EGYSZERŰ LÁNCOLT LISTA (SIMPLE ONE-WAY LIST; S1L)

Sorozat, pl.: $\langle 5, 3, 2, 7 \rangle$

láncolt

$L \to \boxed{5\,|\,\cdot} \to \boxed{3\,|\,\cdot} \to \boxed{2\,|\,\cdot} \to \boxed{7\,|\,\oslash}$

szekvenciális

$\boxed{5\,|\,3\,|\,2\,|\,7}$ (tömb.)

üres $L = \oslash$ $L \to \boxed{NO}$

$*p$

$p \to \boxed{\,|\,}$

$(*p).key$ $(*p).next$

$p \to key$ $p \to next$

HIBÁK:

$*\oslash$, $\oslash \to key$

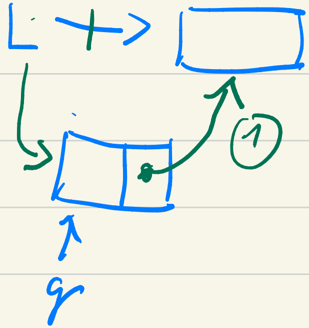
$\oslash \to next$

(SEGMENTATION VIOLATION)

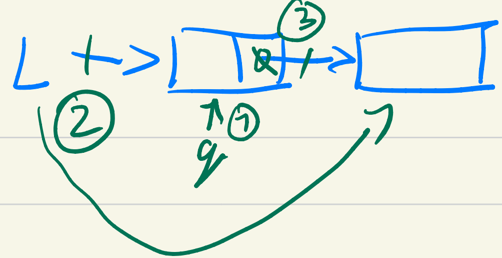
E1
$+key : \mathcal{T}$
... // satellite data may come here
$+next$: **E1***
+E1() { $next := \oslash$ }

// Insert $*q$ at the
// front of list L.
$q \to next := L$ 1
$L := q$ 2

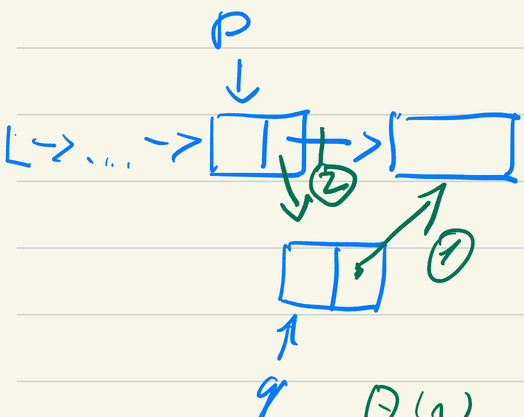
// Unlink the first element of list L.
$q := L$ 1
$L := q \to next$ 2
$[q \to next := \oslash]$ 3



