OBJEKTUM ORIENTÁLT PROGRAMOZÁS

Python nyelven

Láncolt listák létrehozása. Láncolt listák bejárása. Keresés, beszúrás, törlés, hulladékgyűjtés. Fejelt listák. Kétirányú listák

LISTÁK

Láncolt lista

- Olyan adatszerkezet, amelyben az elemek lineáris sorrendben követik egymást
- A lineáris sorrendet mutatók (C#, Python referenciák) segítségével valósíthatjuk meg.

Típusai:

- Szerkezet szerint
  - Egyszeresen láncolt
  - Kétszeresen láncolt
  - Ciklikus
- Szervezés szerint
  - Rendezett
  - Nem rendezett

Lista (sorozat)

- Elemtípusa bármi, akár lista is
- $[a_1, a_2,\ldots a_{i-1}][a_i, a_{i+1},\ldots a_n]$
- Fejrész, farok-rész

Műveletek

- Létesítés

$[][] \Rightarrow [a_1, a_2,\ldots a_{i-1}][a_i, a_{i+1},\ldots a_n]$

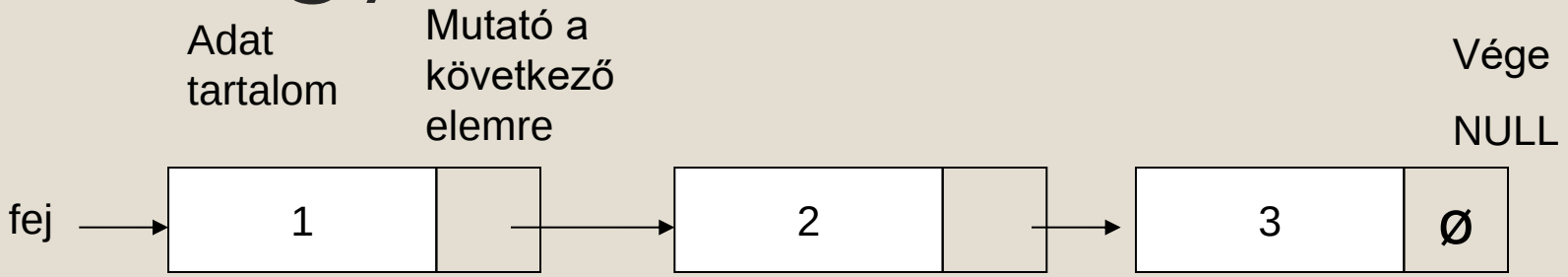
- Megszűntet

$[a_1, a_2,\ldots a_{i-1}][a_i, a_{i+1},\ldots a_n] \Rightarrow [][]$

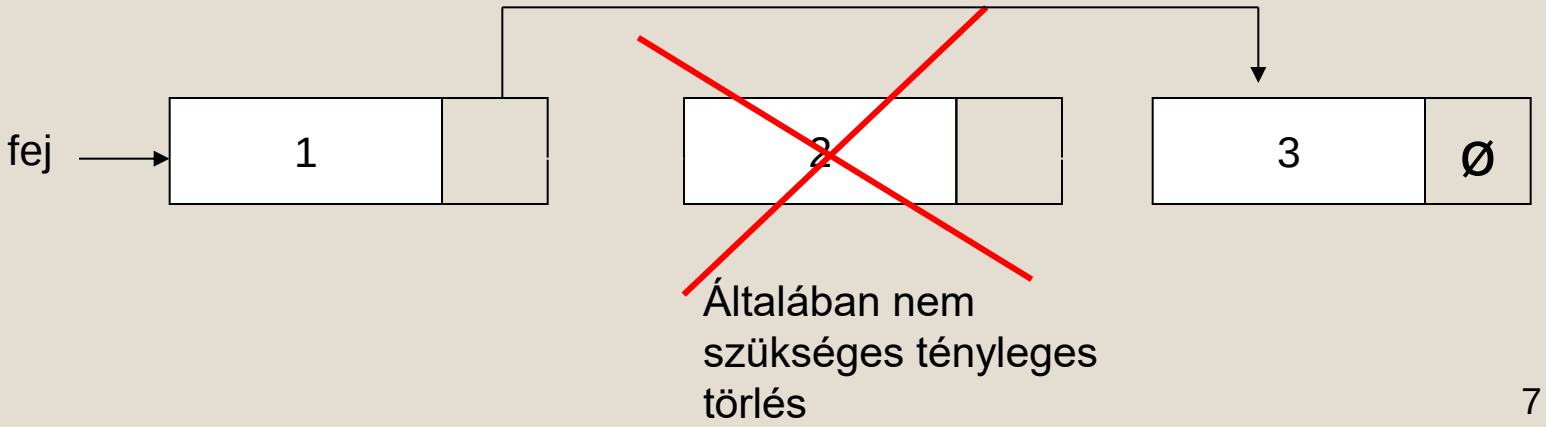
Műveletek

- Létesítés
- Megszüntet
- Üres
- Üres-e
- Elejére
- Végére
- Tovább
- Kiolvas
- Módosít
- Bővít
- töröl
- kapcsol

