# Adatstruktúrák és Algoritmusok

6. gyakorlat

# Dinamikus halmazok: tömb és láncolt lista implementáció

Átnézzük a szükséges fogalmakat.

# Dinamikus halmaz

A **dinamikus halmaz** egy olyan halmaz, amely az őt felhasználó algoritmus során változik (bővül, szűkül, módosul).

# A sorozat adatstruktúra

A **sorozat adatstruktúra** objektumok olyan tárolási módja, amikor az elemek a műveletek által kijelölt lineáris sorrendben követik egymást.
**Tipikus műveletek:** keresés, beszúrás, törlés.
Sorozat implementálása

- tömb,
- láncolt lista,
  - egyszeresen vagy kétszeresen láncolt,
  - szentineles.

# Speciális sorozat adatstruktúrák: verem (STACK), sor (QUEUE)

# Verem (STACK)

A **verem** (STACK) olyan dinamikus halmaz, amelyben előre meghatározott az az elem melyet a TÖRÖL eljárással eltávolítunk. Ez az elem mindig az időben legutoljára a struktúrába helyezett elem lesz. (LIFO last in first out adat szerkezet). Műveletei: beszúrás: PUSH, törlés POP.

# Sor (QUEUE)

A **sor** (QUEUE) olyan dinamikus halmaz, amelyben előre meghatározott az az elem, melyet TÖRÖL eljárással eltávolítunk, és az az elem is, amelyet a BESZÚR eljárással a halmazba teszünk. Törlésre mindig az elemek közül a legrégebben beszúrt kerül. A beszúrt elem lesz a legfrissebb. (FIFO szerkezet). Műveletei: beszúrás, törlés.

# A tömb adatstruktúra

# A tömb attribútumai

**Definition**

A tömb azonos felépítésű (típusú) egymást fizikailag követő memóriarekeszeket jelent. Egy rekeszben egy elemet, adatrekordot helyezünk el. Az egyes tömbelemek helyét az indexük határozza meg. Az elemek fontos része a kulcsmező, melyet kulcs[Ax] révén kérdezhetünk le az A tömb x indexű eleme esetén. A tömböt szokás vektornak is nevezni.

Legyen $A$ egy tömb. Az attribútumok:

- fej $[A]$ = az első elem indexe,
- vége $[A]$ = az utolsó elem indexe,
- hossz $[A]$ = a tömbelemek száma (a tárolt elemek száma),
- tömbméret $[A]$ = annak a memóriaterületnek a nagysága tömbelem egységben mérve, ahová a tömböt elhelyeztük.

# A Beszúrás tömbbe algoritmus

**Beszúrás tömbbe** algoritmus: BESZ($A, x, r$, hibajelzés)
INPUT: $A$ a tömb, $x$ a tömb indexe, amely elem elé történik a beszúrás, $r$ a beszúrandó elem.
OUTPUT: hibajelzés - a beszúrás eredményességét jelzi.

| | BESZ($A, x, r$, hibajelzés) |
|---|---|
| 1. | IF hossz $[A] \neq 0$ |
| 2. | THEN IF (fej $[A] \leqslant x \leqslant$ vége $[A]$) és (tömbméret $[A] >$ vége $[A]$) |
| 3. | THEN FOR $i \longleftarrow$ vége $[A]$ DOWNTO $x$ DO |
| 4. | $A[i+1] \longleftarrow A[i]$ |
| 5. | $A[x] \longleftarrow r, \mathrm{INC}(\text{hossz}[A]), \mathrm{INC}(\text{vége}[A])$ |
| 6. | hibajelzés $\longleftarrow$ "sikeres beszúrás" |
| 6. | ELSE hibajelzés $\longleftarrow$ "sikertelen beszúrás" |
| 7. | ELSE fej $[A] \longleftarrow 1$, vége $[A] \longleftarrow 1$ hossz $[A] \longleftarrow 1$, $A[1] \longleftarrow r$ |
| 8. | RETURN(hibajelzés) |

# Törlés tömbből algoritmus

TÖRL($A, x$, hibajelzés)
INPUT: $A$ a tömb, $x$ a törlendő elem indexe.
OUTPUT: hibajelzés - a törlés eredményességét jelzi.

