

1. előadás

Bevezető

Absztrakció, absztrakt adatszerkezetek, ábrázolási módok

Adatszerkezetek és algoritmusok előadás
2018. február 6.



Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás

Absztrakt
adatszerkezetek

Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétszórt (láncolt) tárolás

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



Előfeltételek

Szak	Tárgykód	Előfeltétel
PTI	INDK421	Bevezetés az informatikába (INDK201)

Tantárgy honlapja

<https://hallg.inf.unideb.hu/honlap/adatszerk>

Előadó honlapja

<https://arato.inf.unideb.hu/szathmary.laszlo>

Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás

Absztrakt
adatszerkezetek

Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétszórt (láncolt) tárolás



Előadás

INDK421E	Szathmáry László	IK-202	K 8–10
----------	------------------	--------	--------

kivéve: 2018. április 3-án és május 1-jén az előadás nem lesz megtartva.

Gyakorlatok

Szathmáry László	IK-F02	Sze 8–10
Szathmáry László	IK-202	Sze 10–12

kivéve: 2018. április 4-én a gyakorlatok nem lesznek megtartva.

Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás

Absztrakt
adatszerkezetek

Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétosztott (láncolt) tárolás



Gyakorlat

A szorgalmi időszakban két 50 perces (40 és 60 pontos) zárthelyi dolgozat megírására kerül sor. Ezek időpontjai és helyszínei:

- 1 2018. ???, csütörtök ...
- 2 2018. ???, csütörtök ...

A gyakorlati aláírás megszerzéséhez a két zárthelyi dolgozatot (külön-külön) legalább **50%-os** eredménnyel kell teljesíteni, és összesen legalább **51 pontot** kell összegyűjteni.

Előadás

A kollokvium **írásban**, az elméleti anyag, a fogalmak, az absztrakt adatszerkezetek és algoritmusok számonkérésével történik.

Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás






Absztrakt
adatszerkezetek

Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétszórt (láncolt) tárolás

Ajánlott irodalom

-  Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest: *Algoritmusok*, Műszaki Könyvkiadó, Budapest, 1997.
-  Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: *Új algoritmusok*, Scolar Informatika, Budapest, 2003.
-  Rónyai Lajos, Ivanyos Gábor, Szabó Réka: *Algoritmusok*, Typotex, Budapest, 2008.
-  Donald E. Knuth: *A számítógépprogramozás művészete 1. (Alapvető algoritmusok)*, Műszaki Könyvkiadó, Budapest, 1994.
-  Donald E. Knuth: *A számítógépprogramozás művészete 3. (Keresés és rendezés)*, Műszaki Könyvkiadó, Budapest, 1994.

Bevezető

Kósa Márk
Pánovics János
Szathmáry László



Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás





Absztrakt
adatszerkezetek

Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétosztott (láncolt) tárolás

Ajánlott irodalom (folyt.)

-  Seymour Lipschutz: *Adatszerkezetek*, Panem–McGraw-Hill, Budapest, 1993.
-  Morvay János, dr. Sebők Ferenc: *Számítógépes adatkezelés*, Központi Statisztikai Hivatal, Nemzetközi Számítástechnikai Oktató és Tájékoztató Központ, Budapest, 1981
-  Brian W. Kernighan, Dennis M. Ritchie: *A C programozási nyelv*, Műszaki Könyvkiadó, Budapest, 1996.
-  Juhász István, Kósa Márk, Pánovics János: *C példatár*, Panem, 2004.



Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás

Absztrakt
adatszerkezetek

Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétszórt (láncolt) tárolás

- Elemek: egyedek
- Tulajdonságok (statikus rész)
- Viselkedés (dinamikus rész)
- Kölcsönhatás
- Komplex rendszer
- Nyílt rendszer
- Dinamikus rendszer



- Modellalkotás, absztrakció
- Adatmodell, eljárásmodell
- Adat, információ

Az adatelemek lehetnek **egyszerűek** (atomiak) és **összetettek**. Minden adatelem rendelkezik valamilyen **értékkel**.

Az adatelemek között jól meghatározott kapcsolatrendszer van. Az adatelemek és a közöttük lévő kapcsolatok definiálják a **logikai (absztrakt) adatszerkezetet**. Független hardvertől, szoftvertől.

Fizikai adatszerkezet (társzerkezet): adatszerkezet az operatív tárban vagy periférián (háttértáron).



Lehetséges csoportosítási szempontok:

- 1 Változhat-e az adatszerkezet elemeinek száma?
 - statikus
 - dinamikus
- 2 Milyen az adatszerkezet elemeinek a típusa?
 - homogén
 - heterogén
- 3 Milyen kapcsolatban állnak egymással az adatelemek az adatszerkezetben?

Egy homogén adatszerkezet lehet

- struktúra nélküli
- asszociatív
- szekvenciális
- hierarchikus
- hálós

A heterogén adatszerkezeteket nem csoportosítjuk ilyen szempont alapján.



