chcemy) zaalokować tak dużej tablicy:

- Tablica z haszowaniem przydaje się, kiedy liczba kluczy *faktycznie używanych* jest znacznie mniejsza niż liczba wszystkich możliwych kluczy.
- Rozmiar tablicy z haszowaniem jest proporcjonalny do liczby faktycznie używanych kluczy.
- Mając dany klucz k , nie umieszczamy elementu przez niego identyfikowanego na pozycji k , a obliczamy wartość pewnej *funkcji od k* , którą wykorzystujemy do znalezienia odpowiedniego miejsca w tablicy. Funkcję tę nazywamy „funkcją haszującą”.

Adresowanie bezpośrednie



Operacje słownikowe są trywialne. Złożoność: $O(1)$.

Tablice z haszowaniem

Element o kluczu k umieszczamy na pozycji $h(k)$ wyliczonej za pomocą funkcji h :

$$h : U \rightarrow \{0, 1, \dots, m-1\}$$

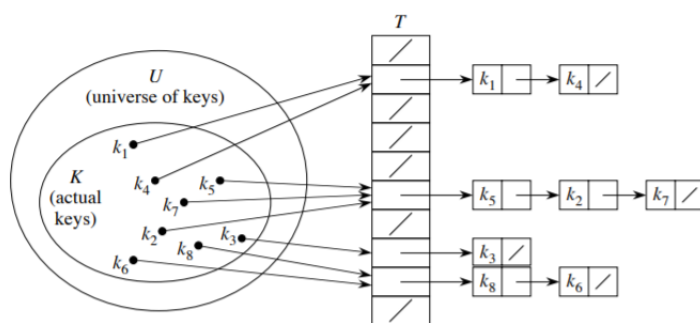
takiej, że $h(k)$ jest poprawnym indeksem tablicy T .

Przykładowe funkcje haszujące:

- Haszowanie modularne: $h(k) = k \bmod m$,
- Haszowanie przez mnożenie: $h(k) = m(kA \bmod 1)$, gdzie $kA \bmod 1 = kA - \lfloor kA \rfloor$ jest częścią ułamkową kA . Szczególnie dobra wartość A to $A = (\sqrt{5} - 1)/2$.

Konflikty/kolizje: Występują gdy kilka kluczy zostaje zmapowanych na tę samą pozycję w tablicy.

Metoda łańcuchowa — rozwiązywanie konfliktów za pomocą list



```

1 CHAINED-HASH-INSERT(T, x)
2   insert x at the head of list T[h(key[x])]

```

Czas operacji: $O(1)$, nawet w przypadku pesymistycznym (przy założeniu, że wstawianego elementu nie ma jeszcze w liście – sprawdzenie tego wymagałoby dodatkowego wyszukiwania).

```
1 CHAINED-HASH-SEARCH(T, k)
2   search for an element with key k in list T [h(k)]
```

Czas operacji jest proporcjonalny do długości listy elementów przypadających na pozycję $h(k)$.

```
1 CHAINED-HASH-DELETE(T, x)
2   delete x from the list T [h(key[x])]
```

Rozwiązywanie konfliktów za pomocą adresowania otwartego

- adresowanie liniowe: $h(k, i) = (h'(k) + i) \bmod m$
- adresowanie kwadratowe: $h(k, i) = (h'(k) + c_1i + c_2i^2) \bmod m$
- haszowanie dwukrotne: $h(k, i) = (h_1(k) + i \cdot h_2(k)) \bmod m$.

```
1 HASH-SEARCH(T, k)
2   i := 0
3   repeat
4     j := h(k, i)
5     if T[j] = k
6       then return j
7     i := i + 1
8   until T[j] = NIL or i = m
9   return NIL
```

Funkcja HASH-SEARCH zwraca indeks pozycji w tablicy zawierający klucz k lub NIL, jeżeli wyszukiwanie się nie powiodło.

```
1 HASH-INSERT(T, k)
2   i := 0
3   repeat
4     j := h(k, i)
5     if T[j] = NIL
6       then T[j] := k
7       return j
8     else
9       i := i + 1
10  until i = m
11  error "hash table overflow"
```

Zadanie 1

Dla danych kluczy (4322, 1334, 1471, 9679, 1989, 6171, 6173, 4199) i funkcji haszującej $x \bmod 10$, które z poniższych zdań są prawdziwe?

1. 9679, 1989, 4199 haszują do tej samej wartości
2. 1471, 6171 haszują do tej samej wartości
3. Wszystkie elementy otrzymują tę samą wartość
4. Każdy element otrzymuje inną wartość

Zadanie 2

Elementy o następujących kluczach: 12, 18, 13, 2, 3, 23, 5, 15 wstawiono do początkowo pustej tablicy z haszowaniem korzystającej z otwartego adresowania liniowego i funkcji haszującej $h(k) = k \bmod 10$. Jak wygląda stan tablicy?

Indeks	Wartość klucza
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Zadanie 3

Tablica haszująca o rozmiarze 10 używa adresowania otwartego oraz funkcji haszującej $h(k) = k \bmod 10$. Stan tablicy po 6 wstawieniach przedstawiono poniżej. Jaka mogła być kolejność wstawiania kluczy?

Indeks	Wartość klucza
0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

1. 46, 42, 34, 52, 23, 33
2. 34, 42, 23, 52, 33, 46
3. 46, 34, 42, 23, 52, 33
4. 42, 46, 33, 23, 34, 52

Zadanie 4

Dana jest tablica o rozmiarze $m = 11$ oraz funkcje haszujące h_1 i h_2 :

- $h_1(\text{key}) = \text{key} \bmod m$
- $h_2(\text{key}) = (\text{key} \bmod (m - 1)) + 1$

Wstaw elementy o następujących kluczach: {22, 1, 13, 11, 24, 33, 18, 42, 31} do tablicy z haszowaniem, która korzysta z następujących metod:

1. Metoda łańcuchowa z funkcją haszującą h_1 : $h(k) = h_1(k)$
2. Liniowe adresowanie otwarte z h_1 : $h(k, i) = (h_1(k) + i) \bmod m$
3. Haszowanie dwukrotne z h_1 i h_2 : $h(k, i) = (h_1(k) + i h_2(k)) \bmod m$

	Łańcuchowo	Adresowanie liniowe	Podwójne haszowanie
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Zadanie 5

Co by się stało, gdybyśmy w poprzednim zadaniu zamienili funkcje h_1 i h_2 ?

Zadanie 6

Jakie ma znaczenie to, czy rezultat funkcji h_2 i rozmiar tablicy mają wspólne dzielniki?

Zadanie 7

Jak usunąć element z tablicy z haszowaniem korzystającej z adresowania otwartego?

W domu: Zapisz w pseudokodzie operację HASH-DELETE(T, k) i popraw procedury HASH-SEARCH i HASH-INSERT tak, aby uwzględniały zachowanie procedury HASH-DELETE.

Zadanie 8

Rozwiąż zadanie <https://www.hackerrank.com/challenges/ctci-ransom-note/> na Hacker-ranku.

Zadanie 9

Rozwiąż zadanie <https://www.hackerrank.com/challenges/ctci-ice-cream-parlor/> na Hacker-ranku.

Zadania, których nie udało się wykonać w czasie ćwiczeń należy rozwiązać w domu.

Bibliografia

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Wprowadzenie do algorytmów*, PWN, 2012 (Wydanie III oryginału, I wydanie PWN, rysunki pochodzą z oryginału w języku angielskim).