Egyszerű láncolt lista

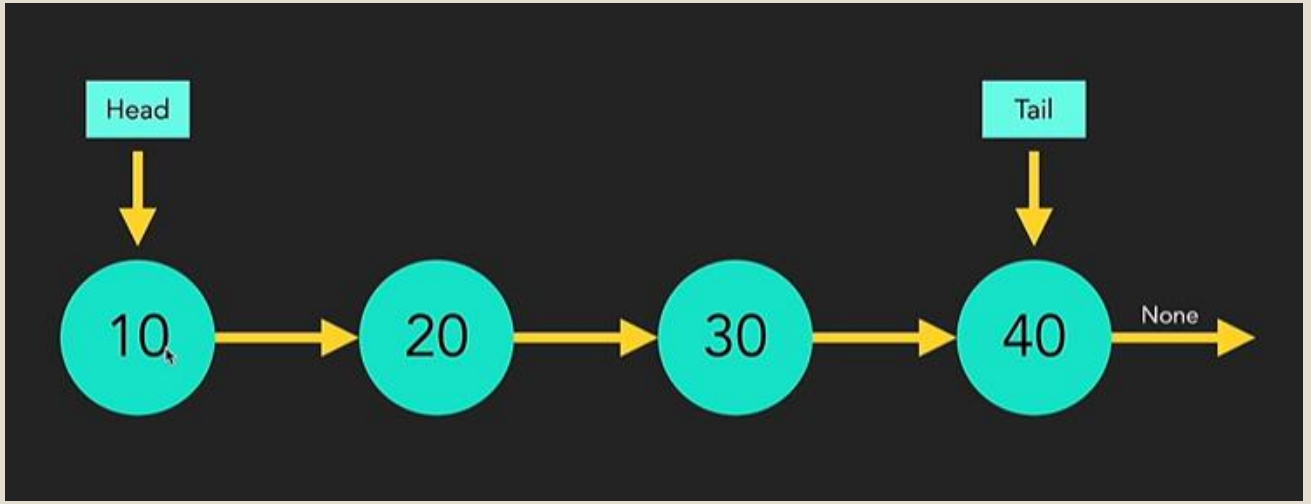


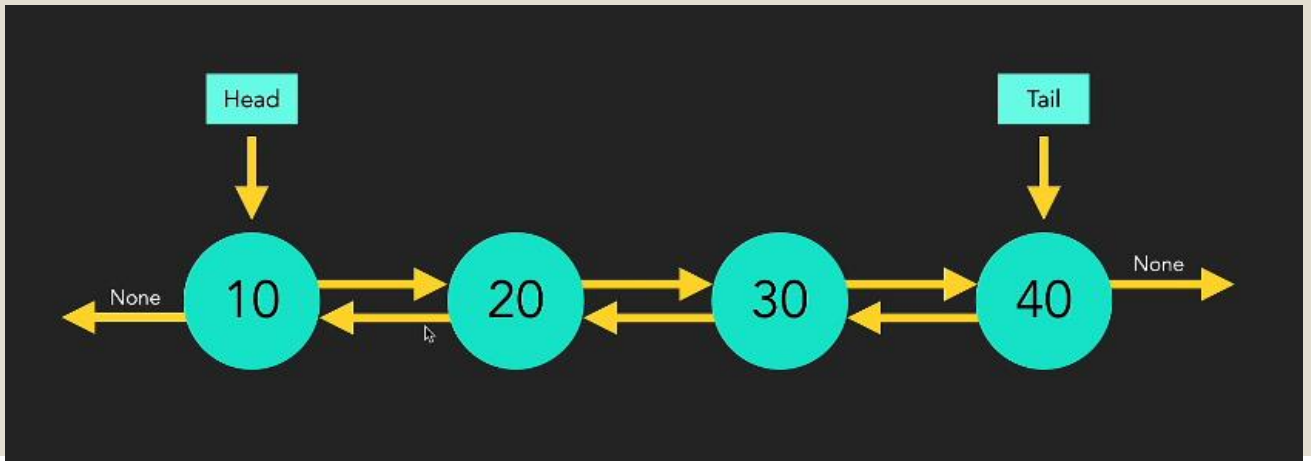
Törlés a listából



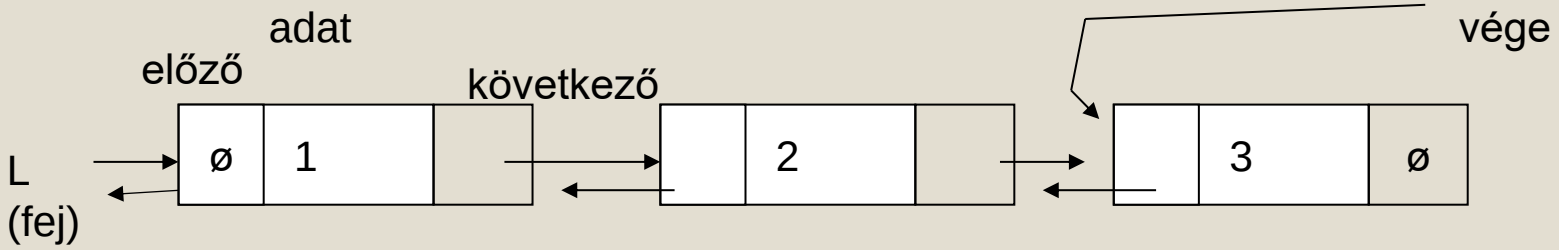
PYTHON IMPLEMENTÁCIÓ

Egyszeresen és kétszeresen láncolt lista összehasonlítása





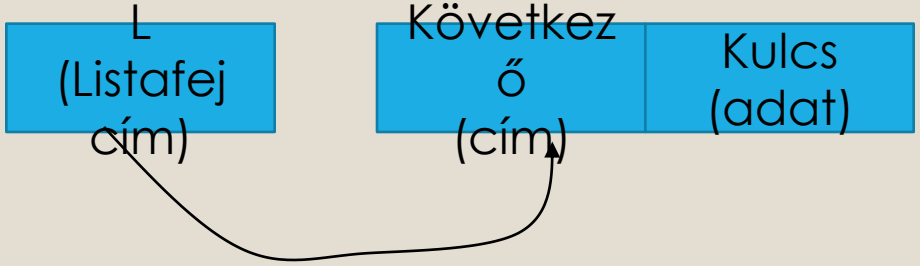
Kétirányú láncolt lista



- A lánc minden eleme két hivatkozást tartalmaz, az előző és az őt követő elemre
- A lánc végét az utolsó elem következő részének kitüntetett értéke jelzi
- A lánc elejét az első elem előző részének kitűntetett értéke jelzi
- Az első elem címét a listafej tartalmazza
- Az utolsó elem címét érdemes tárolni (vége)
- Előny az egyirányú listához képest: kevesebb műveletigény a keresés, törlés, beszúrásnál
- Hátrány: helyfoglalás, bonyolultabb algoritmus