Absztrakt adatszerkezetekkel végezhető műveletek

- 1 Létrehozás
- 2 Módosítás
 - bővítés
 - törlés (fizikai, logikai)
 - csere
- 3 Rendezés
- 4 Keresés
- 5 Elérés
- 6 Bejárás
- 7 Feldolgozás

Bevezető

Kósa Márk
Pánovics János
Szathmáry László



Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás

Absztrakt
adatszerkezetek

Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétszórt (láncolt) tárolás



Ábrázolás alatt az adatszerkezet memóriában való megjelenési formáját értjük. Ez **minden adatszerkezet** esetén lehet

- folytonos (vektorszerű)
- szétszórt (láncolt)

Az adatelemek számára tárhelyeket foglalunk a memóriában. Egy **tárhely** mindig egy bájtcsoportot jelent, amely egy adatelem értékét tárolja, illetve szerkezetleíró információkat is hordozhat.

Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás

Absztrakt
adatszerkezetek

Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétszórt (láncolt) tárolás



Egy tárhelyen egy adatelem értékét tároljuk. A tárhelyek a memóriában folytonos, **összefüggő** tárterületet alkotnak, a tárhelyek **mérete** azonos.

Előnye:

- **közvetlen elérés**, a kezdőcím és az egy adatelemhez tartozó tárhely méretének ismeretében
- a csere művelete könnyen megvalósítható
- hatékony rendező algoritmusok (pl. gyorsrendezés)
- hatékony kereső algoritmusok (pl. bináris keresés)

Hátránya:

- nem segíti a bővítés és a fizikai törlés műveletének végrehajtását

Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás

Absztrakt
adatszerkezetek

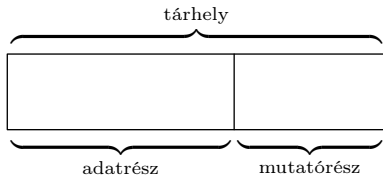
Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétszórt (láncolt) tárolás

Szétszórt (láncolt) tárolás

Egy tárhelyen egy adatelem értékét (**adatrész**) és legalább egy mutató értékét (**mutatórész**) tároljuk. A mutatók értékei memóriacímek lehetnek, amelyek megmondják az adatelem rákövetkezőinek tárbeli helyét. A tárhelyek mérete nem szükségképpen azonos, elhelyezkedésük a memóriában tetszőleges.



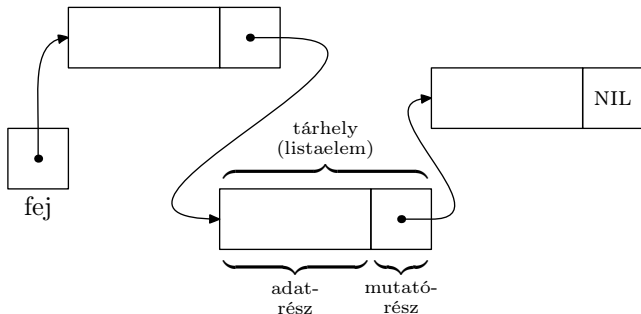
A **szétszórt** ábrázolási mód fajtái:

- egyirányban láncolt lista
- cirkuláris lista
- kétirányban láncolt lista
- multilista



Egyirányban láncolt lista

A tárhely (**listaelem**) az adatelem értékén kívül egy mutatót tartalmaz, amely a következő listaelem címét tartalmazza.



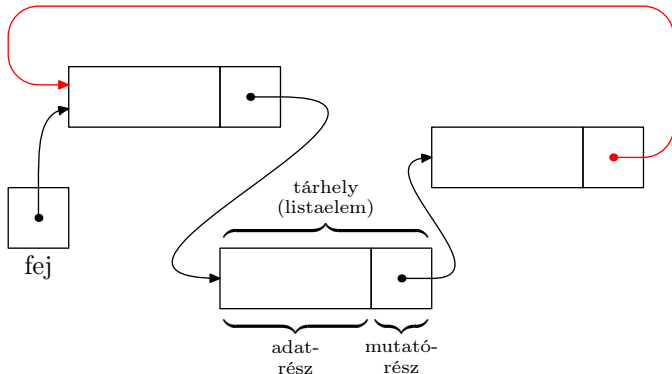
A láncolt lista első elemének tárhelyi címét egy mutató, a **fejmutató** tárolja.

A láncolt lista végét egy speciális érték, a **NIL** érték jelzi. Amennyiben a fejmutató tartalmazza ezt az értéket, akkor az egyirányban láncolt lista **üres**.



Cirkuláris lista

Hasonló az egyirányban láncolt listához, ám itt egyik listaelem mutatórésze sem tartalmazhatja a NIL értéket: az „utolsó” listaelem mutatórészébe az „első” listaelem címe kerül.

