1. előadás

Bevezető

Absztrakció, absztrakt adatszerkezetek, ábrázolási módok

Adatszerkezetek és algoritmusok előadás
2011. február 9.

Kósa Márk és Pánovics János
Debreceni Egyetem
Informatikai Kar

Előfeltételek

Előfeltételek

Szak	Tárgykód	Előfeltétel
PTI	INDK421	Bevezetés az informatikába (INDK201)
PM, PTM, IT	I1202	Az informatika alapjai (I1201)

Kósa Márk
Pánovics János

Időpontok és termek

Előadás

INDK421E I1202E	Pánovics János	I	Sz 10–12	kivéve: III. 30.

Gyakorlatok

Dr. Kósa Márk	M114	H 10–12
Dr. Kósa Márk	M418	Sz 16–18
Dr. Kósa Márk	M418	Sz 18–20
Pánovics János	M125	Sz 14–16
Pánovics János	M125	Sz 18–20

Kósa Márk
Pánovics János

Követelmények

Gyakorlat

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

1. 2011. március 24., csütörtök 18 és 20 óra között
2. 2011. május 12., csütörtök 18 és 20 óra között

A gyakorlati aláírás megszerzéséhez a két zárthelyi dolgozatot (külön-külön) legalább 40%-os eredménnyel kell teljesíteni, és összesen legalább 50 pontot kell összegyűjteni.

Előadás

A vizsgaidőszakban minden héten várható egy vizsgaalkalom. A kollokvium írásban, az elméleti anyag, a fogalmak, az absztrakt adatszerkezetek és algoritmusok számonkérésével történik.

Kósa Márk
Pánovics János

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.
- 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.
- 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

Rendszer

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

Kósa Márk
Pánovics János

Absztrakció, modellezés

- 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).

Kósa Márk
Pánovics János

Absztrakt adatszerkezetek osztályozása

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.

Kósa Márk
Pánovics János

Absztrakt adatszerkezetekkel végezhető műveletek

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

Kósa Márk
Pánovics János

Ábrázolási (tárolási) módok

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.

Kósa Márk
Pánovics János

Folytonos (vektorszerű) 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)

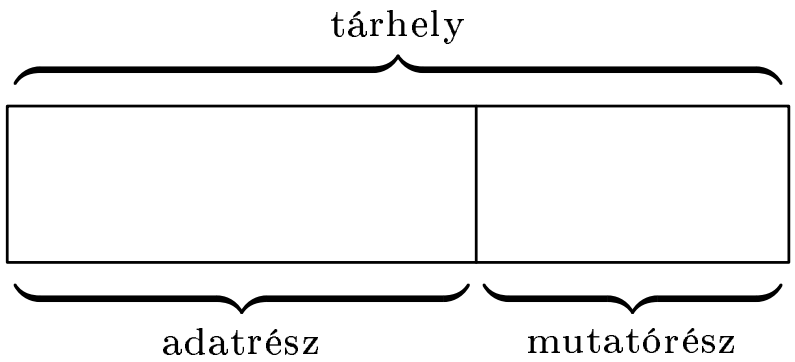
Hátránya:

- nem segíti a bővítés és a fizikai törlés műveletének végrehajtását

Kósa Márk
Pánovics János

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.



