

## COLLECTION, BAG, MULTI-SET

**post:** *caută* = adevărat      dacă e ∈ c  
                      fals          în caz contrar

$\dim(c)$ 

```
pre: c ∈ Col,
```

**post:**  $dim$  = dimensiunea colecției  $c$  (numărul de elemente)  $\in \mathcal{N}$

***vidă*(c)**

```
pre: c ∈ Col
```

**post:** *vidă*= adevărat      în cazul în care c e colecția vidă  
fals      în caz contrar

$$iterator(c, i)$$

```
pre: c ∈ Col
```

**post:**  $i \in I$ ,  $i$  este un iterator pe colecția  $c$

$$\mathit{distrug}e(c)$$

```
pre: c ∈ Col
```

**post:** colecția c a fost 'distrusă' (spațiul de memorie alocat a fost eliberat)

Menționăm că pot fi definite în interfața Tipului Abstract de Date Colecție și operații specifice cum ar fi: reuniunea, intersecția, diferența a două colecții.

Deoarece colecția are o operație care furnizează un iterator pe elementele sale, subalgoritmul care va tipări elementele unei colecții *c* poate fi descris sub forma:

**Subalgoritmul *tipărire(c)* este**

{pre: c este o colecție}

```
{post: se tipăresc elementele colecției}
```

<b>iterator(c,i)</b>	{ colecția își construiește iteratorul }
----------------------	--

**CâtTimp** valid(i) **execută** { cât timp iteratorul e valid }

element(i, e)	{se obține elementul curent din iterație}
---------------	---

**@** tipărește e {se tipărește elementul curent}

următor(i)                      {se deplasează iteratorul}

## SfCâtTimp

## SfTipărire

Complexitatea-timp a subalgoritmului de tipărire este  $\theta(|c|)$ , unde prin  $|c|$  am notat dimensiunea colecției  $c$ .

Ca și modalități de reprezentarea ale unei colecții, avem cel puțin două astfel de posibilități:

- se reprezintă toate elementele colecției : 1, 2, 1, 4, 3, 1, 4, 2, 5;
- se reprezintă colecția sub forma unor perechi de forma  $(e_1, f_1), (e_2, f_2), \dots, (e_n, f_n)$ , unde  $e_1, e_2, \dots, e_n$  reprezintă elementele distincte din colecție, iar  $f_1, f_2, \dots, f_n$  reprezintă frecvențele de apariție (numărul de apariții în colecție) a elementelor corespunzătoare: spre exemplu colecția anterioară s-ar reprezenta sub forma perechilor (1, 3), (2, 2), (4, 2), (3, 1), (5, 1).

Modalități de implementare ale colecțiilor ar putea fi folosind:

- tablouri (dinamice);
- liste înlanțuite;
- tabele de dispersie;
- arbori binari.

**TEMA**. Implementați operațiile specifice TAD Colecție folosind SD menționate. Studiați complexitatea operațiilor în funcție de SD aleasă pentru implementare.