



## 2. előadás

# Struktúra nélküli adatszerkezetek

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

Adatszerkezetek és algoritmusok előadás  
2011. február 16.

## Reprezentáció és implementáció

## Algoritmusok megadásának módjai

## Gazdálkodás a memóriával

## Struktúra nélküli adatszerkezetek


A halmaz

A multihalmaz

Kósa Márk és Pánovics János  
Debreceni Egyetem  
Informatikai Kar



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**.



## Reprezentáció és implementáció

## Algoritmusok megadásának módjai

## Gazdálkodás a memóriával

## Struktúra nélküli adatszerkezetek

A halmaz

### A multihalmaz

A műveletek **algoritmusok**. Algoritmusokat a következő módokon lehet megadni:

- természetes (beszélt) emberi nyelven
- mondatszerű leírással
- folyamatábrával
- pszeudonyelvvel (lásd gyakorlaton)
- valamilyen programozási nyelven



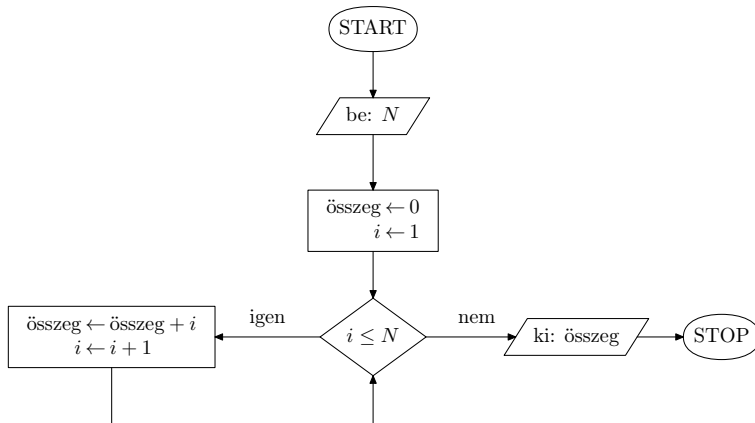
## Hogyan tegyünk be egy elefántot a hűtőszekrénybe?

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

## Hogyan tegyünk 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



- folytonos ábrázolás esetén:
  - szabad tárhelyek összegyűjtése a lefoglalt tárterület végén (időigényes)
  - szemétyűjtögetés (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)
  - szemétyűjtögetés (garbage collection) a szabad helyek láncolásával
  - a szabad helyek nyilvántartása bitvektor segítségével

## Reprezentáció és implementáció

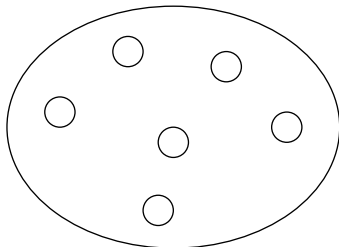
## Algoritmusok megadásának módjai

## Gazdálkodás a memóriával

## Struktúra nélküli adatszerkezetek

A halmaz

A multihalmaz



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).

## Reprezentáció és implementáció

## Algoritmusok megadásának módjai

- ## Gazdálkodás a memóriával

## Struktúra nélküli adatszerkezetek

A halmaz

A multihalmaz





Reprezentáció és  
implementáció

Algoritmusok  
megadásának módjai

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz

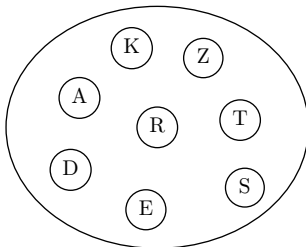
A multihalmaz

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 alapl műveleteinek a segítségével