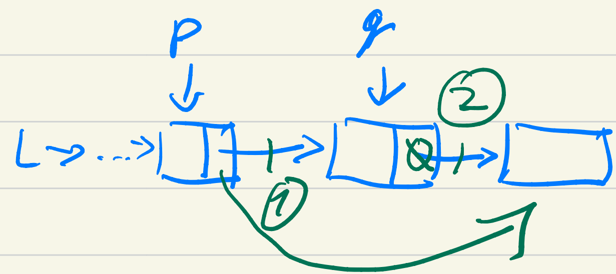
$\Theta(1)$ idő



$\Theta(1)$



$\Theta(1)$ idő



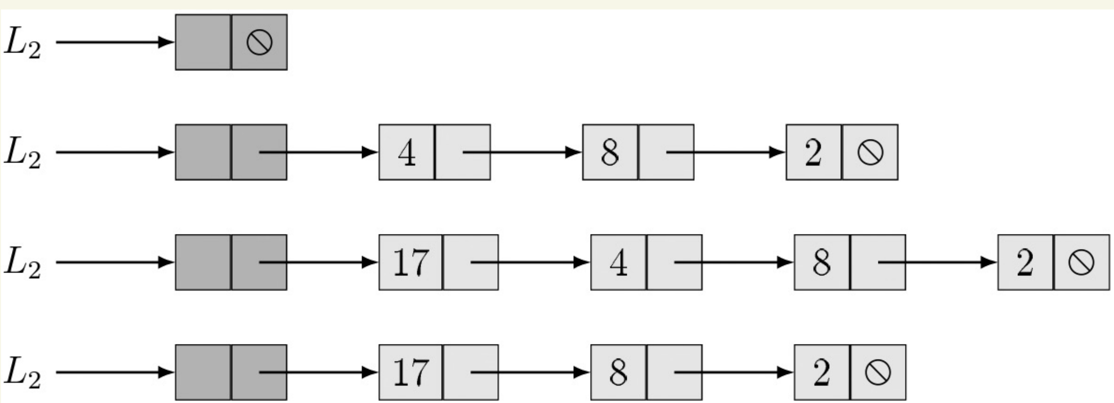
$\Theta(1)$ idő

// Let $*p$ be followed by $*q$.
$q \to next := p \to next$ 1
$p \to next := q$ 2

// Provided that $*p$ is followed
// by $*q$, unlink $*q$.
$p \to next := q \to next$ 1
$[q \to next := \oslash]$ 2

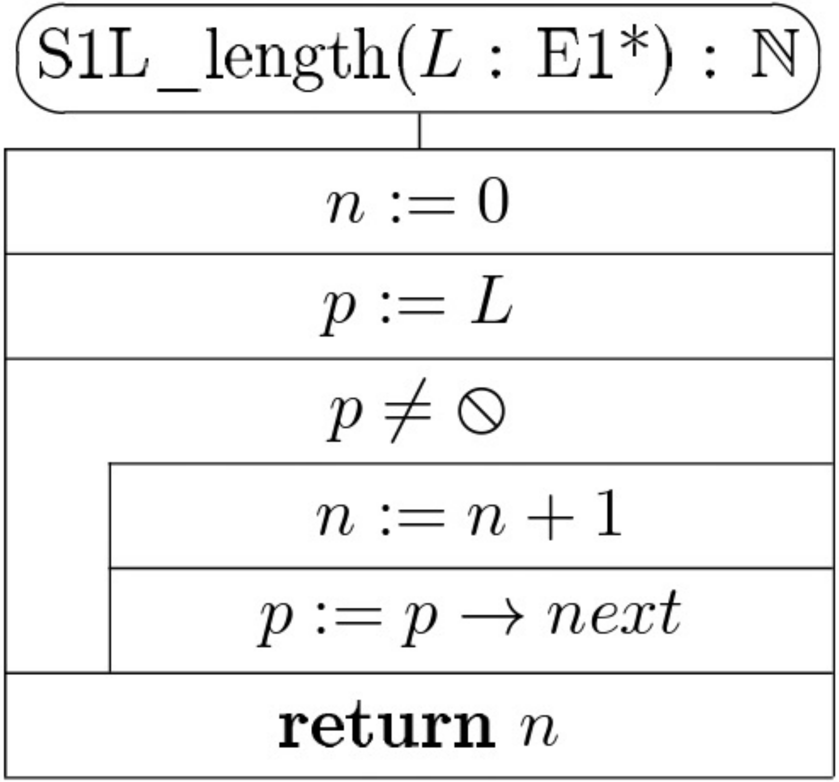
ŐRSZEMES (SENTINEL) LISTÁK (FEJELEM v. VÉGELEM a listán)

FEJELEMES LISTA



H1L

HEADER+ 1-way List



H1L_length(H : E1*) : ℕ

return S1L_length(H->next)

$T(n) \in \Theta(n)$

$S(n) \in \Theta(1)$

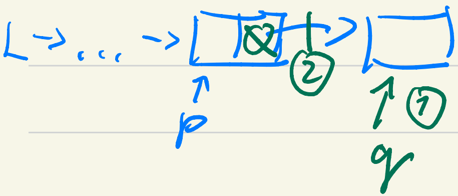
S1L

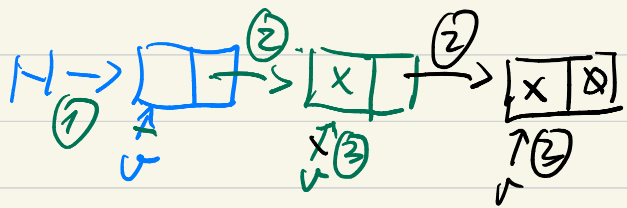
cut$(L : E1^* \; ; \; n : \mathbb{N}) : E1^*$