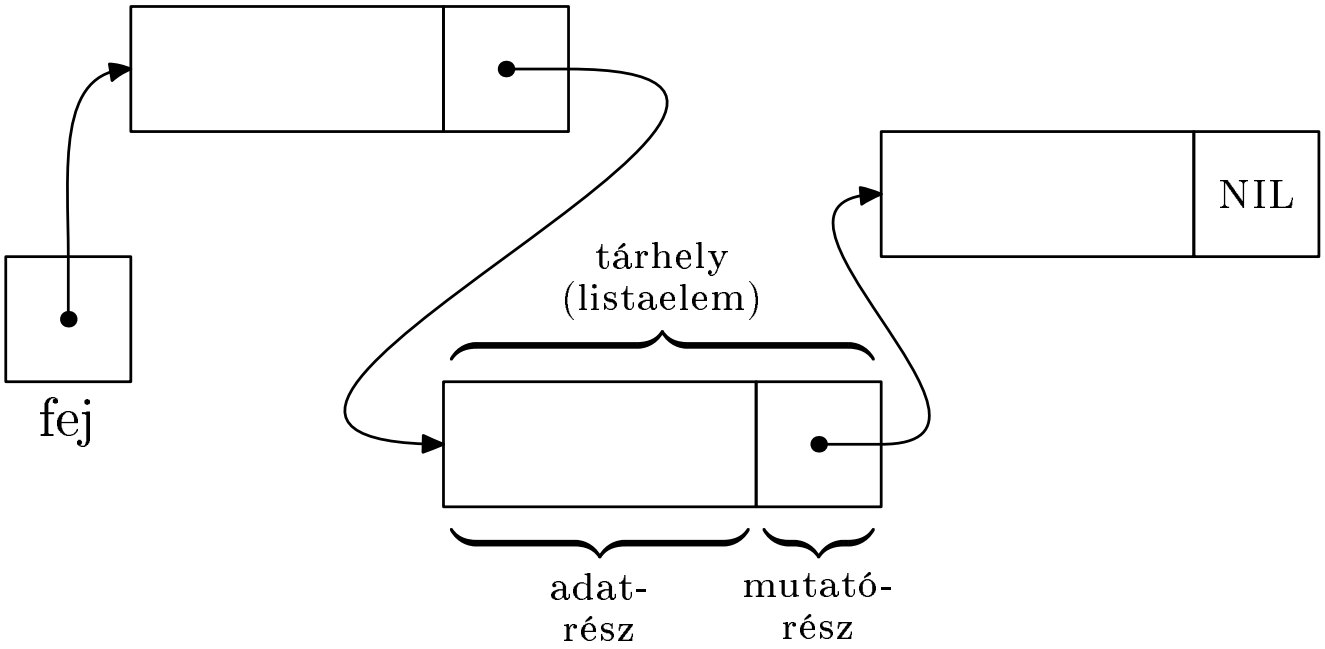
A szétszórt ábrázolási mód fajtái:

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

Kósa Márk
Pánovics János

Egyirányban láncolt lista

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

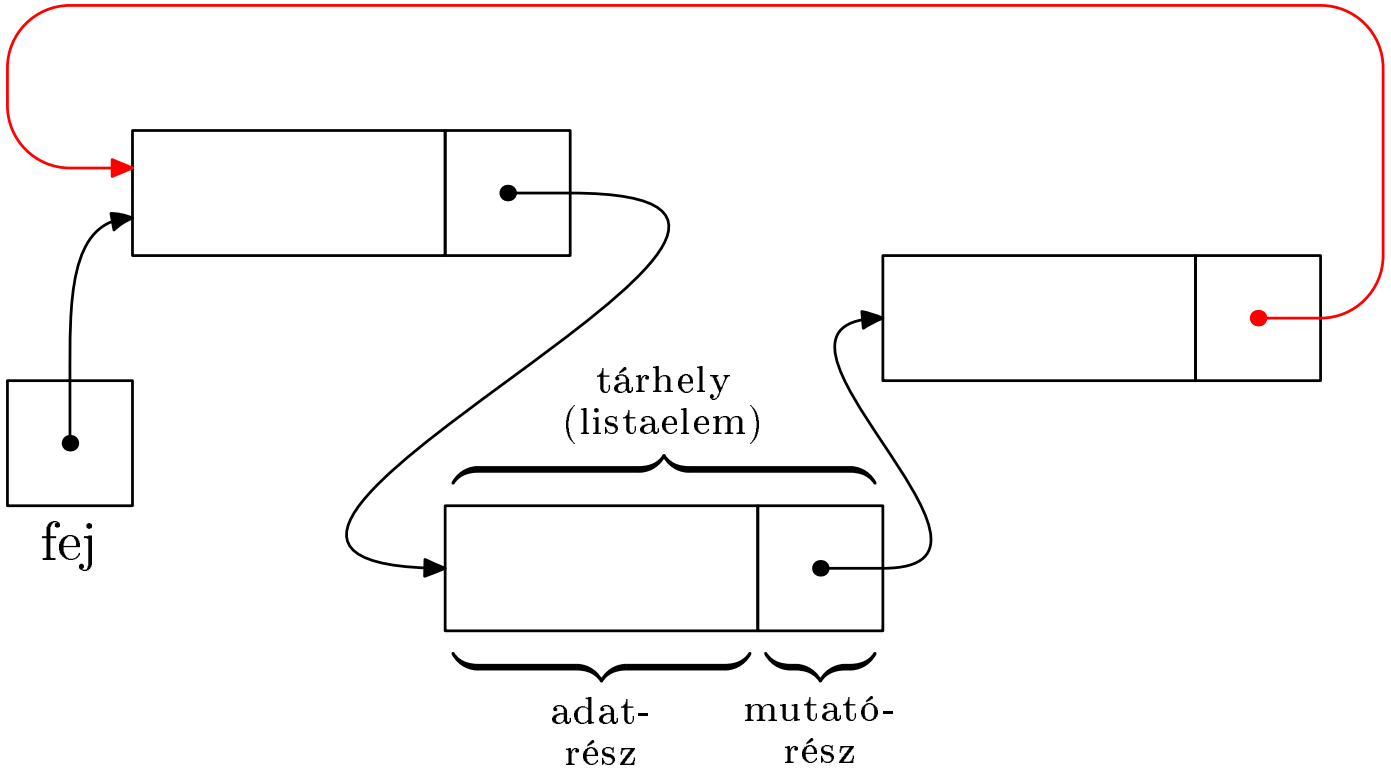


A láncolt lista első elemének tárbeli címét egy mutató, a fejmutató tárolja.
A láncolt lista végét egy speciális érték, a NIL érték jelzi.
Amennyiben a fejmutató tartalmazza ezt az értéket, akkor az egyirányban láncolt lista üres.

Kósa Márk
Pánovics János

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.

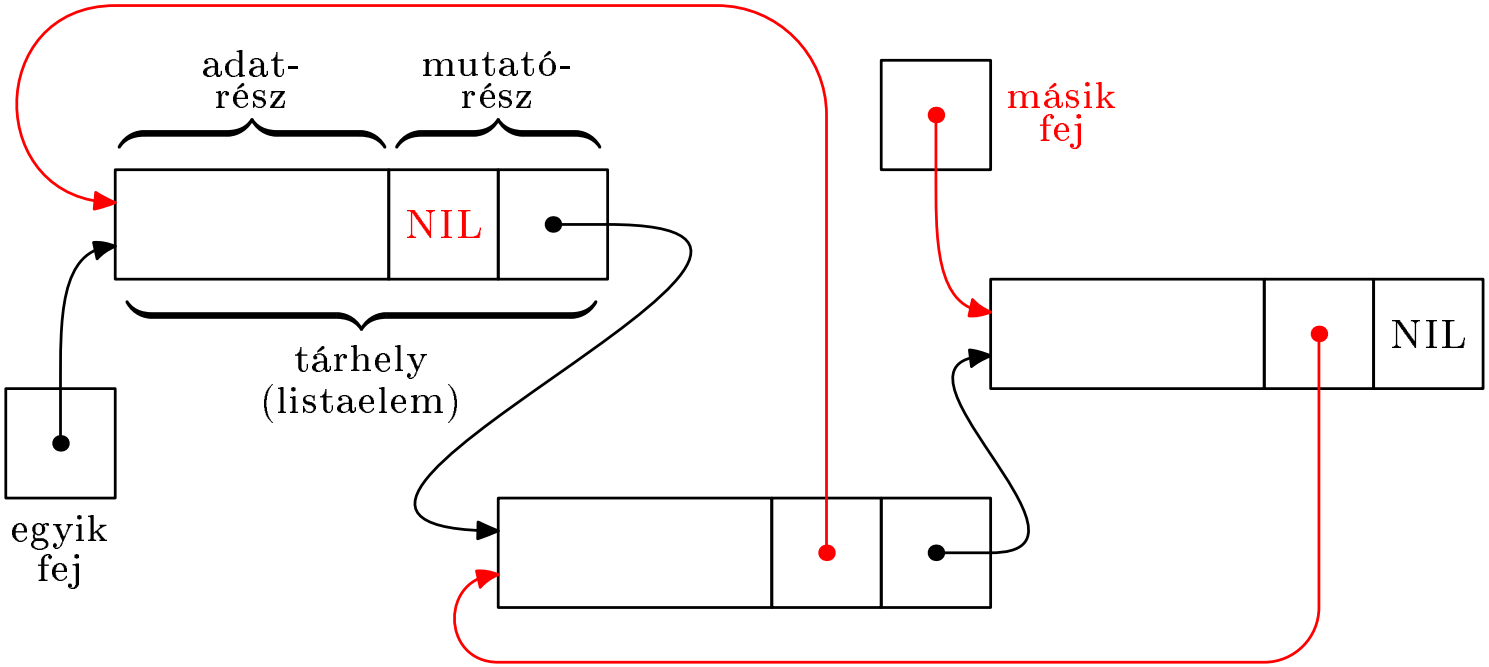


A cirkuláris lista „első” elemének tárbeli címét most is egy mutató, a fejmutató tárolja. Amennyiben a fejmutató a NIL értéket tartalmazza, akkor a cirkuláris lista üres.

Kósa Márk
Pánovics János

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.