## A halmaz adatszerkezet reprezentációja

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



## Reprezentáció és implementáció

## Algoritmusok megadásának módjai

## Gazdálkodás a memóriával

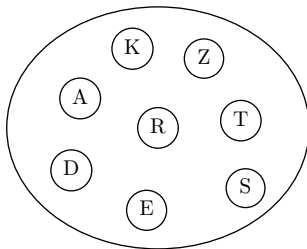
## Struktúra nélküli adatszerkezetek

A halmaz

A multihalmaz

## A halmaz adatszerkezet reprezentációja

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

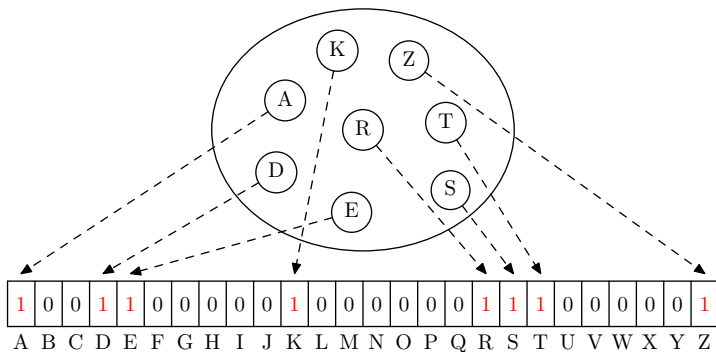


A halmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy egy bit méretű tárterületet.



## 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  
megadásának módjai

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

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

Reprezentáció és  
implementáció

Algoritmusok  
megadásának módjai

Gazdálkodás a  
memóriával

Struktúra nélküli  
adatszerkezetek

A halmaz

A multihalmaz



# A halmaz adatszerkezet megjelenése adattípusként a programozási nyelvekben



## C#

`HashSet<T>` generikus kollekció 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 megadásának módjai

## Gazdálkodás a memóriával

## Struktúra nélküli adatszerkezetek

A halmaz

A multihalmaz

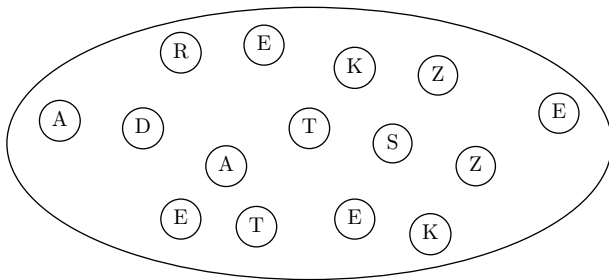
- **elemé,  $\in$** : megmondja, hogy egy adatelem benne van-e a multihalmazban vagy sem
- **unió,  $\cup$** : két multihalmaz unióját adja
- **metszet,  $\cap$** : két multihalmaz metszetét adja
- **különbség,  $\setminus$** : két multihalmaz különbségét adja

214



## Struktúra nélküli adatszerkezetek

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



## Algoritmusok megadásának módjai

## Gazdálkodás a memóriával

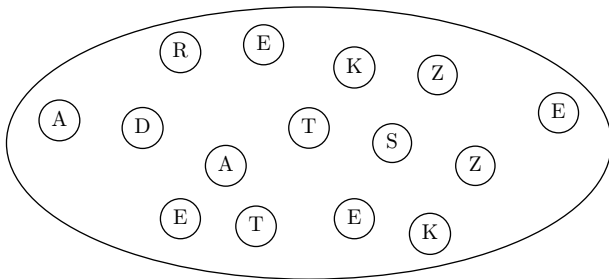
## Struktúra nélküli adatszerkezetek

A halmaz

A multihalmaz

## A multihalmaz adatszerkezet reprezentációja

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

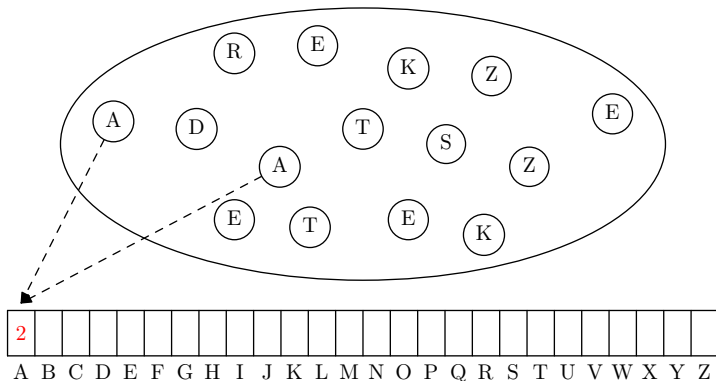


A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot).