- $p := L$
- while $n > 1$:
  - $n := n - 1$
  - $p := p \to next$
- $q := p \to next$ 1
- $p \to next := \emptyset$ 2
- **return** q

H1L_read$() : E1^*$

- $H := v :=$ **new** E1 1
- while read(x):
  - $v :=$ 2 $v \to next :=$ 2 **new** E1
  - $v \to key := x$
  - // satellite data may be read here
- **return** H





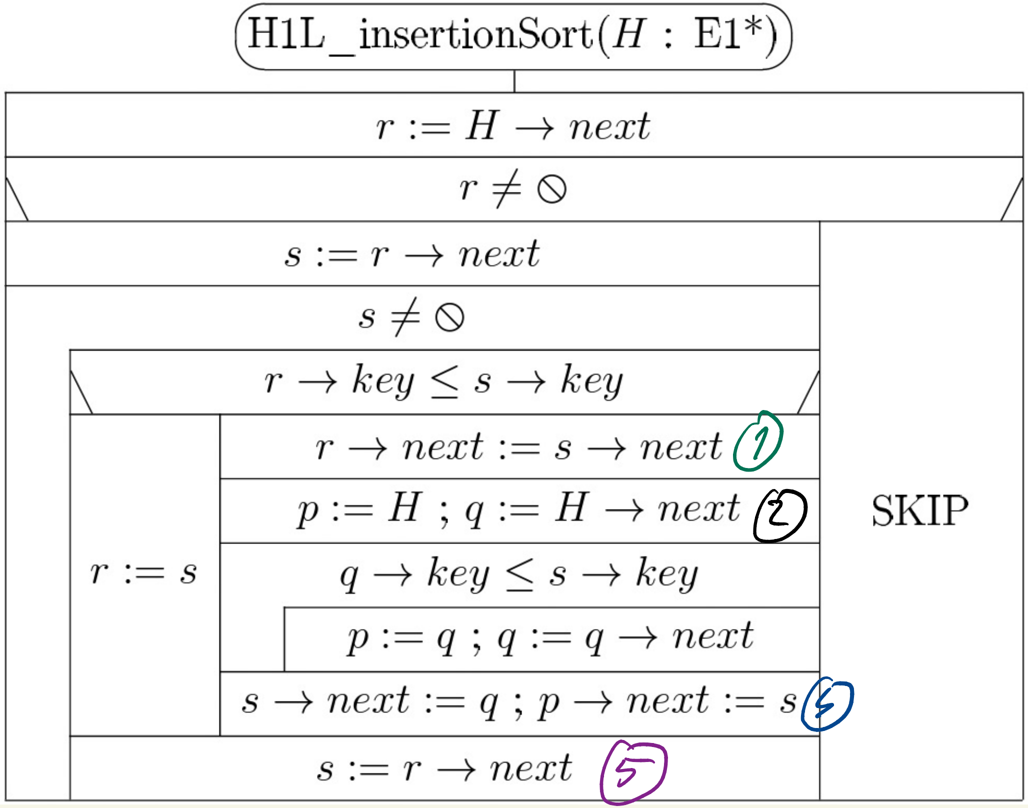
$T(n) \in \Theta(n)$

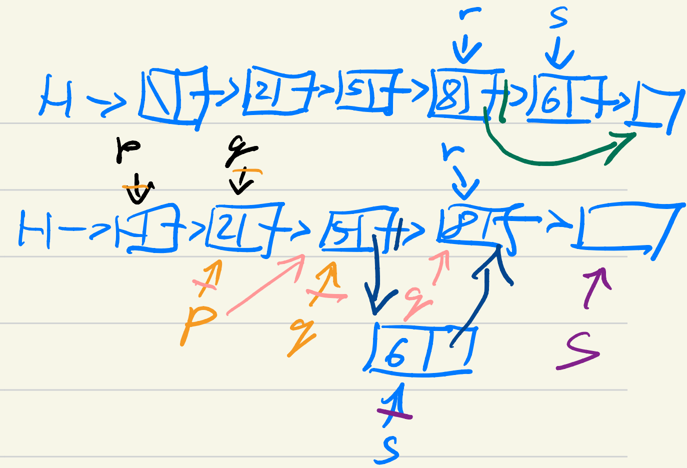
$S(n) \in \Theta(1)$

$T(n) \in \Theta(n)$

$S(n) \in \Theta(n)$

$S_{in}(n) \in \Theta(1)$





$mT(n) \in \Theta(n)$

$MT(n) \in \Theta(n^2)$

$AT(n) \in \Theta(n^2)$

$S(n) \in \Theta(1)$

S1L rend.