| | TÖRL($A, x$, hibajelzés) |
|---|---|
| 1. | IF hossz $[A] \neq 0$ |
| 2. | THEN IF fej $[A] \leqslant x \leqslant$ vége $[A]$ |
| 3. | THEN FOR $i \longleftarrow x$ TO vége $[A] - 1$ DO |
| 4. | $A[i] \longleftarrow A[i+1]$ |
| 5. | $\mathrm{DEC}(\text{hossz}[A]), \mathrm{DEC}(\text{vége}[A])$ |
| 6. | hibajelzés $\longleftarrow$ "sikeres törlés" |
| 7. | ELSE hibajelzés $\longleftarrow$ "nem létező elem" |
| 8. | ELSE hibajelzés $\longleftarrow$ "üres tömb" |
| 9. | RETURN(hibajelzés) |

# Példa

TÖMB $A = [5, 9, 10, 23, 92, ., ., .]$
fej$[A] = 1$, vége$[A] = 5$, hossz$[A] = 5$, tömbméret$[A] = 8$.
Szúrjuk be az $x = 3$ helyre az $r = 8$-at. (BESZÚRÁS, $T(n) = \Theta(n)$)
**1. lépés**: van-e hely?
**2. lépés**: ciklus $i = 5 \ldots 3$ $A[i+1] \longleftarrow A[i]$
$A[6] \longleftarrow A[5]$ $A[5, 9, 10, 23, 92, 92, ., .]$
$A[5] \longleftarrow A[4]$ $A[5, 9, 10, 23, 23, 92, ., .]$
$A[4] \longleftarrow A[3]$ $A[5, 9, 10, 10, 23, 92, ., .]$
**3. lépés**: $A[3] \longleftarrow x$ $A = [5, 9, 8, 10, 23, 92, ., .]$
INC(vége$[A]$): vége$[A] = 6$.
INC(hossz$[A]$): hossz$[A] = 6$.

# Lineáris keresés tömbben

**Lineáris keresés tömbben** algoritmus: LINKER($A, k, x$)
INPUT: $A$ a tömb, $k$ a keresett kulcs.
OUTPUT: $x$ a $k$ kulcsú elem indexe, vagy $\mathrm{NIL}$, ha nincs ilyen elem..

| | LINKER($A, k, x$) |
|---|---|
| 1. | IF Hossz[A]$\neq 0$ |
| 2. | THEN $x \longleftarrow$ fej $[A]$ |
| 3. | WHILE ($x \leqslant$ vége $[A]$) és $A[x] \neq k$) DO |
| 4. | $\mathrm{INC}(x)$ |
| 4. | IF ($x >$ vége $[A]$ |
| 5. | THEN $x \longleftarrow \mathrm{NIL}$ |
| 6. | ELSE $x \longleftarrow \mathrm{NIL}$ |
| 7. | RETURN($x$) |

# Bináris keresés rekurzív változata

BINKER($A, i, j, k, x$)
INPUT: $A$ a rendezett!!! tömb, $i$ a keresés kezdőindexe, $j$ a keresés végindexe ($i$ és $j$ nem vesznek részt a keresésben).
OUTPUT: $x$, ahol $x$ a $k$ kulcsú elem indexe, ha $A[x] = k$, vagy $\mathrm{NIL}$, ha $k$ kulcsú elem nincs a tömbben.

| | BINKER($A, i, j, k, x$) |
|---|---|
| 1. | $l \longleftarrow \lfloor \frac{i+j}{2} \rfloor$ |
| 2. | IF $i = l$ THEN $x \longleftarrow \mathrm{NIL}$, RETURN($x$) |
| 3. | ELSE IF $A[l] = k$ |
| 4. | THEN $x \longleftarrow l$, RETURN($x$) |
| 5. | ELSE IF $A[l] < k$ |
| 6. | THEN BINREK($A, l, j, k, x$) |
| 7. | ELSE BINREK($A, i, l, k, x$) |

