



## 2. előadás

# Struktúra nélküli adatszerkezetek

Reprezentáció és implementáció, algoritmusok, a szabad helyek kezelése, a halmaz és a multihalmaz

*Adatszerkezetek és algoritmusok* előadás  
2018. február 13.

Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek


A halmaz

A multihalmaz

Kósa Márk, Pánovics János és Szathmáry László  
Debreceni Egyetem  
Informatikai Kar

# Reprezentáció és implementáció

Absztrakt adatszerkezetek grafikus (képi) megjelenítésénél használt jelölések:

 : adatelem

 : kapcsolat két adatelem között

Amikor egy absztrakt adatszerkezethez megadjuk a tárolási módját és a leképezését, akkor megadjuk az absztrakt adatszerkezet **reprezentációját**.

absztrakt adatszerkezet  $\xrightarrow{\text{leképezés}}$  ábrázolás

Ha a reprezentáció mellé megadjuk a műveletek megvalósítását is, akkor megadjuk az absztrakt adatszerkezet **implementációját**.





Olyan eljárás (**elemi lépések sorozata**), melynek során a következők teljesülnek:

- jól meghatározott objektumokon jól meghatározott műveleteket végzünk
- minden lépés elvégzése után egyértelműen definiált helyzet áll elő
- véges sok lépés után véget ér
- nem csak egy feladatra, hanem egy feladatosztály tagjaira érvényes

Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai  
Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz  
A multihalmaz



Algoritmusokat a következő módokon lehet megadni:

- természetes (beszélt) emberi nyelven
- pontokba szedett természetes nyelvi „utasításokkal”
- folyamatábrával
- pszeudonyelvvel (lásd gyakorlaton)
- valamilyen programozási nyelven

Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz

A multihalmaz

## Példa elefánttal és zsiráffal



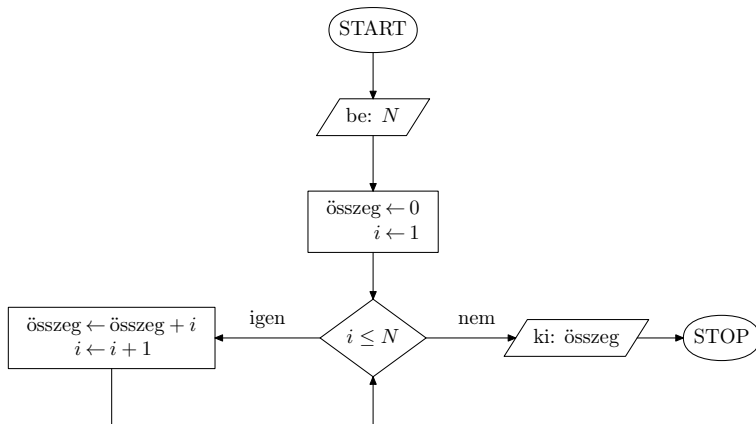
### Hogyan tegyük be egy elefántot a hűtőszekrénybe?

- 1 Nyissuk ki a hűtőszekrény ajtaját.
- 2 Tegyük be az elefántot a hűtőszekrénybe.
- 3 Zárjuk be a hűtőszekrény ajtaját.

### Hogyan tegyük be egy zsiráfot a hűtőszekrénybe?

- 1 Nyissuk ki a hűtőszekrény ajtaját.
- 2 Vegyük ki az elefántot a hűtőszekrényből.
- 3 Tegyük be a zsiráfot a hűtőszekrénybe.
- 4 Zárjuk be a hűtőszekrény ajtaját.

# Példa folyamatábrával





## Vezérlési szerkezetek

- 1 szekvencia (utasítások végrehajtása a felírás sorrendjében)
- 2 szelekció (elágazások)
- 3 iteráció (ciklusok)

Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz

A multihalmaz



Az adatszerkezetek tárolásához **memória** kell, ami **véges**. A szabad helyekkel gazdálkodni kell! Lehetséges módszerek

- folytonos ábrázolás esetén:
  - szabad tárhelyek összegyűjtése a lefoglalt tárterület végén (időigényes)
  - szemétygyűjtögetés (garbage collection) elemmozgatással
  - minden tárhelyhez hozzárendelünk egy bitet, ami a foglaltságot jelzi (nincs elemmozgatás)
- szétszórt ábrázolás esetén:
  - szabad helyek láncolt listája (probléma: különböző méretű tárhelyek)
  - szemétygyűjtögetés (garbage collection) a szabad helyek láncolásával
  - a szabad helyek nyilvántartása bitvektor segítségével

Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

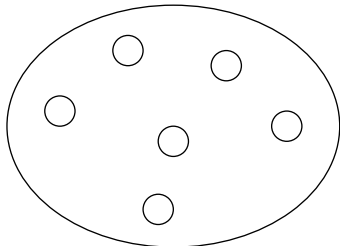
A halmaz

A multihalmaz



# Halmaz és multihalmaz

A halmaz és a multihalmaz **struktúra nélküli**, **homogén** és **dinamikus** adatszerkezetek.



A halmaz minden eleme különböző. A multihalmazban előfordulhatnak azonos elemek is.

Mindkét adatszerkezetre igaz, hogy az adatszerkezetben lévő elemek között **nincs kapcsolat** (ezért struktúra nélküli adatszerkezetek).



# A halmaz adatszerkezet



A **halmaz** adatszerkezet a matematikai halmaz fogalom megjelenése az adatszerkezetek szintjén. Mindig **véges** – ennyiben nem felel meg teljesen a matematikai halmaz fogalmának.

## A halmaz alapműveletei

- **elem**,  $\in$ : megmondja, hogy egy adatelem benne van-e a halmazban vagy sem
- **unió**,  $\cup$ : két halmaz unióját adja
- **metszet**,  $\cap$ : két halmaz metszetét adja
- **különbség**,  $\setminus$ : két halmaz különbségét adja

Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz

A multihalmaz



Az adatszerkezetekkel végezhető hagyományos műveletek megvalósítása halmazok esetén:

- **Létrehozás** kétféleképpen:
  - explicit módon, a halmaz elemeinek felsorolásával (esetleg üresen)
  - egy predikátum segítségével
- **Bővítés** unióképzéssel
- **Törlés** csak fizikai, különbségképzéssel
- **Csere** nincs
- **Rendezés, keresés, elérés, bejárás** nem értelmezettek
- **Feldolgozás** a halmaz alapműveleteinek a segítségével

Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

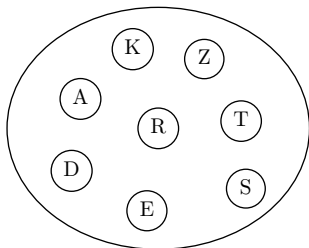
Struktúra nélküli  
adatszerkezetek

A halmaz

A multihalmaz

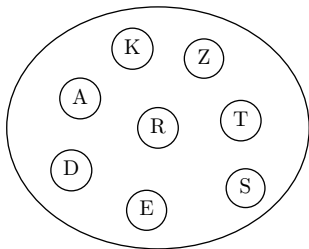
## A halmaz adatszerkezet reprezentációja

Klasszikus reprezentációja folytonosan, **karakterisztikus függvény** segítségével történik.



## A halmaz adatszerkezet reprezentációja

Klasszikus reprezentációja folytonosan, **karakterisztikus függvény** segítségével történik.

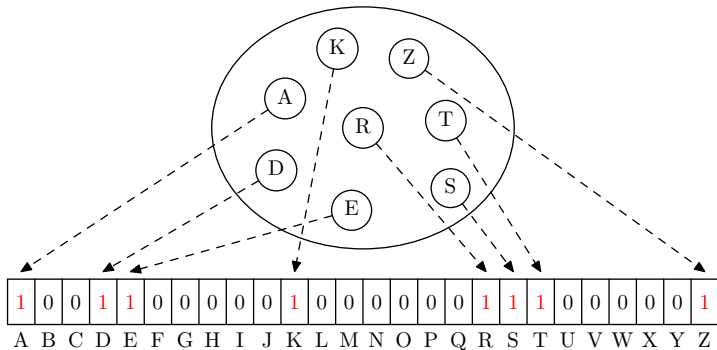


A halmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy egy bit méretű tárterületet.



## A halmaz adatszerkezet reprezentációja

Klasszikus reprezentációja folytonosan, **karakterisztikus függvény** segítségével történik.



A halmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy egy bit méretű tárterületet. Az adott értékű adatelemhez tartozó bit fogja jelezni, hogy az adatelem benne van-e a halmazban (1) vagy sem (0).



Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz

A multihalmaz

# A halmaz adatszerkezet implementációja

Folytonos reprezentáció esetén a halmaz alapműveleteinek megvalósítása visszavezethető egyszerű bitműveletekre:

## Unióképzés

$$x \in A \cup B \Leftrightarrow x \in A \vee x \in B$$

## Metszetképzés

$$x \in A \cap B \Leftrightarrow x \in A \wedge x \in B$$

## Különbségképzés

$$x \in A \setminus B \Leftrightarrow x \in A \wedge x \notin B$$



# A halmaz adatszerkezet megjelenése adattípusként a programozási nyelvekben

## Java

Set interfész, melynek műveletei (többek között):

- `isEmpty()`
- `contains(Object o), containsAll(Collection<?> c)`
- `add(E e), addAll(Collection<? extends E> c)`
- `remove(Object o), removeAll(Collection<?> c)`
- `size()`

## C++

set generikus osztály, melynek műveletei (többek között):

- `empty()`
- `size()`
- `insert(...)` többféle specifikációval
- `erase(...)` többféle specifikációval
- `find(const key_type& x)`



Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz

A multihalmaz



# A halmaz adatszerkezet megjelenése adattípusként a programozási nyelvekben

## Python

set osztály, melynek műveletei (többek között):

- `len(s)`
- `x in s`
- `issubset(other)`
- `issuperset(other)`
- `union(other)`
- `intersection(other)`
- `difference(other)`
- `add(elem)`
- `remove(elem)`
- `clear()`



Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz

A multihalmaz

# A halmaz adatszerkezet megjelenése adattípusként a programozási nyelvekben

## C#

HashSet<T> generikus kollekción osztály, melynek műveletei (többek között):

- Add(T item)
- Remove(T item)
- Clear()
- UnionWith(IEnumerable<T> other)
- IntersectWith(IEnumerable<T> other)
- ExceptWith(IEnumerable<T> other)
- Contains(T item)
- IsSubsetOf(IEnumerable<T> other)
- IsSupersetOf(IEnumerable<T> other)
- Overlaps(IEnumerable<T> other)
- SetEquals(IEnumerable<T> other)



Reprezentáció és  
implementáció

Algoritmusok

Algoritmusok megadásának  
módjai

Algoritmusok építőelemei

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz

A multihalmaz

## A multihalmaz adatszerkezet

A **multihalmaz** abban különbözik a halmaztól, hogy megengedi az adatelemek ismétlődését, benne több azonos értékű elem is előfordulhat.

### A multihalmaz alapműveletei

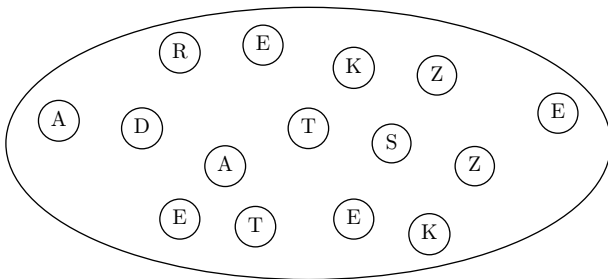
- **elem**,  $\in$ : megmondja, hogy egy adatelem benne van-e a multihalmazban vagy sem
- **unió**,  $\cup$ : két multihalmaz unióját adja
- **metszet**,  $\cap$ : két multihalmaz metszetét adja
- **különbség**,  $\setminus$ : két multihalmaz különbségét adja

Multihalmazoknál az adatszerkezetekkel végezhető hagyományos műveletek megvalósítása hasonló a halmazokéhoz (lásd ott). A multihalmaz **feldolgozása** a multihalmaz alapműveleteinek a segítségével történik.



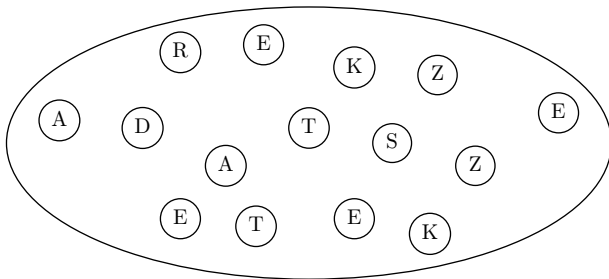
## A multihalmaz adatszerkezet reprezentációja

Klasszikus reprezentációja folytonosan, **karakterisztikus függvény** segítségével történik.



