TABELA DE DISPERSIE

- continuare -

C. Rezolvare coliziuni prin *adresare deschisă* – OPEN ADDRESSING

- Toate elementele sunt memorate în interiorul tabelei, nu există liste memorate înafara tabelei.
- Fiecare intrare în tabelă conține fie un element al containerului, fie un marcaj pentru locație liberă (ex. NIL).
- Nu se folosesc pointeri pentru înlănțuiri.
- Factorul de încărcare este subunitar $\alpha < 1$, altfel tabela este plină
- Dezavantaj: tabela se poate umple ($\alpha = 1$). Soluție: se crește *m*, ceea ce presupune redispersarea elementelor.
- Avantaj: spațiul de memorie suplimentar (nu se memorează pointeri) oferă tabelei un număr mai mare de locații pentru același spațiu de memorie, putând rezulta coliziuni mai puține și acces rapid.
- Secvența de locații care se examinează nu se determină folosind ***pointeri,*** ci se ***calculează***
- La **adăugare** în tabelă, se examinează succesiv locațiile, până se găsește o locație liberă în care să punem cheia (elementul). În loc să fie fixată ordinea de verificare a tabelei (ex: 0,1,2...,m-1) care ar necesita timp $\theta(m)$**,** secvența de poziții examinate depinde de cheia (elementul) care se inserează.

- Se extinde funcția de dispersie $d : U \times \{0,...m-1\} \rightarrow \{0,..,m-1\}$ - al doilea argument al funcției se numește ***număr de verificare***
- Pentru o cheie $c \in U$ secvența $< d(c,0), d(c,1),..., d(c,m-1) >$ se numește ***secvența de verificare*** a cheii *c*
- **<u>Cerință</u>**
    - secvența de verificare a oricărei chei $c \in U$ $< d(c,0), d(c,1),..., d(c,m-1) >$ să fie o permutare a $< 0,1,...,m-1 >$

<u>Ipoteza dispersiei uniforme simple</u> (SUH)

- Pentru orice cheie $c \in U$, permutarea $< d(c,0), d(c,1),..., d(c,m-1) >$ poate să apară sub forma oricărei permutări a $< 0,1,...,m-1 >$

C.1. Verificare liniară – LINEAR PROBING

$$d(c,i) = (d'(c) + i) \bmod m \qquad \forall i = 0,1,..., m-1$$

$d':U \rightarrow \{0,..,m-1\}$ este o funcție de dispersie uzuală (ex: $d'(c) = c \bmod m$)

- Fiind dată o cheie ***c*** secvența ei de verificare este $< d'(c), d'(c)+1, d'(c)+2, ..., m-1, 0, 1, ... d'(c)-1 >$
- Problema: **grupare primară** – se formează șiruri lungi de locații ocupate, crescând timpul mediu de căutare

C.2. Verificare pătratică – QUADRATIC PROBING

$$d(c,i) = (d'(c) + c_1 \cdot i + c_2 \cdot i^2) \bmod m \qquad \forall i = 0,1,...,m-1$$

$d':U \rightarrow \{0,..,m-1\}$ este o funcție de dispersie uzuală (ex: $d'(c) = c \bmod m$), $c_1 \neq 0$ și $c_2 \neq 0$ sunt constante auxiliare fixate la inițializarea funcției de dispersie.

- Constantele $c_1 \neq 0$ și $c_2 \neq 0$ se pot determina euristic
    - Pentru a asigura faptul că secvența de verificare este o permutare $\{0,..,m-1\}$
        - *m* se alege putere a lui 2 și $c_1 = c_2 = 0.5$
        - *m* se alege număr prim de forma $4 \cdot k + 3$, $d'(c) = c \bmod m$ și $c_1 = 0$, $c_2 = (-1)^i$
- Fiind dată o cheie ***c*,** prima poziție examinată este $d'(c)$, după care următoarele poziții examinate sunt decalate cu cantități ce depind într-o manieră pătratică de locația anterior examinată.
- Problema: **grupare secundară** – dacă 2 chei au aceeași poziție de start a verificării, atunci secvența lor de verificare coincide (dacă $d(c',0) = d(c'',0) \Rightarrow d(c',i) = d(c'',i) \quad \forall i = 0,1,...,m-1$)
- Experimental: funcționează **mai bine** decât **verificarea liniară**

C.3. Dispersia dublă – DOUBLE HASHING

$$d(c,i) = (d_1(c) + i \cdot d_2(c)) \bmod m \qquad \forall i = 0,1,...,m-1$$

d_1 și d_2 sunt funcții de dispersie aleatoare.