# Példa

TÖMB $A = [5, 9, 10, 23, 92, ., ., .]$
fej$[A] = 1$, vége$[A] = 5$, hossz$[A] = 5$, tömbméret$[A] = 8$.
Keressük a $k = 23$ kulcsú elemet a tömbben.
$\mathrm{Bin}(A, 0, 6, k = 23, x)$
$I \longleftarrow \left\lfloor \frac{0+6}{2} \right\rfloor = 3$
$i \stackrel{?}{=} I \qquad (0 \stackrel{?}{=} 3) \qquad$ hamis
$A[3] \stackrel{?}{=} 23 \qquad$ hamis
$A[3] < 23$ igaz $\Rightarrow \mathrm{Bin}(A, 3, 6, 23, x)$
$I \longleftarrow \left\lfloor \frac{3+6}{2} \right\rfloor = 4.$
$i \stackrel{?}{=} I \qquad (3 \stackrel{?}{=} 4) \qquad$ hamis
$A[4] \stackrel{?}{=} 23$ igaz $\Rightarrow$ RETURN(4).

# Bináris keresés iteratív változata

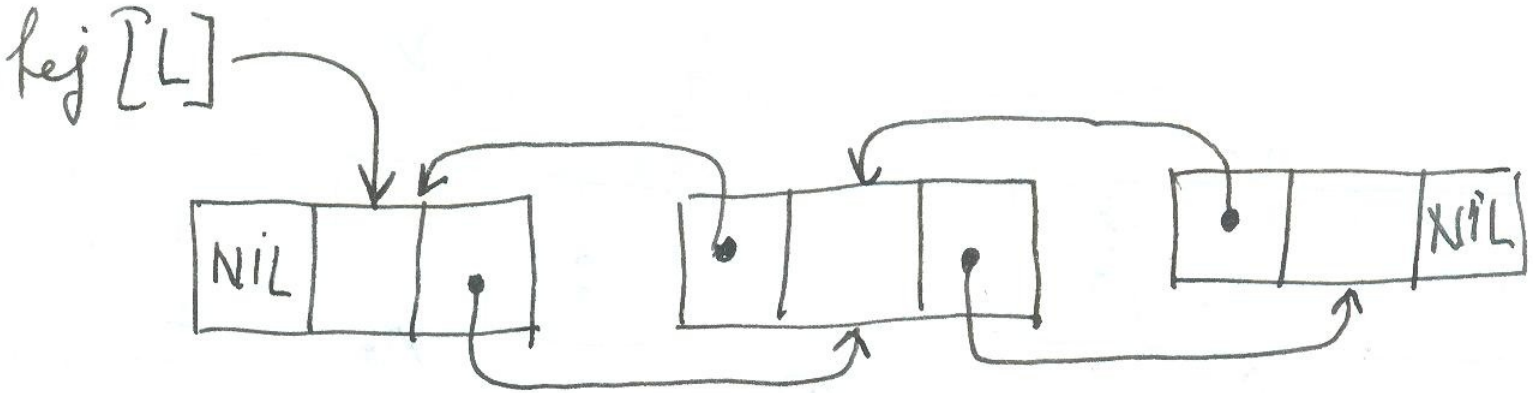
BINIT$(A, i, j, k, x)$
INPUT: $A$ a tömb, $i$ a keresés kezdőindexe, $j$ a keresés végindexe ($i$ és $j$ nem vesznek részt a keresésben).
OUTPUT: $x$, ahol $x = k$ kulcsú elem indexe, ha $k$-t megtaláltuk, illetve $\mathrm{NIL}$, ha $k$ nincs benne a tömbben.

| | BINIT$(A, i, j, k, x)$ |
|---|---|
| 1. | $l \longleftarrow \lfloor \frac{i+j}{2} \rfloor$ |
| 2. | WHILE $i < l$ DO |
| 3. | IF $A[l] = k$, THEN $x \longleftarrow l$, RETURN$(x)$ |
| 4. | IF $A[l] < k$ THEN $i \longleftarrow l$ |
| 5. | ELSE $j \longleftarrow l$ |
| 6. | $l \longleftarrow \lfloor \frac{i+j}{2} \rfloor$ |
| 7. | $x \longleftarrow \mathrm{NIL}$ |
| 8. | RETURN$(x)$ |

# A láncolt lista

# Kétirányban láncolt lista


Az egyirányban láncolt lista hasonló, de hiányzik belőle az előre mutató mutató, tehát csak egy irányban tudunk lépkedni az elemeken.
Először foglalkozunk a láncolt listával, de nem írjuk le részletesen a műveleteket (beszúrás, keresés, törlés), ezeket a műveleteket csak a szentineles láncolt lista esetén ismertetjük.
Jelölje $L =$ a láncolt listát. Egyetlen **attribútuma** van a fej$[L] =$, ami a legelső listaelemre mutató mutató.

