Multime

Ce este o multime (set) ?
O multime (set) este o structura de date utilizata pentru a reprezenta o colectie finita de elemente distincte. Elementele dintr-o multime nu se repeta si ordinea lor nu conteaza.
Exemplu : m = {1,2,3,5,4}

Proprietati importante ale multimilor :
- Unicitatea : elementele nu se repeta
- Ordinea nu conteaza : elementele nu au o ordine specifica

Reprezentarea interna :
Multimile pot fi reprezentate sub forma unui vector pentru a permite accesul prin indexare, desi ordinea nu este relevanta logic.

Submultimi :
O submultime este reprezentata printr-un vector caracteristic, care indica prezenta (1) sau absenta (0) a elementelor din multimea principala.
Exemplu :

```
Mulțimea M = {110, 200, 318, 400}
Submulțimea S = {200, 400} se reprezintă prin vectorul (0, 1, 0, 1)
```

Operatii asupra multimilor si submultimilor :
- Reuniunea
- Intersectia

Operatii specifice multimilor (interfata - TAD Multime) :
- creeaza(m) - creeaza o multime vida
- adauga(m, e) - adauga un element in multime (doar daca nu exista deja)
- sterge(m, e) - elimina un element din multime
- cauta(m, e) - verifica existenta unui element
- dim(m) - verifica daca multimea este vida
- iterator(m, i) - permite accesul secvential la elemente
- distruge(m) - elibereaza spatiul ocupat in memorie

Implementari posibile ale multimilor :
Multimile pot fi implementate folosind :
- Tablouri dinamice;
- Vectori booleeni (de biti);
- Liste inlantuite;
- Tabele de dispersie;
- Arbori binari de cautare echilibrati;

Matrice

Ce este o matrice?
O matrice este un tablou bidimensional static, organizat in linii si coloane. Se foloseste pentru a stoca date sub forma de tabel.

Operatii de baza pentru matrice (interfata TAD) :
- creeaza(m, nrLin, nrCol) - creeaza o matrice nula cu un numar specificat de linii si coloane
- nrLinii(m) - returneaza numarul de linii
- nrColoane(m) - returneaza numarul de coloane
- element(m, i, j, e) - acceseaza elementul de pe linia i, coloana j
- modifica(m, i, j, e) - modifica sau adauga elementul pe pozitia i, coloana j

Observatii generale despre matrice :

- De obicei, tablourile bidimensionale se memoreaza secvential (in memorie liniara)
- Pot exista tablouri bidimensionale dinamice (cu dimensiuni ajustabile)
- Daca mai multe elemente din matrice sunt nule (matrici rare), memorarea tuturor elementelor este ineficienta, in acest caz, se folosesc reprezentari speciale pentru a stoca doar elementele nenule

Exemple de reprezentare a unei matrice rare :

Se considera matricea :

$$\begin{pmatrix} 0 & -2 & 0 & -7 & 0 \\ -6 & 0 & 0 & 0 & 0 \\ 0 & -9 & -8 & 0 & -5 \\ 0 & 0 & 0 & 0 & -2 \end{pmatrix}$$

Aceasta matrice poate sa fie reprezentata astfel :

A. Reprezentare prin triplete (linie, coloana, valoare) :

Tripletele sunt ordonate dupa linie si coloana.

Linie	1	1	2	3	3	3	4
Coloana	2	4	1	2	3	5	5
Valoare	-2	-7	-6	-9	-8	-5	-2

Aceasta reprezentare poate fi implementata prin :

- Vector dinamic ordonat
- Lista inlantuita ordonata
- Arbore binar de cautare

B. Reprezentare condensata pe coloane :

Se folosesc trei vectori :

- Coloana : [1,2,4,5,6,8]
- Linie : [2,1,3,3,1,3,4]
- Valoare : [-6,-2,-9,-8,-7,-5,-2]

Interpretare este :

Coloana (j)	Început (`Coloana[j]`)	Sfârșit (`Coloana[j+1]-1`)	Elemente nenule din coloana j
1	1	1	(linie=2, valoare=-6)
2	2	3	(linie=1,-2), (linie=3,-9)
3	4	4	(linie=3, valoare=-8)
4	5	5	(linie=1, valoare=-7)
5	6	7	(linie=3,-5), (linie=4,-2)

Coloana[j] : indica pozitia in vectorii Linie si Valoare la care incepe prima valoarea nenula a coloanei j
Coloana[j + 1] - 1 : indica pozitia in vectorii Linie si Valoare la care se termina valorile nenule a coloanei j

Astfel:
Coloana[1] = 1 si Coloana[2]-1 = 1, deci coloana 1 contine doar un element nenul care se afla pe pozitia 1 din vectorul de valori : 6 si se afla pe linia corespunzatoare valorii de pe pozitia 1 din vectorul linie : 2

Coloana[2] = 2 si Coloana[3]-1 = 3, deci coloana 2 contine doua elemente nenule care se afla pe pozitiile 2 si 3 din vectorul de valori : -2 si -9 care se afla pe liniile corespunzatoare valorii de pe pozitia 2 respectiv pe pozitia 3 din vectorul linie : 1, 3

C. Reprezentare condensata pe linii :
Similara cu reprezentarea condensata pe coloane, dar rolul liniilor si coloanelor sunt inversate.