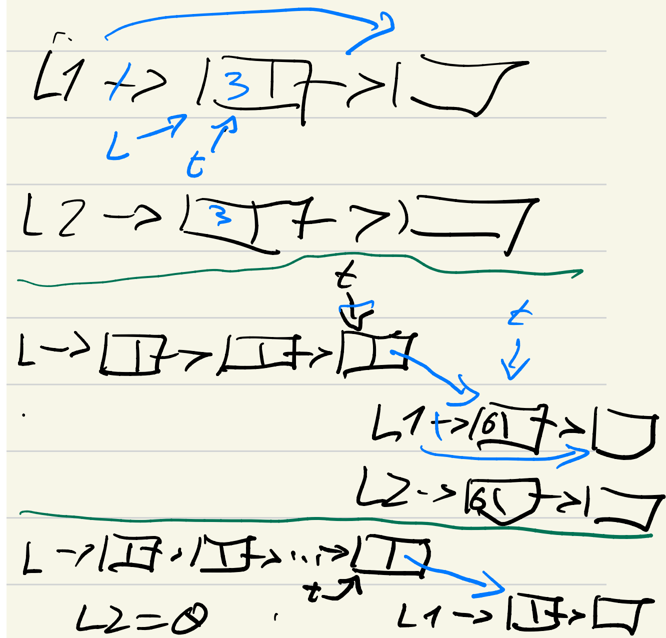
mergeSort(&L : E1*)

```
// L is an S1L.
n := S1L_length(L)
ms(L, n)
```

ms(&L : E1* ; n : N)

```
IF n > 1
    n1 := ⌊n/2⌋
    L2 := cut(L, n1)
    ms(L, n1)
    ms(L2, n − n1)
    L := merge(L, L2)
ELSE
    SKIP
```

$mT(n), MT(n) \in \Theta(n \cdot \log n)$



$S(n) \in \Theta(\log n)$

(majdnem helyben rendező)

merge(L1, L2 : E1*) : E1*

```
IF L1 → key ≤ L2 → key
    L := t := L1
    L1 := L1 → next
ELSE
    L := t := L2
    L2 := L2 → next
WHILE L1 ≠ ⦸ ∧ L2 ≠ ⦸
    IF L1 → key ≤ L2 → key
        t := t → next := L1
        L1 := L1 → next
    ELSE
        t := t → next := L2
        L2 := L2 → next
IF L1 ≠ ⦸
    t → next := L1
ELSE
    t → next := L2
return L
```

EGYIRÁNYÚ, VÉGELEMES LISTÁK

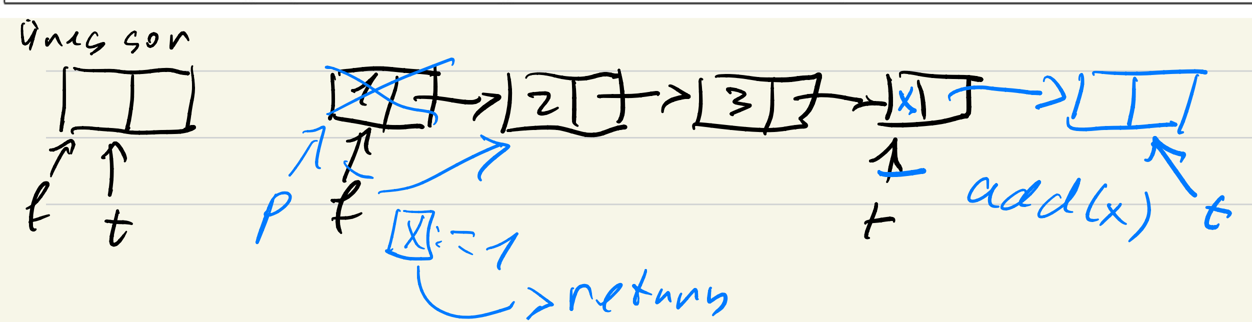
Queue

$-$ *first*, *trailer* : E1* // a one-way list with trailer represents the queue
$-$ $n : \mathbb{N}$ // n is the actual length of the queue