Keresés listában (az első k kulcsú elem keresése)

```
Listában-keres(L,k)  //L lista fej referenciája
   X=L   //L a lista feje
            Ciklus amíg x<>null és x.kulcs<>k
                     x=x.következő
            Ciklus vége
Return x
```



Beszúrás lista elejére

```
Listába-beszúr(L,x)
        K=Elemfoglal(x)
        K.következő=(fej)L
        Ha (fej)L<>null
                        L.előző=K
        (fej)L=K
        K.előző=null
```

Törlés láncolt listából

Listából-töröl(L,x) //x referencia

Ha x.előző<>null // →

Akkor (x.előző).következő=x.következő

Különben L=x.következő

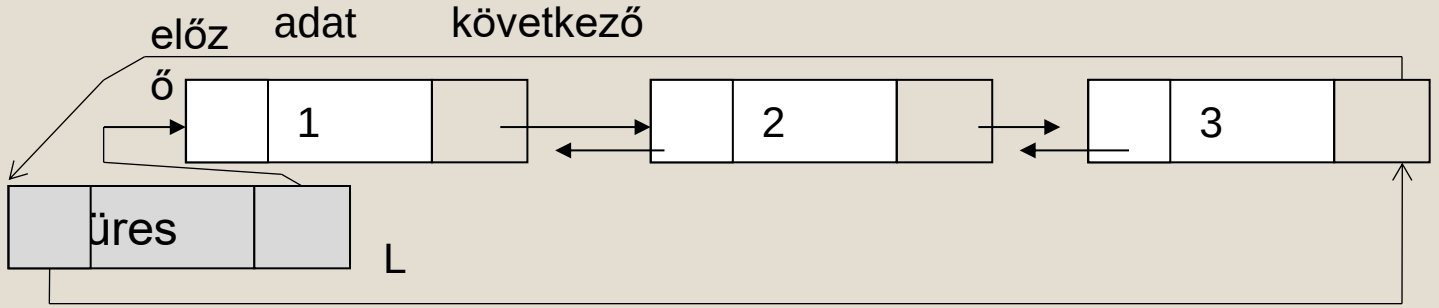
Ha x.következő<>null // ←

Akkor (x.következő).előző=x.előző

Strátzsa (őrszem, sentinel) technika

- Strázsa elemek: a lista elejére és/vagy a végére felvett kiegészítő elemek. Értékes adatot nem tárolnak, csak technikai szerepük van. Kiküszöbölik az első, utolsó elemek problémáját
- Előnyök: egyszerűbb, gyorsabb beszúrás és törlés
- Hátrány: helyfoglalás, bejárás körülményesebb

Ciklikus kétirányú strázsás lista



Törlés(L,x)

(X.előző).következő=x.következő

(X.következő).előző=x.előző

Őrszemes Listában-keres

```
Listában-keres(L,k)
        x=őrszem.következő
        ciklus amíg x!=őrszem és x.kulcs!=k
                x=x.következő
        ciklus vége
        return x
```

Klasszikus feladatok listákra

- Polinomok műveletei
- Feladat: Két polinom összeadása
  - Adjunk össze két x,y,z változójú polinomot!
  - $Ex^Ay^Bz^C$ polinom a kitevők szerint csökkenően rendezettek

Feladat

- A lista adatszerkezet az egyik alappillér a fa és gráf adatszerkezetek megértéséhez, ezért fontos, hogy értsük a különböző listaműveleteket.
- Feladat listaműveletek megvalósítása egyirányú és vagy kétirányú vagy őrszemes listával. 6 pont

KÖSZÖNÖM A FIGYELMET!