## A multihalmaz adatszerkezet reprezentációja

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

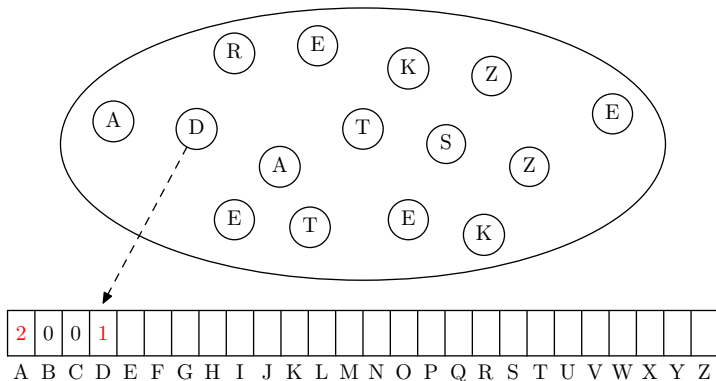


A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.



## A multihalmaz adatszerkezet reprezentációja

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

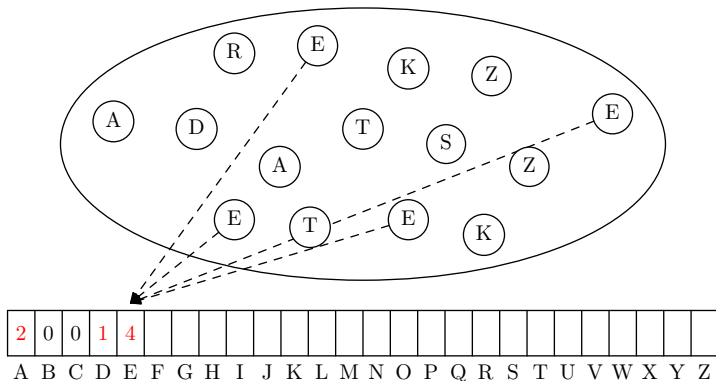


A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.



## A multihalmaz adatszerkezet reprezentációja

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

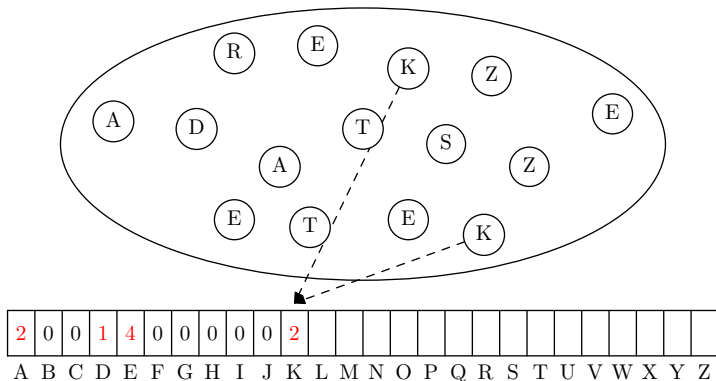


A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.