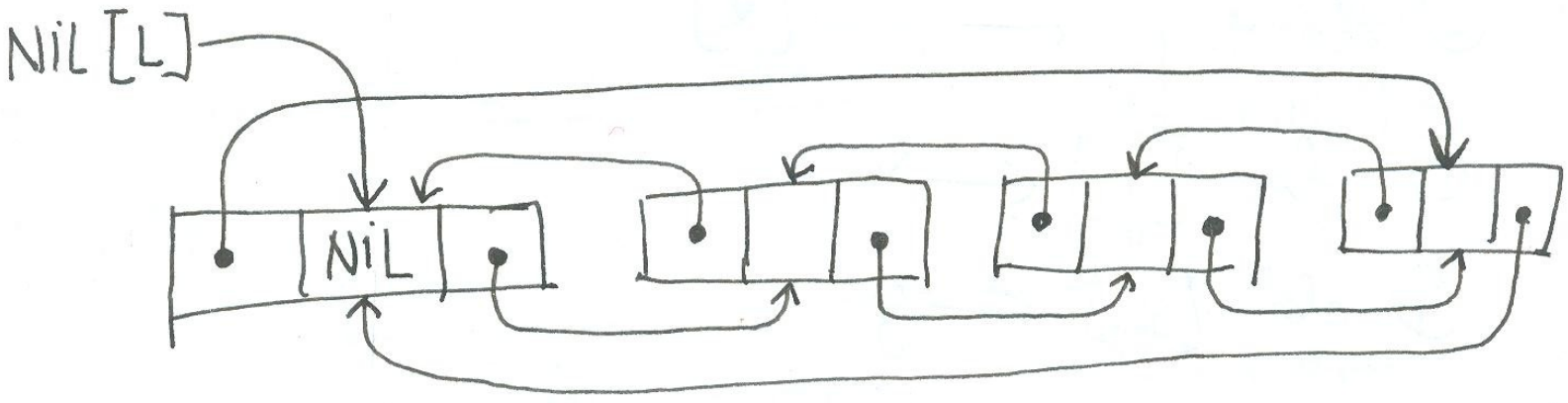



Tehát:

- fej[$L$]: az első listaelemre mutat (vagy $\mathrm{NIL}$, ha a lista üres).
- elő: az előző listaelemre mutat, illetve a lista első eleme esetén $\mathrm{NIL}$
- köv: a következő listaelemre mutat, illetve az utolsó listaelem esetén $\mathrm{NIL}$

# A szentineles láncolt lista

**Szentinel:** strázsa vagy őrszem, amely zárt láncolt listává teszi a listát.




A szentineles láncolt listának egyetlen attribútuma van a $\mathrm{NIL}(L)$, ami a szentinelre mutató mutató. A szentinelnek $\mathrm{NIL}$ a kulcsa. A szentinel is rendelkezik elő, illetve köv mutatókkal, amelyek jelentése:

- elő $[\mathrm{NIL}[L]]$ az utolsó elemre mutat, illetve $\mathrm{NIL}$, ha a lista üres;
- köv $[\mathrm{NIL}[L]]$ az első elemre mutat, illetve $\mathrm{NIL}$, ha a lista üres.

# Műveletek szentineles láncolt listában

1. Keresés;
2. Beszúrás első elemként. (Külön vízsgáljuk azokat az eseteket, amikor a lista üres, illetve nem üres);
3. Beszúrás utolsó elemként (**Szorgalmi házi feladat**);
4. Beszúrás adott mutatójú elem elé;
5. Beszúrás adott mutatójú elem után (Szorgalmi házi feladat);
6. Adott mutatójú elem törlése.