D. Reprezentare inlantuita prin liste circulare :
Fiecare element nenul este stocat intr-un nod, fiecare nod contine un triplet de forma :
Nod(val=x, linie=y, coloana=z)

Dictionar

Ce este un dictionar (Map)?
Un dictionar este o structura care stocheaza elementele sub forma unor perechi (cheie, valoare). Cheia este utilizata pentru identificarea rapida a valorii asociate.

Proprietatile dictionarului :
- Cheile sunt unice intr-un dictionar.
- Fiecare cheie are asociata, de regula, o singura valoare.
- Daca este nevoie de mai multe valori pentru aceeasi cheie, vom folosi un multi-dictionar (MultiMap).

Operatii principale ale dictionarului (interfata TAD) :
- creeaza(d) - creeaza un dictionar vid
- adauga(d, c, v) - adauga perechea (c, v) in dictionar (daca exista deja cheia atunci valoarea se inlocuieste)
- cauta(d, c, v) - cauta cheia si returneaza valoarea asociata (daca exista cheia)
- sterge(d, c, v) - sterge perechea identificata prin cheia c si returneaza valoarea asociata care tocmai a fost stearsa
- dim(d) - returneaza numarul de perechi (cheie, valoare) din dictionar
- vid(d) - verifica daca dictionarul este gol
- chei(d, m) - returneaza multimea tuturor cheilor din dictionar
- valori(d, c) - returneaza multimea tuturor valorilor din dictionar
- perechi(d, m) - returneaza multimea tuturor perechilor (cheie, valoare) din dictionar
- iterator(d, i) - permite parcurgerea elementelor dictionarului printr-un iterator
- distruge(d) - elibereaza memoria alocata dictionarului

Variante speciale ale dictionarului :

1. Multi-dictionar (MultiMap)
- Permite asocierea mai multor valori unei singure chei.
- Operatia de stergere va sterge doar perechea exacta cheie-valoare.

2. Dictionar ordonat (SortedMap)
- Cheile sunt ordonate (alfabetic / numeric).
- Operatiile care returneaza elemente (iterator, perechi) ofera elementele in ordine.

3. Multi-dictionar ordonat (Sorted MultiMap)
- Combina MultiMap cu SortedMap -> permite asocierea mai multor valori unei singure chei + ordine intre chei.

Modalitati de implementare ale dictionarelor :
- Tablouri dinamice
- Liste inlantuite
- Tabele de dispersie
- Arbori binari de cautare

Colectie

Ce este o colectie?
Este o structura finita de elemente, spre deosebire de multime, elementele dintr-o colectie pot aparea de mai multe ori (nu sunt unice).
Ordinea elementelor nu conteaza.

Se mai numeste si **multi-set** sau **bag** datorita faptului ca permite aparitii multiple ale aceluiasi element.

Exemplu de colectie :
c = {1, 2, 3, 1, 3, 2, 4, 2, 2}

Operatii principale (interfata TAD) :
- creeaza(c) - creeaza o colectie vida
- adauga(c, e) - adauga un element nou in colectie (permite duplicarea elementelor)
- sterge(c, e) - sterge o singura aparitie a unui element (daca exista)
- cauta(c, e) - verifica daca un element exista in colectie
- dim(c) - returneaza numarul total al elementelor din colectie (cu tot cu duplicate)
- vida(c) - verifica daca colectia este goala
- iterator(c, i) - permite parcurgerea tuturor elementelor din colectie
- distruge(c) - elibereaza memoria ocupata de colectie

Reprezentari ale Colectiei :
1. **Reprezentare simpla** (toate elementele, chiar si duplicatele)
{1, 2, 1, 4, 3, 1, 4, 2, 5}
2. **Reprezentarea frecventei** (elementul si cate aparitii are)
(1,3), (2,2), (4,2), (3,1), (5,1)

Modalitati de implementare :
- Tablouri dinamice
- Liste inlantuite
- Tabele de dispersie
- Arbori binari echilibrati

Coada

Ce este o coada?
O coada este o structura liniara in care elementele sunt introduse si sterse dupa principiul FIFO (First in, First Out), adica :
- Primul element introdus este primul care va fi eliminat.
- Elementele noi se adauga intotdeauna la sfarsitul cozii.
- Elementele existente sunt eliminate intotdeauna din fata cozii.

Exemplu intuitiv :
Coada de oameni la magazin, primul venit e primul servit.

Operatii principale (interfata TAD Coada) :
- creeaza(c) - Creeaza o coada vida
- adauga(c, e) - Adauga un element la sfarsitul cozii (Exceptie daca coada este plina)
- sterge(c, e) - Sterge primul element introdus in coada si-l returneaza (Exceptie daca coada este vida)
- element(c, e) - Returneaza primul element introdus, fara sa-l elimine din coada (Exceptie daca coada este vida)

- vida(c) - Verifica daca coada este goala
- plina(c) - Verifica daca coada a atins capacitatea maxima (daca este definita o capacitate)
- distruge(c) - Eliberează memoria ocupata de coada

Observatii :
- Coada nu permite accesul direct la elementele intermediare
- Pentru tiparirea unei cozi trebuie folosita o coada auxiliara

Variante speciale ale cozilor :
- Coada cu prioritati (Priority Queue) : Elementele au prioritate, primul element sters va fi cel cu prioritatea cea mai mare, nu cel introdus primul.
- Coada dublu terminata (Deque) : Permite adaugari si stergeri la ambele capete ale cozii.

Implementari frecvente ale cozilor :
Cozile pot fi implementate folosind :
- Tablouri circulare (vectori dinamici)
- Liste inlantuite