+ Queue(){ $first := trailer :=$ **new** E1 ; $n := 0$ } // create an empty queue
+ add($x : \mathcal{T}$) // join x to the end of the queue
+ rem() : $\mathcal{T}$ // remove and return the first element of the queue
+ first() : $\mathcal{T}$ // return the first element of the queue
+ length() : $\mathbb{N}$ {**return** n}
+ isEmpty() : $\mathbb{B}$ {**return** $n = 0$}
+ setEmpty() // reinitialize the queue
+ $\sim$ Queue() { setEmpty() ; **delete** *trailer* }

$T(n) \in \Theta(1)$

$T(n) \in \Theta(n)$



$AT_{add+setEmpty+rem}(n) \in \Theta(1)$

Queue::add($x : \mathcal{T}$)

$trailer \to key := x$
$trailer := trailer \to next :=$ **new** E1
$n{+}{+}$

Queue::rem() : $\mathcal{T}$

$n > 0$	
$x := first \to key$	QueueUnderflow
$n{-}{-}$	
$p := first$	
$first := first \to next$	
delete p	
return x	

Queue::first() : $\mathcal{T}$

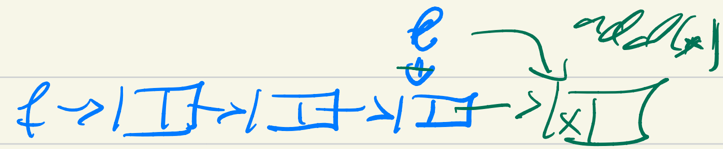
$n > 0$	
return $first \to key$	QueueUnderflow

Queue::setEmpty()

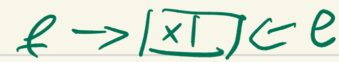
$first \neq trailer$
$p := first$
$first := first \to next$
delete p
$n := 0$

SOR ALTERNATÍV LÁNCOLT ÁBRÁZOLÁSAI:

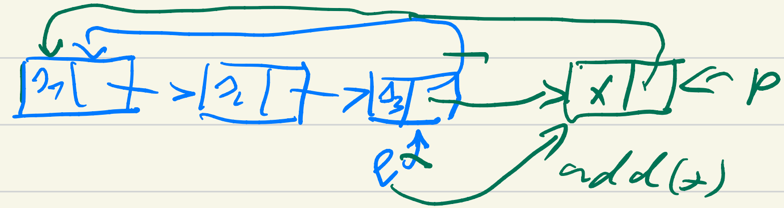
- S1L + last pointer



  üres sor: $f = 0$ add(x)

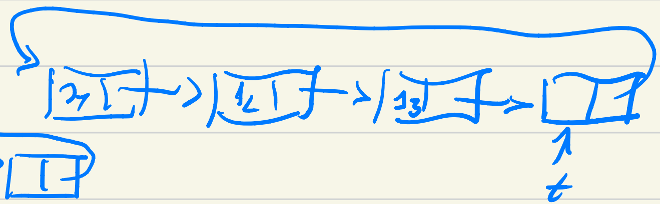


- C1L (trailer nélkül)