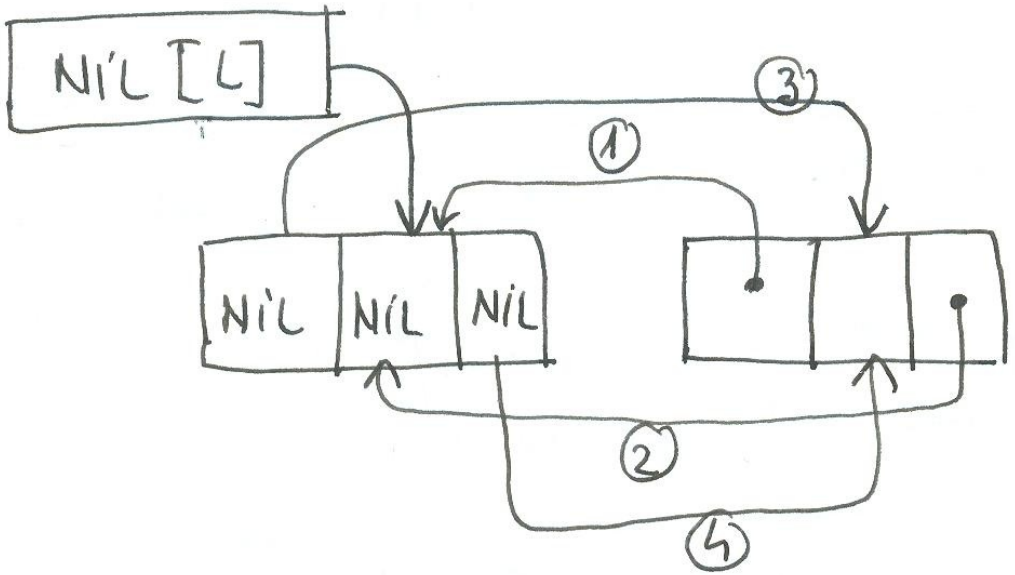
# 1. Keresés

KERES($L, k, x$)
INPUT: $L$ a lista, $k$ a keresett kulcs.
OUTPUT: $x$, ami a kulcs mutatója, ha a kulcs a listában van, vagy $\mathrm{NIL}$, ha nincs a listában.

| | KERES($L, k, x$) |
|---|---|
| 1. | $x \longleftarrow$ köv $[\mathrm{NIL}[L]]$ |
| 2. | WHILE ($x \neq \mathrm{NIL}[L]$ és kulcs$[x] \neq k$) DO |
| 3. | $\quad x \longleftarrow$ köv$[x]$ |
| 4. | IF $x = \mathrm{NIL}[L]$ THEN $x \longleftarrow \mathrm{NIL}$ |
| 5. | RETURN($x$) |

