

# Tabele de dispersie

SD 2017/2018

Tabele cu adresare directă

Tabele de dispersie

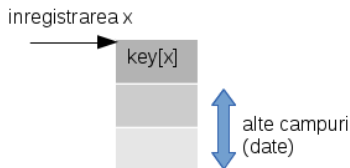
Dispersie externă

Funcții de dispersie

Dispersie internă

# Tabele de simboluri

- ▶ Tabela de simboluri  $S$  cu  $n$  înregistrări;
- ▶ Fiecare înregistrare are asociată o cheie (unică);
- ▶ Operații:  $cauta(S, k)$ ,  $insereaza(S, x)$ ,  $sterge(S, x)$ ;
- ▶ Cum poate fi organizată structura de date  $S$ ?



# Tabela cu adresare directă

- ▶  $U = \{0, 1, \dots, m - 1\}$  mulțimea univers a cheilor;
- ▶ Un tablou  $T[0..m - 1]$ :

$$T[k] = \begin{cases} x & \text{daca } x \in S \text{ și } x.\text{cheie} = k \\ NULL & \text{altfel.} \end{cases}$$

- ▶ Fiecare poziție (slot) din tablou corespunde unei chei din universul  $U$ .
- ▶ Dacă  $|S| = n$ , atunci  $n \leq m$ .

# Tabela cu adresare directă - Operații

- ▶ Operații

**Function** *cauta*( $T, k$ )

**begin**

    return  $T[k]$

**end**

**Procedure** *insereaza*( $T, x$ )

**begin**

$T[x.cheie] = x$

**end**

**Procedure** *sterge*( $T, x$ )

**begin**

$T[x.cheie] = NULL$

**end**

- ▶ Complexitatea timp a operațiilor:  $\Theta(1)$

# Tabela cu adresare directă

- ▶ Spațiul de memorare:  $\Theta(|U|)$ .
- ▶ **Probleme:**
  - ▶ cheile pot să nu fie numere întregi;
  - ▶ domeniul de valori al cheilor este foarte mare:
    - ▶ numere pe 64 de biți (18.446.744.073.709.551.616 chei diferite)
    - ▶ șiruri de caractere;
  - ▶ mulțimea de chei memorate este foarte mică relativ la  $U$ .
- ▶ **Soluție:** tabela de dispersie
  - ▶ o generalizare a noțiunii de tabelă cu adresare directă;
  - ▶ o structură de date eficientă pentru implementarea dicționarelor.

Tabele cu adresare directă

Tabele de dispersie

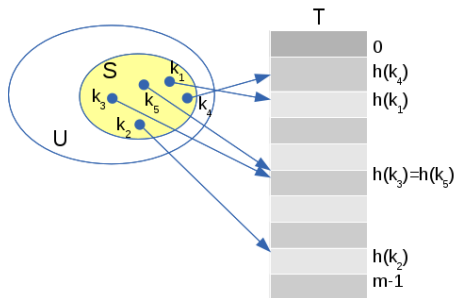
Dispersie externă

Funcții de dispersie

Dispersie internă

# Tabela de dispersie

- ▶ Utilizează o **funcție de dispersie** (*hash*)  $h$  pentru a asocia cheilor din universul  $U$  o valoare din mulțimea  $\{0, 1, \dots, m-1\}$ .



- ▶ Un element cu cheia  $k$  are asociată poziția  $h(k)$  în tabela  $T$ .
- ▶ Funcția de dispersie reduce domeniul de valori a indicilor și implicit dimensiunea vectorului memorat.
- ▶ **Coliziune:**  $\exists x_1, x_2 \in S$  astfel încât  $h(x_1.cheie) = h(x_2.cheie)$



Tabele cu adresare directă

Tabele de dispersie

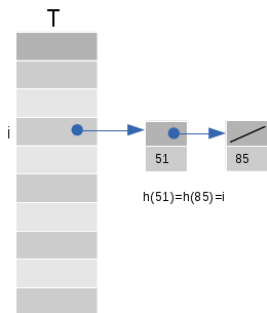
**Dispersie externă**

Funcții de dispersie

Dispersie internă

# Rezolvarea coliziunilor prin înlănțuire (dispersie externă)

- ▶ Înregistrările care au asociate același slot vor fi memorate într-o listă liniară.  $T$  devine tablou de pointeri.



- ▶ Soluție simplă, dar necesită spațiu suplimentar de memorie.
- ▶ Cazul cel mai nefavorabil: toate cheile au asociate același slot
  - ▶ timpul de acces:  $\Theta(n)$ .

# Dispersie externă – Operații

**Function** *cauta*( $T, k$ )

**begin**

caută elementul cu cheia  $k$  în lista  $T[h(k)]$

**end**

**Procedure** *insereaza*( $T, x$ )

**begin**

inserează  $x$  la începutul listei  $T[h(x.cheie)]$

**end**

**Procedure** *sterge*( $T, x$ )

**begin**

sterge  $x$  din lista  $T[h(x.cheie)]$

**end**

# Dispersie externă – analiza complexității

- ▶ *Căutare:*

Complexitatea în cazul cel mai nefavorabil depinde de lungimea listei.