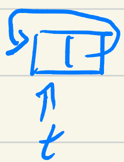


  üres sor: $l = 0$

- C1L

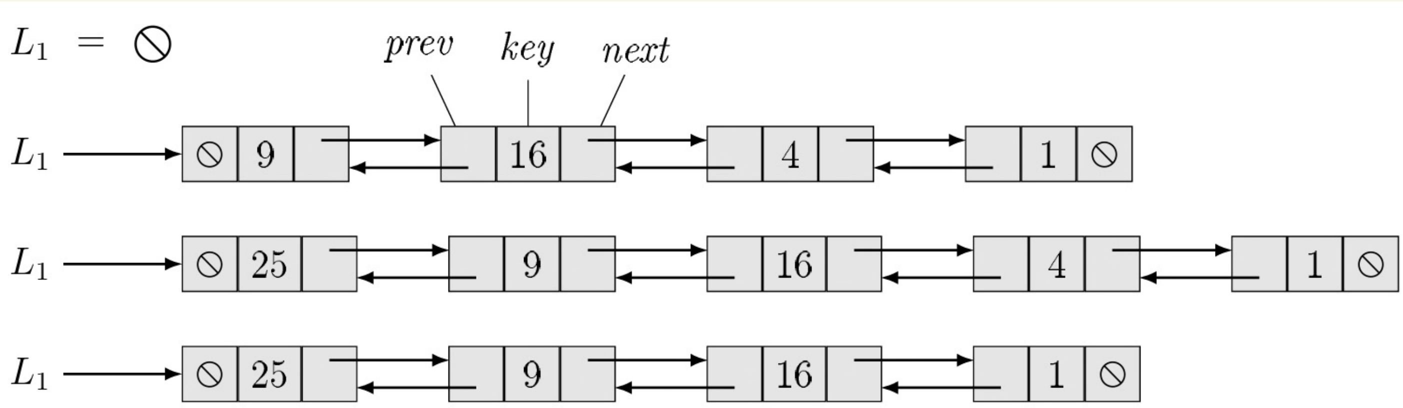


  üres sor:



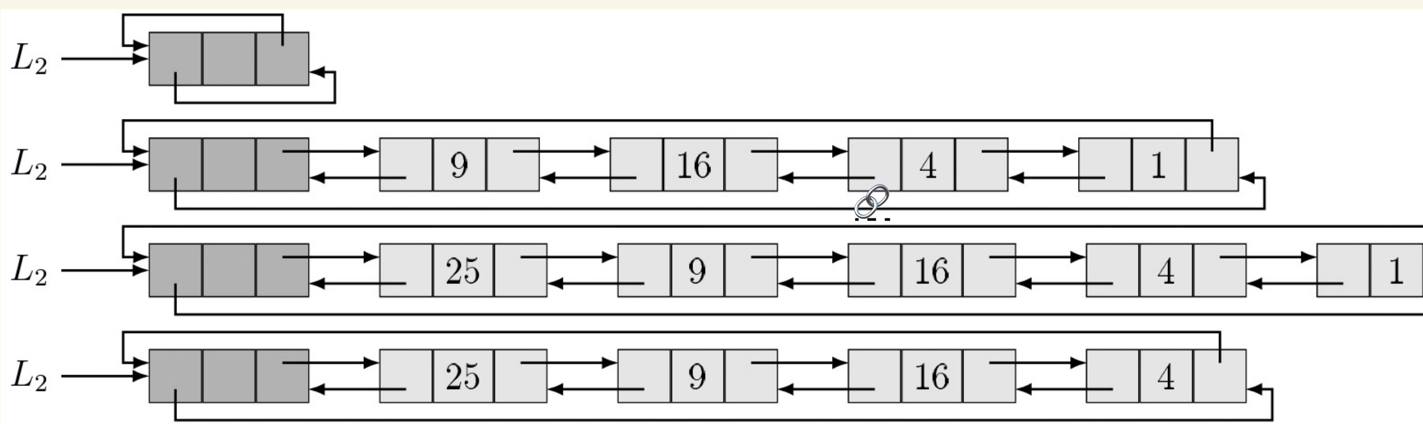
- H1L + last pointer (ld. H1L_read())

TWO-WAY LISTS



S2Ls

CYCLIC TWO-WAY LISTS (C2Ls)



C2Ls

E2
$+prev, next$: **E2*** // refer to the previous and next neighbour or be **this**
$+key : \mathcal{T}$
+ E2() { $prev := next :=$ **this** }

precede(q, r : E2*)

// $(*q)$ will precede $(*r)$
$p := r \rightarrow prev$
$q \rightarrow prev := p$; $q \rightarrow next := r$
$p \rightarrow next := r \rightarrow prev := q$

follow(p, q : E2*)

// $(*q)$ will follow $(*p)$
$r := p \rightarrow next$
$q \rightarrow prev := p$; $q \rightarrow next := r$
$p \rightarrow next := r \rightarrow prev := q$

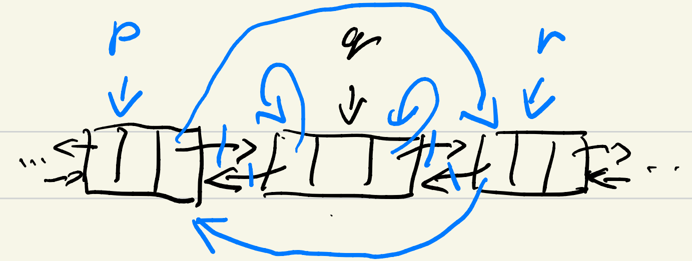
unlink(q : E2*)