# 2. Beszúrás első elemként üres listába

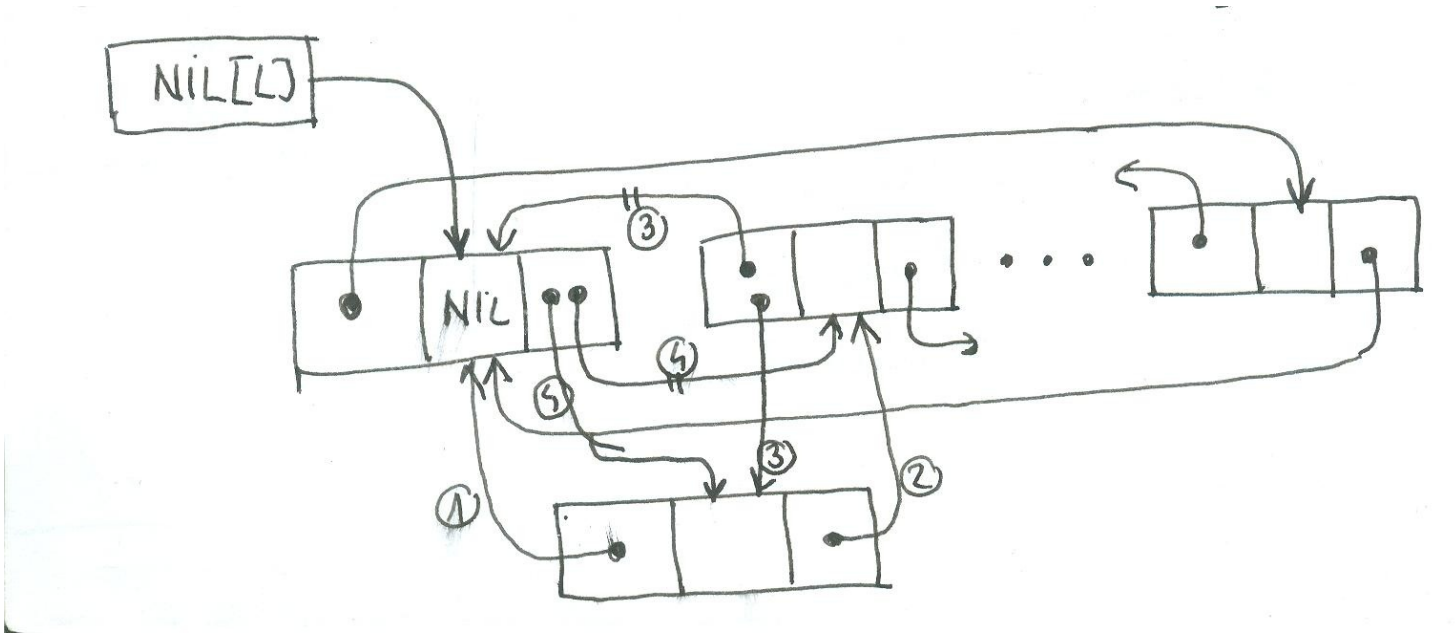
Az $x$ mutatójú elemet szúrjuk be.




1. elő$[x] \longleftarrow \mathrm{NIL}[L]$
2. köv$[x] \longleftarrow \mathrm{NIL}[L]$
3. elő $[\mathrm{NIL}[L]] \longleftarrow x$
4. köv $[\mathrm{NIL}[L]] \longleftarrow x$

# Beszúrás első elemként nemüres listába

Az $x$ mutatójú elemet szúrjuk be.




1. elő$[x] \longleftarrow \mathrm{NIL}[L]$
2. köv$[x] \longleftarrow$ köv $[\mathrm{NIL}[L]]$
3. elő [köv $[\mathrm{NIL}[L]]] \longleftarrow x$
4. köv $[\mathrm{NIL}[L]] \longleftarrow x$

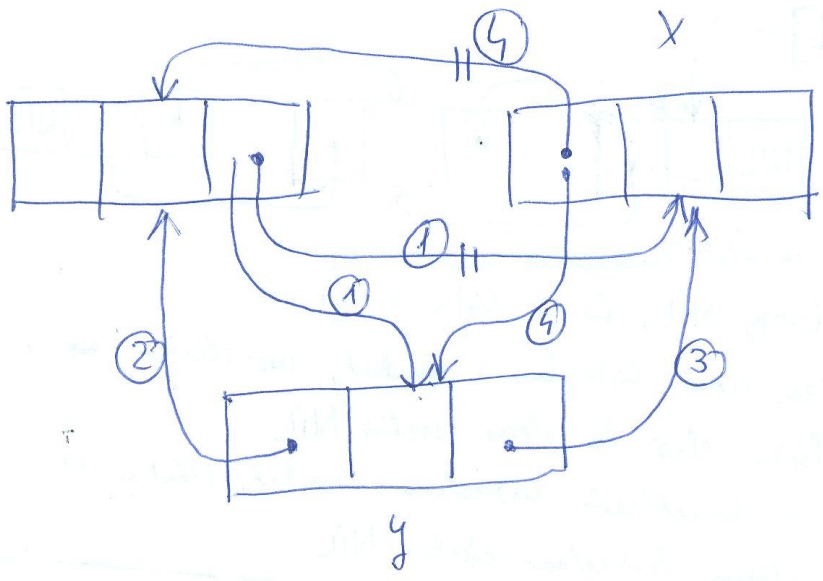
# Beszúrás első elemként

Mindkét fenti esetet magába foglaló pszeudokód:
BESZÚR($L, x$)
INPUT: $L$ a lista, $x$ a beszúrandó elem mutatója.