## A multihalmaz adatszerkezet reprezentációja

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

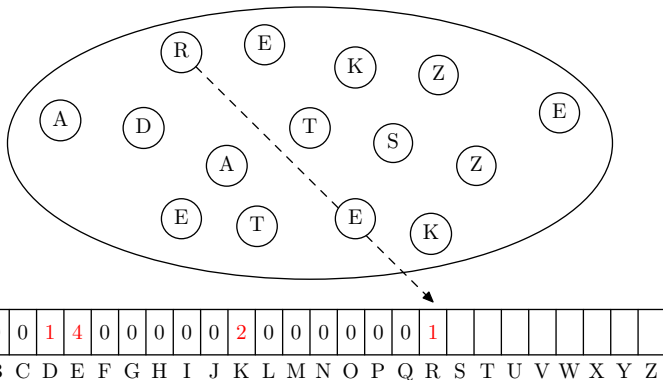


A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.



## A multihalmaz adatszerkezet reprezentációja

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

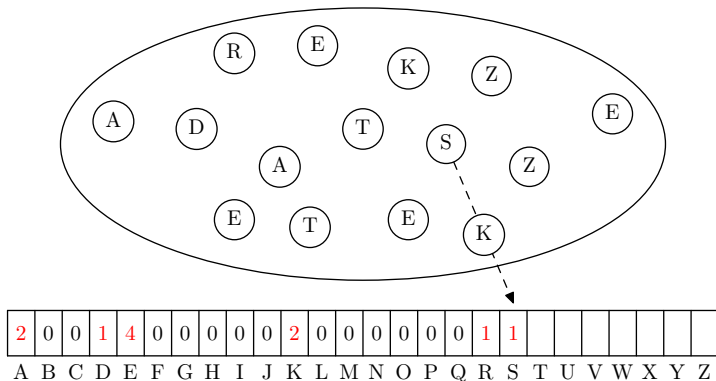


A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.



## A multihalmaz adatszerkezet reprezentációja

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



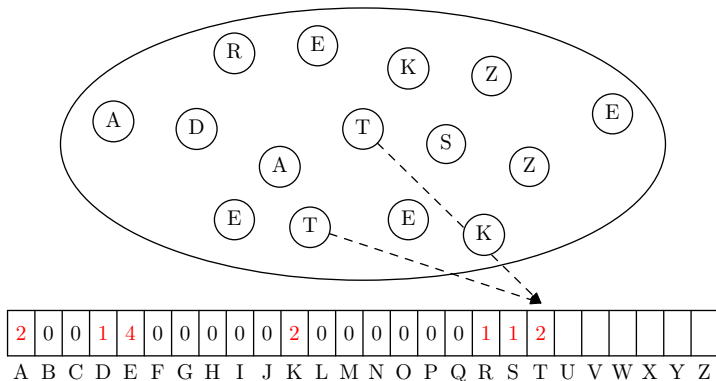
A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.





## A multihalmaz adatszerkezet reprezentációja

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

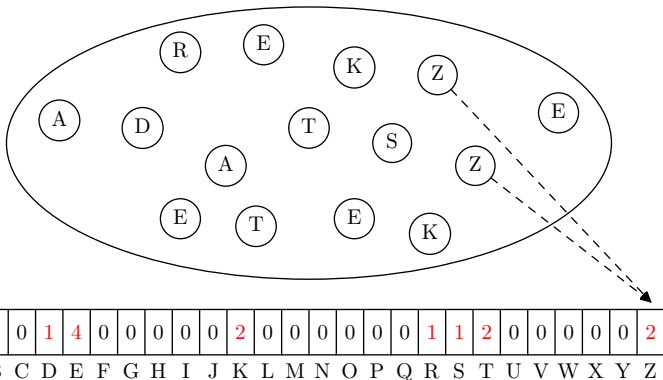


A multihalmaz lehetséges elemeit sorba rendezzük, s mindegyikhez hozzárendelünk egy tárterületet (általában 1 bájtot). A tárhelyeken az azonos értékű elemek **előfordulásainak számát** tároljuk.



## A multihalmaz adatszerkezet reprezentációja

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



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



## Reprezentáció és implementáció

## Algoritmusok megadásának módjai

## Gazdálkodás a memóriával

## Struktúra nélküli adatszerkezetek

A halmaz

A multihalmaz

A multihalmaz