## A multihalmaz adatszerkezet reprezentációja

Klasszikus reprezentációja folytonosan, **karakterisztikus függvény** segítségével történik.



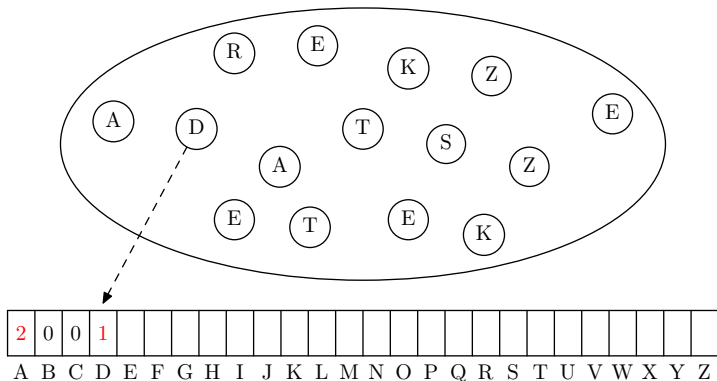
A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot).





## A multihalmaz adatszerkezet reprezentációja

Klasszikus reprezentációja folytonosan, **karakterisztikus függvény** segítségével történik.

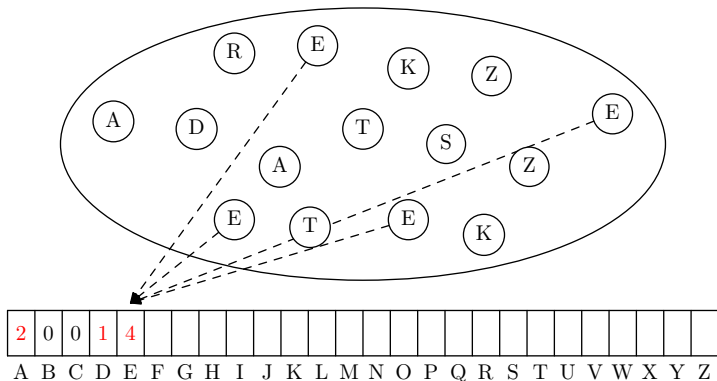


A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájt). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.



## A multihalmaz adatszerkezet reprezentációja

Klasszikus reprezentációja folytonosan, **karakterisztikus függvény** segítségével történik.



A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.







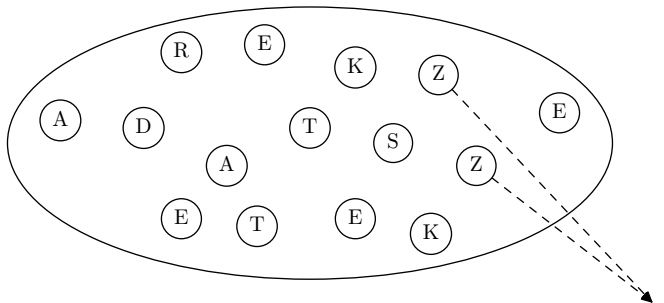






## A multihalmaz adatszerkezet reprezentációja

Klasszikus reprezentációja folytonosan, **karakterisztikus függvény** segítségével történik.



2	0	0	1	4	0	0	0	0	0	0	2	0	0	0	0	0	0	1	1	2	0	0	0	0	0	2
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	

A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.



## A multihalmaz adatszerkezet implementációja

Folytonos reprezentáció esetén a multihalmaz alapműveleteinek megvalósítása visszavezethető egyszerű aritmetikai (számtani) műveletekre:

### Unióképzés

$$x \text{ A} \cup \text{B-ben} = x \text{ A-ban} + x \text{ B-ben}$$

### Metszetképzés

$$x \text{ A} \cap \text{B-ben} = \min\{x \text{ A-ban}, x \text{ B-ben}\}$$

### Különbségképzés

$$x \text{ A} \setminus \text{B-ben} = \max\{0, x \text{ A-ban} - x \text{ B-ben}\}$$