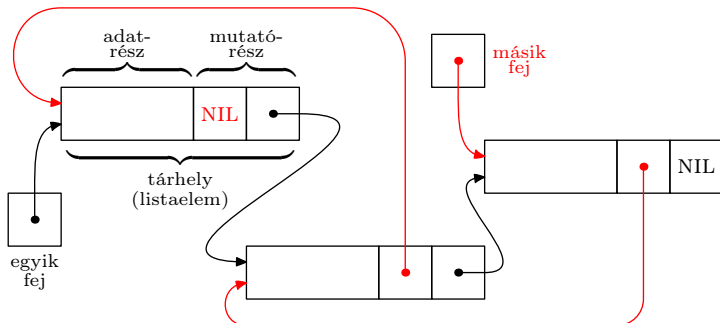


A cirkuláris lista „első” elemének tárbeli címét most is egy mutató, a **fejmutató** tárolja. Amennyiben a fejmutató a NIL értéket tartalmazza, akkor a cirkuláris lista **üres**.



Kétirányban láncolt lista

Hasonló az egyirányban láncolt listához, ám itt minden listaelem mutatórésze **két részből** áll: az egyik mutató az adott listaelemet **megelőző**, a másik az adott listaelemet **követő** listaelemre mutat.

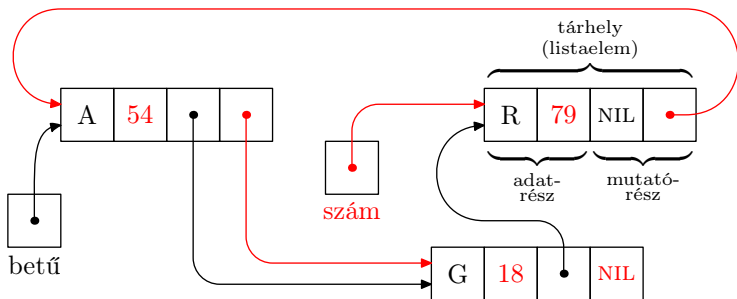


Két lánc alakul ki, **két fejmutatóval**. A fejmutatók a kétirányban láncolt lista **első** és **utolsó** elemére mutatnak. Ha mindkét fejmutató értéke **NIL**, akkor a kétirányban láncolt listának nincs egyetlen eleme sem, azaz **üres**.



Multalista (1)

Ebben a változatban a listaelemek adatrésze összetett. Az adatrész minden komponensére fölépíthető egy egyirányban láncolt lista.



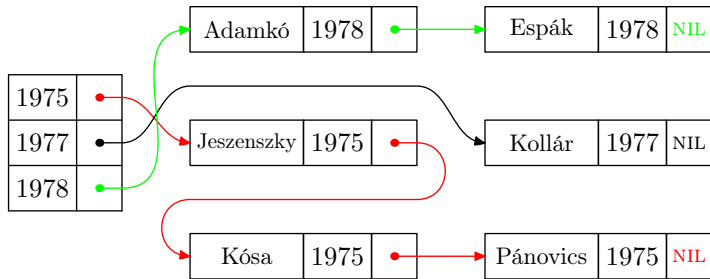
Annyi lánc alakítható ki, ahány komponensből áll az adatrész. Minden lista külön fejmutatóval rendelkezik, és minden listaelem mindegyik láncban előfordul egyszer.



Multalista (2)



Ebben a változatban a listaelemek adata része általában összetett. Az adata rész valamely komponensének értékeit figyelembe véve építjük föl az egyirányban láncolt listákat.



Annyi lánc alakul ki, ahány **különböző** értéket az adata rész adott komponense felvesz. Minden lista külön fejmutatóval rendelkezik, és minden listaelem csak egy láncban szerepel, pontosan egyszer.

Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás

Absztrakt
adatszerkezetek

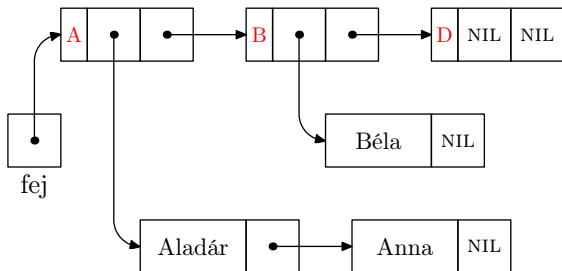
Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szétszórt (láncolt) tárolás



Ebben a változatban a listaelemek tartalmaznak egy-egy **fejmutatót** is, melyek **újabb** láncolt listák első elemeire mutatnak.



Az allisták szerkezete **eltérhet** a főlista szerkezetétől.

Általános tudnivalók

Rendszerelmélet

Absztrakció,
modellalkotás

Absztrakt
adatszerkezetek

Ábrázolási módok

Folytonos (vektorszerű)
tárolás

Szészórt (láncolt) tárolás