| | BESZÚR($L, x$) |
|---|---|
| 1. | IF (köv $[\mathrm{NIL}[L]] = \mathrm{NIL}$ |
| 2. | THEN elő$[x] \longleftarrow \mathrm{NIL}[L]$ |
| 3. | köv$[x] \longleftarrow \mathrm{NIL}[L]$ |
| 4. | elő $[\mathrm{NIL}[L]] \longleftarrow x$ |
| 5. | köv $[\mathrm{NIL}[L]] \longleftarrow x$ |
| 6. | ELSE elő$[x] \longleftarrow \mathrm{NIL}[L]$ |
| 7. | köv$[x] \longleftarrow$ köv $[\mathrm{NIL}[L]]$ |
| 8. | elő [köv $[\mathrm{NIL}[L]]] \longleftarrow x$ |
| 9. | köv $[\mathrm{NIL}[L]] \longleftarrow x$ |
| 10. | RETURN |

# 3. Beszúrás adott mutatójú elem elé

$x$ mutatójú elem elé szúrjuk be az $y$ mutatójú elemet.



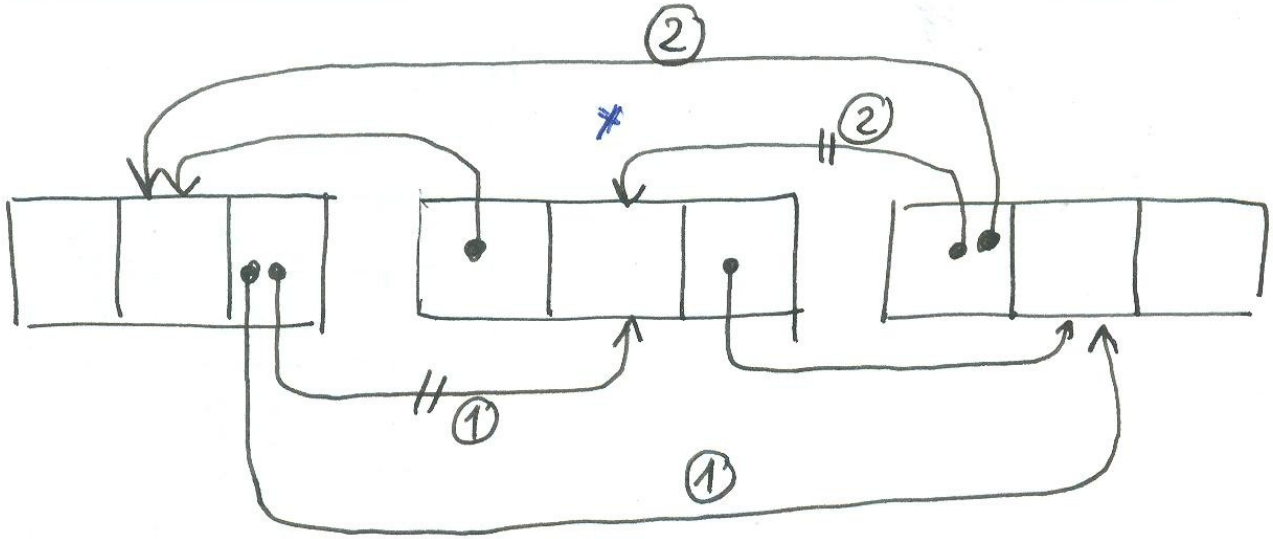

1. köv [elő[$x$]] $\longleftarrow$ $y$
2. elő[$y$] $\longleftarrow$ elő[$x$]
3. köv[$y$] $\longleftarrow$ $x$
4. elő[$x$] $\longleftarrow$ $y$

# 4. Beszúrás adott mutatójú elem után

Szorgalmi házi feladat. Könnyen visszavezethető az előző feladatra.

# 5. Adott mutatójú elem törlése

TÖRÖL($L, x$)




INPUT: $L$ a lista, $x$ a törlendő elemre mutató mutató. Szentinelt nem törlünk.

| | TÖRÖL($L, x$) |
|---|---|
| 1. | köv [elő[$x$]] $\longleftarrow$ köv[$x$] |
| 2. | elő [köv[$x$]] $\longleftarrow$ elő[$x$] |
| 3. | RETURN |

KÖSZÖNÖM A FIGYELMET!