- ▶ *Inserare:*

Complexitatea în cazul cel mai nefavorabil:  $O(1)$ .

- ▶ *Ștergere:*

$O(1)$  dacă avem liste liniare dublu înlănțuite; dacă lucrăm cu liste liniare simplu înlănțuite, trebuie întâi să căutăm  $x$  și să reținem predecesorul acestuia pentru a putea reface legatura.

# Dispersie externă – analiza complexității în cazul mediu

- ▶ **Ipoteza dispersiei uniforme simple:** fiecare cheie  $k \in U$  are o probabilitate egală de a fi memorată în oricare locație din tabela  $T$  și independent de locațiile altor chei.
- ▶ **Factorul de încărcare** al tabeli  $T$  este

$$\alpha = n/m,$$

unde  $n$  este numărul de chei ( $|S|$ ), iar  $m$  numărul de locații (dimensiunea tabloului  $T$ ).

- ▶ Timpul de calcul al funcției de dispersie este  $\Theta(1)$ .

# Dispersie externă – analiza complexității în cazul mediu

## Teoremă:

*Considerând o tabelă de dispersie în care coliziunile sunt rezolvate prin înlănțuire, în ipoteza dispersiei uniforme simple, o căutare **fără succes** are complexitatea timp în **cazul mediu**  $\Theta(1 + \alpha)$ .*

## Teoremă:

*Într-o tabelă de dispersie în care coliziunile sunt rezolvate prin înlănțuire, în ipoteza dispersiei uniforme simple, o căutare **cu succes** are complexitatea timp în **cazul mediu**  $\Theta(1 + \alpha)$ .*

## Corolar:

*Dacă numărul de sloturi este cel puțin proporțional cu numărul de elemente ( $n = O(m)$  sau, echivalent,  $\alpha = O(1)$ ), atunci operația de căutare are complexitatea, în **medie**,  $O(1)$ .*

Tabele cu adresare directă

Tabele de dispersie

Dispersie externă

**Funcții de dispersie**

Dispersie internă

# Funcția de dispersie

- ▶ *Deterministă*: pentru o cheie  $k$ , funcția trebuie să furnizeze întotdeauna aceeași valoare  $h(k)$ .
- ▶ *Aleatoare*: vizează minimizarea coliziunilor.
- ▶ O funcție hash bună distribuie cheile uniform în locațiile tablei.
- ▶ Ipoteza dispersiei uniforme simple este dificil de garantat, dar există tehnici euristice care funcționează bine în practică (atât timp cât deficiențele acestora pot fi evitate).



# Funcții de dispersie – Metoda diviziunii

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

- ▶ Presupunem că toate cheile sunt numere naturale.
  - ▶ dacă cheile nu sunt numere naturale, atunci trebuie găsită o modalitate de a le interpreta ca numere naturale;
  - ▶ *Exemplu:* presupunem un identificator de forma (112, 116); în baza 128, acesta devine  $(112 \times 128) + 116 = 14452$ .
- ▶ Nu se alege pentru  $m$  o valoare care are un divizor mic  $d$ . Preponderența cheilor congruente modulo  $d$  poate afecta în mod negativ uniformitatea.
- ▶ Dacă  $m = 2^r$ , atunci valoarea funcției depinde doar de ultimii  $r$  biți ai lui  $k$ .
  - ▶ *Exemplu:*  $k = 1011000111011010$  și  $r = 6 \mapsto h(k) = 011010$ .
- ▶ Se alege  $m$  un număr prim care nu este apropiat de o putere a lui 2 sau 10.

# Funcții de dispersie – Metoda înmulțirii

$$h(k) = \lfloor m(kA - \lfloor kA \rfloor) \rfloor$$

- ▶  $A \in (0, 1)$  este o constantă.
- ▶ Valoarea lui  $m$  nu este critică (de obicei o putere a lui 2).

$$h(k) = (kA \bmod 2^w) rsh(w - r)$$

- ▶  $m = 2^r$ , (mașină în care cuvintele sunt pe  $w$ -biți).
- ▶  $A$  este un număr impar din intervalul  $(2^{w-1}, 2^w)$ .
- ▶  $rsh$  este operatorul de deplasare la dreapta pe biți.

# Funcții de dispersie – Metoda înmulțirii

- ▶ *Exemplu:*  $m = 2^3$  și cuvinte pe  $w = 7$  biți.

$$\begin{array}{r} \phantom{x} \phantom{0000000} 1011001 = A \\ x \phantom{0000000} 1101011 = k \\ \hline 1001010\textcolor{red}{011}0011 \\ \phantom{0000000} \textcolor{blue}{\longleftrightarrow} \phantom{0000000} \\ \phantom{0000000} h(k) \end{array}$$

- ▶ Nu se alege  $A$  prea aproape de  $2^{w-1}$  sau  $2^w$ .
- ▶ Knuth:  $A = (\sqrt{5} - 1)/2$ .
- ▶ Înmulțirea modulo  $2^w$  este mai rapidă în comparație cu împărțirea; operatorul *rsh* este rapid.