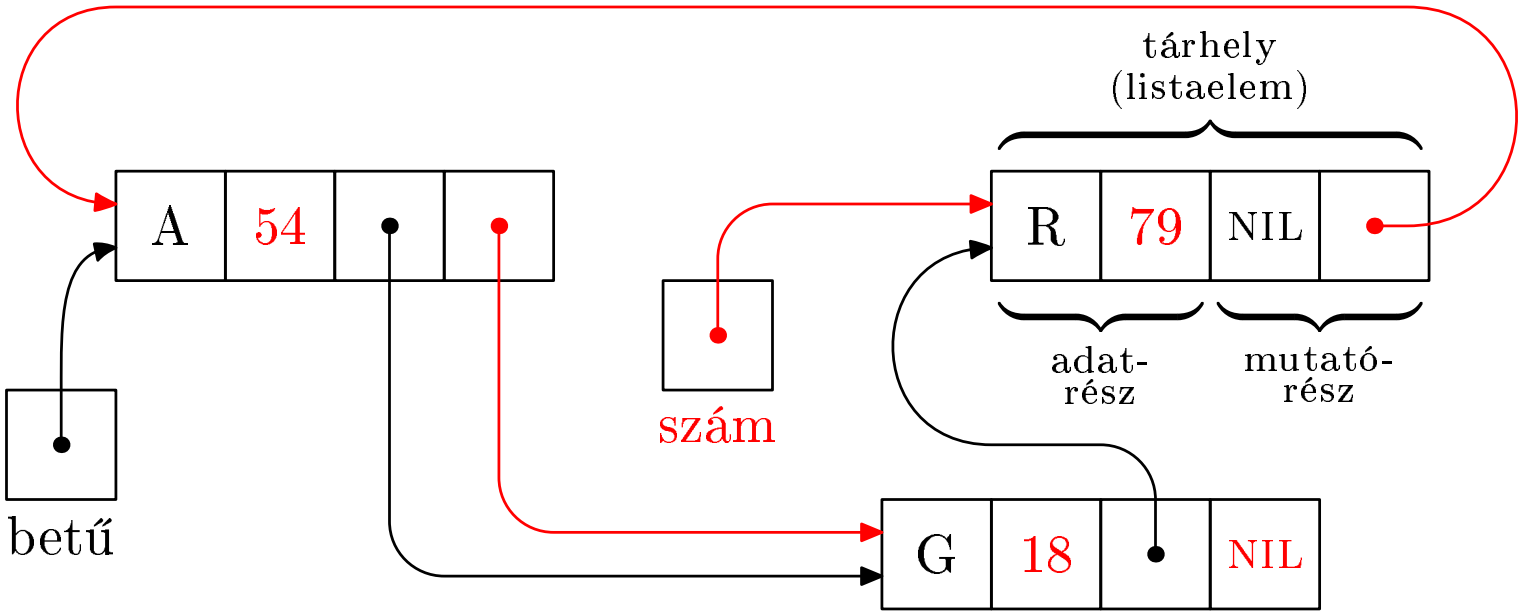


Két lánc alakul ki, két fejmutatóval. A fejmutatók a kétirányban láncolt lista első és utolsó elemére mutatnak. Ha mindkét fejmutató értéke NIL, akkor a kétirányban láncolt listának nincs egyetlen eleme sem, azaz üres.

Kósa Márk
Pánovics János

Multilista (1)