- Este considerată una dintre cele mai bune metode disponibile pentru adresarea deschisă
- Fiind dată o cheie ***c*,** prima poziție examinată este $d_1(c)$, după care următoarele poziții examinate sunt decalate față de poziția anterioară cu $d_2(c) \bmod m$.
- $d_2(c)$ și *m* să fie prime între ele pentru a fi parcursă întreaga tabelă
- **Exemplu**
    - *m* prim

- $d_1(c) = c \bmod m \qquad d_2(c) = (1 + c \bmod m')$
- m' se alege de obicei m-1 sau m-2

- Performanța dispersiei duble apare ca fiind foarte apropiată de performanța schemei ideale a dispersiei uniforme ($\theta(m^2)$ secvențe de verificare posibile pentru o cheie)

EXEMPLE

1. Adresare deschisă cu verificare liniară, m=10, $d'(c)=c \bmod m$

c	5	15	13	22	20	35	30	32	2
d'(c)	5	5	3	2	0	5	0	2	2

2. Adresare deschisă cu dispersie dublă, m=13, $d_1(c)=c \bmod m$, $d_2(c)=1 + c \bmod (m-2)$

c	79	69	96	14
d_1(c)	1	4	5	1
d_2(c)	3	4	9	4

<u>Implementarea operațiilor</u>

Presupuneri și notații:

- Pp. că în container memorăm doar chei
- O locație din tabelă va conține:
  - NIL (constantă simbolică) – dacă locația e liberă (nu conține o cheie)
  - O cheie din container
- Reprezentarea containerului folosind o TD cu adresare deschisă

Container

m: Intreg {nr.de locații din tabelă}

ch: TCheie[0..m-1] {cheile din container}

d: TFunctie {funcția de dispersie asociată}

Subalgoritmul ADAUGĂ (*c*, *ch*) este
{*c*:Container, *ch*:TCheie}
i←0 {numărul de verificare}
gasit←fals {nu am găsit poziția de adăugare}
repetă
j←*c*.d(*ch*,*i*) {locația de verificat}

        dacă *c*.ch[*j*]=NIL atunci
            c.ch[*j*]←*ch* {memorez cheia}
            gasit←adev {am găsit poziția unde putem adăuga}
        altfel
            i←*i*+1 {căutăm mai departe pe secvența de verificare}
        sfdacă
    până_când (*i*=*c*.m) sau (*gasit*)
    dacă *i*=*c*.m atunci {tabela e plină}
        @ depășire tabelă
    Sfdacă
sfADAUGĂ

Funcția CAUTĂ (*c*, *ch*) este
{*c*:Container, *ch*:TCheie}
    i←0 {numărul de verificare}
    gasit←fals {nu am găsit cheia}
    repetă
        j←*c*.d(*ch*,*i*) {locația de verificat}
        dacă *c*.ch[*j*]=*ch* atunci {am găsit cheia}
            gasit←adev
        altfel
            i←*i*+1 {căutăm mai departe pe secvența de verificare}
        sfdacă
    până_când (*c*.ch[*j*]=NIL) sau (*i*=*c*.m) sau (*gasit*)
    CAUTĂ← *gasit*
sfCAUTĂ

Analiza dispersiei cu adresare deschisă

1. Teorema. **Într-o TD cu adresare deschisă, în *ipoteza dispersiei uniforme simple* (SUH), cu factor de încărcare** $\alpha = \frac{n}{m} < 1$ **numărul *mediu* de verificări este CEL MULT**

   CORMEN, THOMAS H. - LEISERSON, CHARLES - RIVEST, RONALD R.: Introducere în algoritmi. Cluj-Napoca: Editura Computer Libris Agora, 2000.

   - $\frac{1}{1-\alpha}$ pentru **adăugare** și **căutare fără succes**
   - $\frac{1}{\alpha} \cdot \ln \frac{1}{1-\alpha}$ pentru **căutare cu succes**

CONCLUZII

- Dacă α e constant $\Rightarrow$ $\theta(1)$ în **medie** pentru operații
- Caz defavorabil $O(n)$

PROBLEME

1. Considerând o tabelă de dispersie cu adresare deschisă, scrieți un algoritm pentru operația de **ștergere** și modificați operația **adaugă** și **caută** pentru a încorpora valoarea specială **ȘTERS.**
2. Se consideră o tabelă de dispersie cu adresare deschisă, cu dispersie uniformă și factor de încărcare 0.5. Dați margini superioare pentru numărul mediu de verificări într-o căutare cu succes și o căutare fără succes.