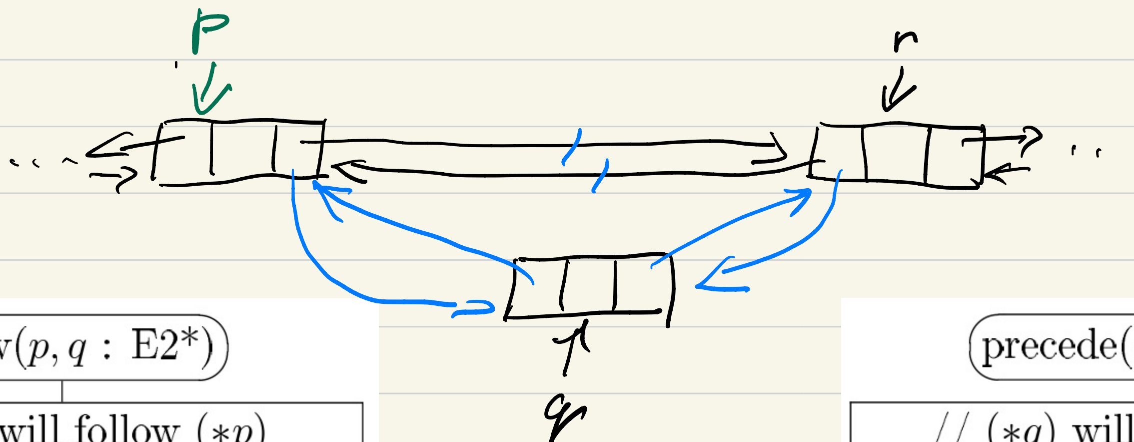
// remove $(*q)$
$p := q \rightarrow prev$; $r := q \rightarrow next$
$p \rightarrow next := r$; $r \rightarrow prev := p$
$q \rightarrow prev := q \rightarrow next := q$

$T(n) \in \Theta(1)$

unlink(q : E2*)

// remove $(*q)$
$p := q \rightarrow prev$; $r := q \rightarrow next$
$p \rightarrow next := r$; $r \rightarrow prev := p$
$q \rightarrow prev := q \rightarrow next := q$





follow(p, q : E2*)

// $(*q)$ will follow $(*p)$
$r := p \rightarrow next$
$q \rightarrow prev := p$; $q \rightarrow next := r$
$p \rightarrow next := r \rightarrow prev := q$

ins_between(p,q,r)

precede(q, r : E2*)

// $(*q)$ will precede $(*r)$
$p := r \rightarrow prev$
$q \rightarrow prev := p$; $q \rightarrow next := r$
$p \rightarrow next := r \rightarrow prev := q$

$H \to [/]—[5]—[2]—[7]—[/] \leftarrow H$ a $\langle 5; 2; 7\rangle$ sorozat egy lehetséges ábrázolása,
$H \to [/]—[/] \leftarrow H$ a $\langle\rangle$ üres sorozaté.

C2L_read(&H : E2*)

$H :=$ **new** E2
read(x)
$p :=$ **new** E2
$p \to key := x$
precede(p, H)

setEmpty(H : E2*)

$p := H \to prev$
$p \neq H$
unlink(p)
delete p
$p := H \to prev$

$T(n)$
$\in \Theta(n)$

$H \to [/]—[/] \leftarrow H$
$H \to [/]—[5]—[/] \leftarrow H$
$H \to [/]—[5]—[2]—[/] \leftarrow H$
$H \to [/]—[5]—[2]—[7]—[/] \leftarrow H$

```
insertionSort(H : E2*)
r := H → next ; s := r → next
s ≠ H
    r → key ≤ s → key
    r := s  |  unlink(s)
            |  p := r → prev
            |  p ≠ H ∧ p → key > s → key
            |      p := p → prev
            |  follow(p, s)
    s := r → next
```

$H \to [/]—[5]—[2]—[7]—[2]—[/] \leftarrow H$

$H \to [/]—[2]—[5]—[7]—[2]—[/] \leftarrow H$

$H \to [/]—[2]—[5]—[7]—[2]—[/] \leftarrow H$

$H \to [/]—[2]—[2]—[5]—[7]—[/] \leftarrow H$

$mT(n) \in \Theta(n)$

$\left.\begin{array}{l} MT(n) \\ AT(n) \end{array}\right\} \in \Theta(n^2)$

```
length(H : E2*) : ℕ
n := 0
p := H → next
p ≠ H
    n := n + 1
    p := p → next
return n
```