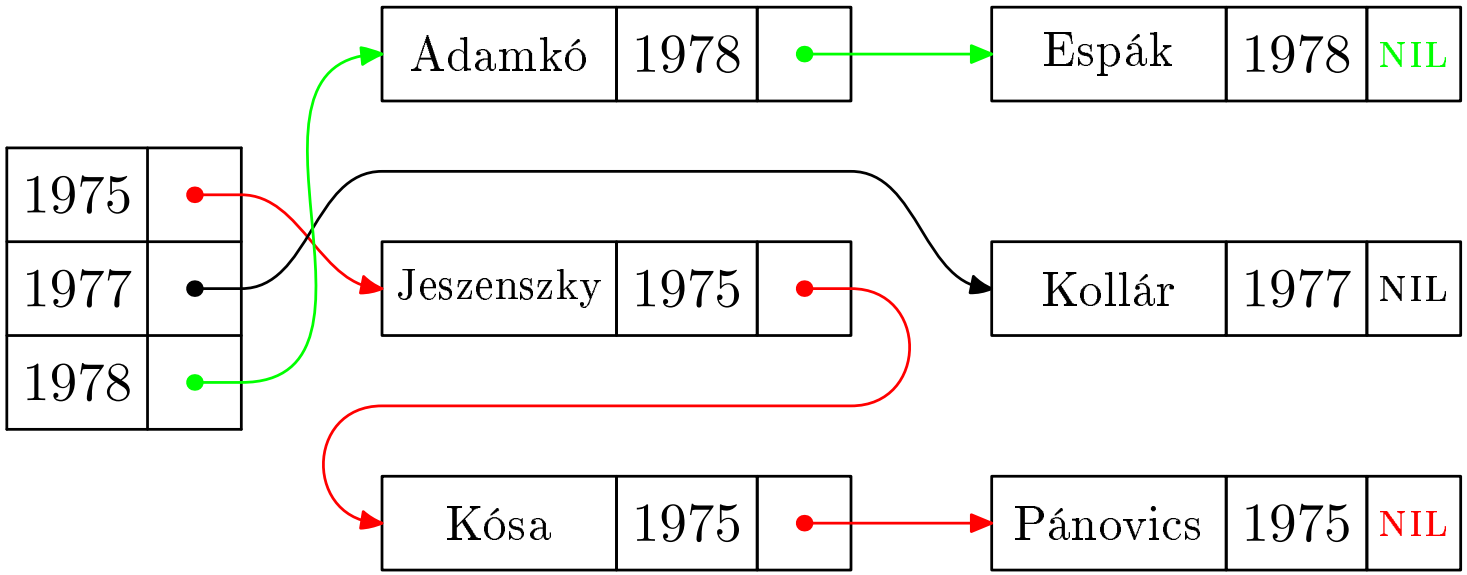
Ebben a változatban a listaelemek adatrésze összetett. Az adatrész minden komponensére fölépíthető egy egyirányban láncolt lista.



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

Multilista (2)

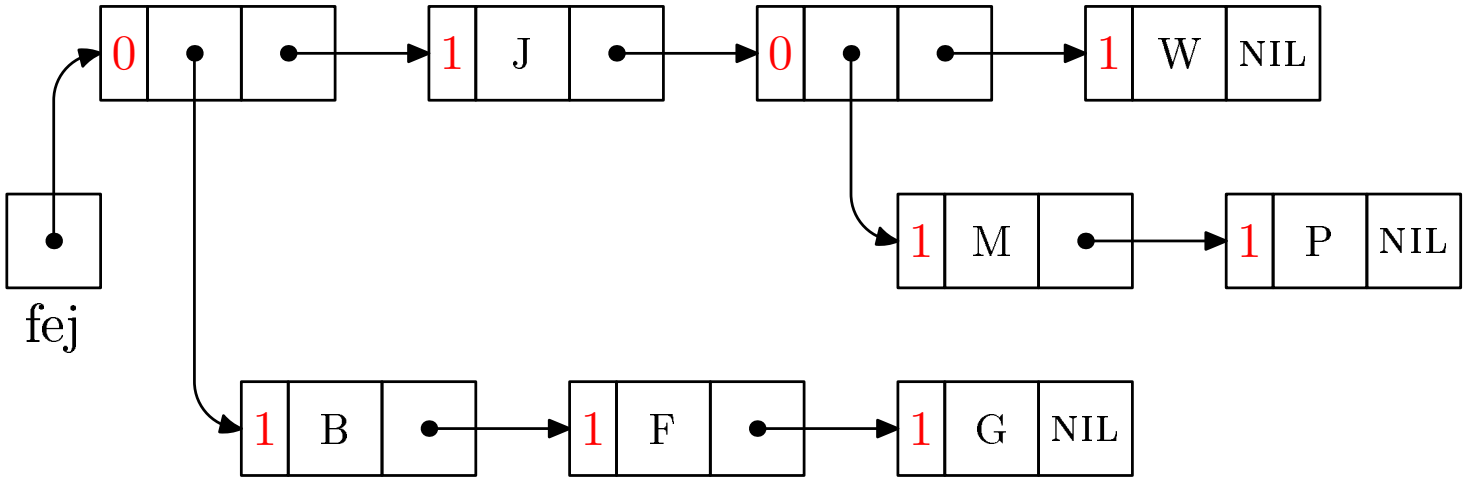
Ebben a változatban a listaelemek adatrésze általában összetett. Az adatrész valamely komponensének értékeit figyelembe véve építjük föl az egyirányban láncolt listákat.



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

Multilista (3)

Ebben a változatban a listaelemek adatrészében tárolt információt vagy feldolgozandó értékként, vagy mutatóként értelmezzük. Hogy pontosan miként, azt az adatrészben egy bit jelzi. A mutatóként értelmezett információ egy egyirányban láncolt lista fejmutatójának tekinthető.



Az ábrán ha a bit értéke 0, akkor az adatrészben tárolt információt mutatónak, ha 1, akkor feldolgozandó értéknek tekintjük.

Kósa Márk
Pánovics János