# Funcții de dispersie – Dispersia universală

$$h(k) = [(ak + b) \bmod p] \bmod m$$

- ▶  $p$  număr prim cu  $p > |U|$ ;
- ▶  $a, b$  numere aleatoare din  $\{0, \dots, p-1\}$ .
- ▶  $k_1 \neq k_2, Pr_{a,b}\{h(k_1) = h(k_2)\} = 1/m$ .

Tabele cu adresare directă

Tabele de dispersie

Dispersie externă

Funcții de dispersie

Dispersie internă

# Rezolvarea coliziunilor prin adresare deschisă

- ▶ *Dispersie internă*
- ▶ Toate elementele sunt memorate în interiorul tabeli  $T$ ; nu este utilizat spațiu suplimentar de memorie, în afara tabeli de dispersie.
- ▶ Funcția de inserare examinează tabela până când este găsită o locație liberă.
- ▶ Funcția de dispersie depinde atât de cheie cât și de numărul examinării:

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

- ▶ Secvența de examinări  $\langle h(k, 0), h(k, 1), \dots, h(k, m - 1) \rangle$  trebuie să fie o permutare a  $\{0, 1, \dots, m - 1\}$ .
- ▶ Dezavantaje: tabela se poate umple; ștergerea poate deveni dificilă.

```
Function cauta( $T, k$ )  
begin  
     $i \leftarrow 0$   
    repeat  
         $j \leftarrow h(k, i)$   
        if  $T[j] == k$  then  
            return  $j$   
        else  
             $i \leftarrow i + 1$   
    until  $T[j] == \text{NULL}$  OR  $i == m$ ;  
    return NULL  
end
```

```
Function insereaza( $T, k$ )  
begin  
     $i \leftarrow 0$   
    repeat  
         $j \leftarrow h(k, i)$   
        if  $T[j] == \text{NULL}$  then  
             $T[j] \leftarrow k$   
            return  $j$   
        else  
             $i \leftarrow i + 1$   
    until  $i == m$ ;  
    return  $-1$   
end
```



## Examinare liniară:

$$h(k, i) = (h'(k) + i) \bmod m$$

- ▶  $h'(k)$  o funcție de dispersie uzuală.
- ▶ Pentru o cheie  $k$ , secvența de examinare este

$$h'(k), h'(k) + 1, h'(k) + 2, \dots, m - 1, 0, 1, \dots, h'(k) - 1.$$

- ▶ Avantaj: metodă simplă.
- ▶ Dezavantaj: grupare primară (*primary clustering*) – se formează șiruri lungi de locații ocupate; crește timpul mediu de căutare.

## Examinare pătratică:

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

- ▶  $h'(k)$  o funcție de dispersie uzuală.
- ▶ Pentru o cheie  $k$ , prima locație examinată este  $h'(k)$ , iar următoarele poziții examinate sunt decalate cu cantități ce depind într-o manieră pătratică de poziția anterior examinată.
- ▶ Dezavantaj: grupare secundară – dacă două chei au aceeași poziție de start a examinării, atunci secvențele de verificare coincid.
- ▶ Funcționează mai bine decât verificarea liniară.

## Dispersie dublă:

$$h(k, i) = (h_1(k) + ih_2(k)) \bmod m$$

- ▶  $h_1(k)$  și  $h_2(k)$  două funcții de dispersie uzuale.
- ▶ Pentru o cheie  $k$ , prima locație examinată este  $h_1(k)$ , iar următoarele poziții examinate sunt decalate față de poziția anterioară cu  $h_2(k) \bmod m$ .
- ▶ Această metodă produce în general rezultate foarte bune, cu condiția ca  $h_2(k)$  să fie relativ prim cu  $m$ . O modalitate de a realiza acest lucru este să considerăm  $m$  o putere a lui 2 și să alegem  $h_2(k)$  astfel încât să rezulte doar numere impare.

# Dispersie internă – Analiza complexității

**Ipoteza dispersiei uniforme:** fiecare cheie are aceeași probabilitate de a avea oricare din cele  $m!$  permutări ca secvență de examinări.

## Teoremă:

*Într-o tabelă de dispersie cu adresare deschisă, în ipoteza dispersiei uniforme, cu factor de încărcare  $\alpha < 1$ , numărul mediu de verificări este cel mult*

- ▶  $\frac{1}{1-\alpha}$  pentru operația de căutare fără succes, și
- ▶  $\frac{1}{\alpha} \ln \frac{1}{1-\alpha}$  pentru operația de căutare cu succes.

## Corolar:

*Dacă  $\alpha$  este constant, atunci accesarea unei tabele de dispersie cu adresare deschisă necesită în medie un timp constant,  $\Theta(1)$ .*

- ▶ Tabelele de dispersie sunt folosite la: indexarea în baze de date, compilatoare - tabela de simboluri, *cache*, etc.
- ▶ Aplicații ale funcțiilor de dispersie: *CRC*, *Cryptographic hash functions*, etc.