$T(n) \in \Theta(n)$

$S(n) \in \Theta(1)$

ELŐRENDEZETT INPUTOKRA IS $\Theta(n)$ műveletigény

Szig. mon. növ. rend. listák

unionIntersection($H_u, H_i : E2*$) C2L-ek

$q := H_u \to next$; $r := H_i \to next$

while $q \neq H_u \wedge r \neq H_i$:

$q \to key < r \to key$	$q \to key > r \to key$	$q \to key = r \to key$
$q := q \to next$	$p := r$	$q := q \to next$
	$r := r \to next$	$r := r \to next$
	unlink(p)	
	precede(p, q)	

while $r \neq H_i$:

$p := r$; $r := r \to next$; unlink(p)

precede(p, H_u)

$n_u := |H_u|$

$n_i := |H_i|$

$MT(n_u, n_i) \in \Theta(n_u + n_i)$

$mT(n_u, n_i) \in \Theta(n_i)$

Jel.:

$J :=$ H_u és a H_i listák kulcshalmazainak $\cap$-e

$$q' := \begin{cases} q \to key & \text{ha } (*q) \text{ valódi listaelem} \\ \infty & \text{ha } (*q) \text{ a fejelem} \end{cases}$$

A ciklusok invariáns tul.-a:

$$[\min(q', r'), \max(q', r')) \cap J = \emptyset$$

Illustration of the run of the program:

$H_u \to [/]—\overset{q}{[2]}—[4]—[6]—[/] \leftarrow H_u$

$H_i \to [/]—\underset{r}{[1]}—[4]—[5]—[8]—[9]—[/] \leftarrow H_i$

$H_u \to [/]—[1]—\overset{q}{[2]}—[4]—[6]—[/] \leftarrow H_u$

$H_i \to [/]—\underset{r}{[4]}—[5]—[8]—[9]—[/] \leftarrow H_i$

$$H_u \rightarrow [/]\text{—}[1]\text{—}[2]\text{—}[\overset{q}{4}]\text{—}[6]\text{—}[/] \leftarrow H_u$$
$$H_i \rightarrow [/]\text{—}[\underset{r}{4}]\text{—}[5]\text{—}[8]\text{—}[9]\text{—}[/] \leftarrow H_i$$

$$H_u \rightarrow [/]\text{—}[1]\text{—}[2]\text{—}[4]\text{—}[\overset{q}{6}]\text{—}[/] \leftarrow H_u$$
$$H_i \rightarrow [/]\text{—}[4]\text{—}[\underset{r}{5}]\text{—}[8]\text{—}[9]\text{—}[/] \leftarrow H_i$$

$$H_u \rightarrow [/]\text{—}[1]\text{—}[2]\text{—}[4]\text{—}[5]\text{—}[\overset{q}{6}]\text{—}[/] \leftarrow H_u$$
$$H_i \rightarrow [/]\text{—}[4]\text{—}[\underset{r}{8}]\text{—}[9]\text{—}[/] \leftarrow H_i$$

$$H_u \rightarrow [/]\text{—}[1]\text{—}[2]\text{—}[4]\text{—}[5]\text{—}[6]\text{—}[\overset{q}{/}] \leftarrow H_u$$
$$H_i \rightarrow [/]\text{—}[4]\text{—}[\underset{r}{8}]\text{—}[9]\text{—}[/] \leftarrow H_i$$

$$H_u \rightarrow [/]\text{—}[1]\text{—}[2]\text{—}[4]\text{—}[5]\text{—}[6]\text{—}[8]\text{—}[\overset{q}{/}] \leftarrow H_u$$
$$H_i \rightarrow [/]\text{—}[4]\text{—}[\underset{r}{9}]\text{—}[/] \leftarrow H_i$$

$$H_u \rightarrow [/]\text{—}[1]\text{—}[2]\text{—}[4]\text{—}[5]\text{—}[6]\text{—}[8]\text{—}[9]\text{—}[\overset{q}{/}] \leftarrow H_u$$
$$H_i \rightarrow [/]\text{—}[4]\text{—}[/] \leftarrow H_i$$