Seminar 4 - Multidictionar Ordonat implementare cu LSI

Dictionar

- container cu perechi (cheie, valoare)
- cheile sunt distincte (o cheie are asociata o singura valoare)

Multidictionar

- container cu perechi (cheie, valoare)
- pentru o cheie putem sa avem mai multe valori

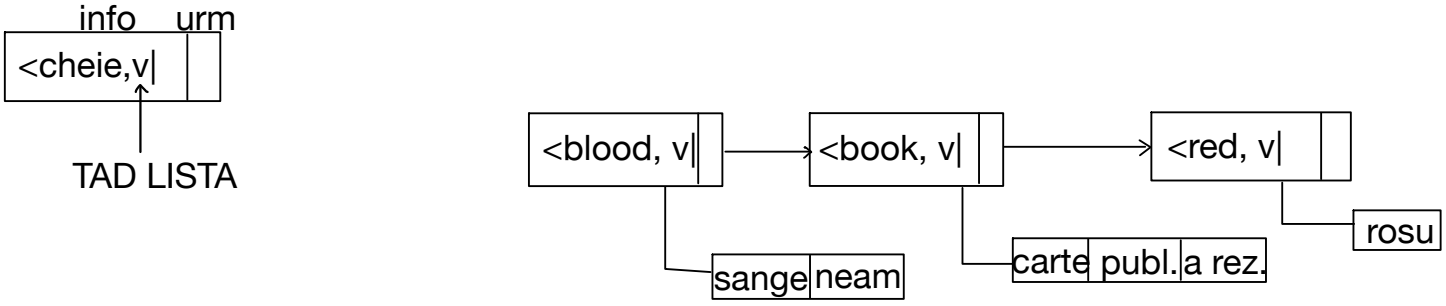
Multidictionar Ordonat

- container cu perechi (cheie, valoare)
- pentru o cheie putem sa avem mai multe valori
- cheile sunt ordonate dupa o relatie data

Continut MDO

- blood : sange, neam
- book : a rezerva, carte, publicatie
- read : rosu

Reprezentare MDO



MDO

prim : *Nod
r : relatie

Nod

e : TElement
urm : *Nod

TElement

c : TCheie
v : TLista

IteratorMDO

curent : *Nod
mdo : MDO
itL: IteratorLista

IMPLEMENTARE MDO

Operatii :

- creeaza(mdo, R)
- distruge(mdo)
- cauta(mdo, c, lista)
- adauga(mdo, c, v)
- sterge(mdo, c, v)

```
creeaza(mdo, R) :
    mdo.prim = NULL
    mdo.R =  R
sf subalg

distruge(mdo) :
    cat timp mdo.prim != NULL
        deSters = mdo prim
        mdo prim = [mdo.prim].urm
        distruge([deSters].e.v)
        dealoca(deSters)
sf subalg

cautaNod(mdo,c,nodC,prec) :
    crt = mdo.prim
    prec = NULL
    gasit = fals
    cat timp crt!=NULL && !gasit && (R[crt]e.c,c) executa
        daca c == [crt].e.c atunci
            gasit = adevarat
        altfel
            precd = crt
            crt = [crt].urm
        sf daca
    sf cat timp
    daca gasit atunci
        nodC = crt
        prec = precd
    altfel
        nodC = NULL
        prec = precd
    sf daca
sf subalg

cauta(mdo, c, lista) :
    cauta(mdo, c, nodC, prec)
    daca nodC == NULL atunci
        creeaza(lista)
    altfel
        lista = nodC.e.v
    sf daca
sf subalg

adauga(mdo, c, v)
    cautaNod(mdo, c, nodC, prec)
    daca nodC != NULL atunci
        adaugaSfarsit([nodC].e.v,v)
    altfel
        adaugaCheieNoua(mdo,c,v,prec)
    sf daca
sf subalg
```

IMPLEMENTARE MDO ITERATOR

Operatii :

- creeaza(it, mdo)
- element(it)
- valid(it)
- urmator(it)

```
creeaza(it, mdo)
    it.mdo = mdo
    curent = mdo.prim
    daca curent != NULL atunci
        iterator(it,itL,[curent].e.v)
    sf daca
sf subalg

element(it,e)
    c = [crt].e.c
        element(it.itL,v)
    e = <c,v>
sf subalg

urmator(it)
    urmator(it, itL)
    daca valid(it, itL) atunci
        curent = [curent].urm
        daca curent != NULL
            iterator(it,itL,[curent].e.v
        sf daca
    sf daca
sf subalg
```