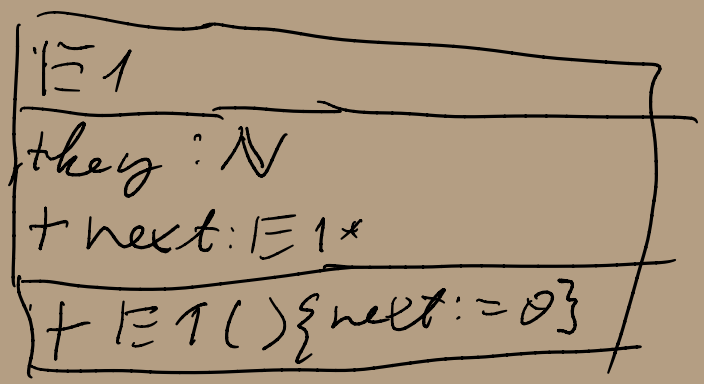
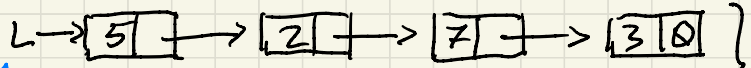
Linked Lists

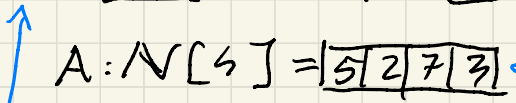


2021
spring
semester

Simple one-way lists (S1L)



(Empty list: $L = \Omega$)



ALTERNATIVE REPRESENTATIONS OF THE ABSTRACT SEQUENCE $\langle 5, 2, 7, 3 \rangle$

LINKED REPRESENTATION

SEQUENTIAL REPR.

A[i] – refers directly to the i th element : $\Theta(1)$ time

L[i] is not possible.

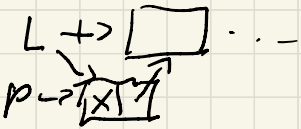
element (L : E1* ; i : N) : E1*

```
p := L ; j := 1
-------------------
j < i
|  j++ ; p := p -> next
-------------------
return p
```

$\Theta(i)$ time

insert_at_front (L : E1* ; x : N)

```
p := new E1
p -> key := x
p -> next := L
L := p
```



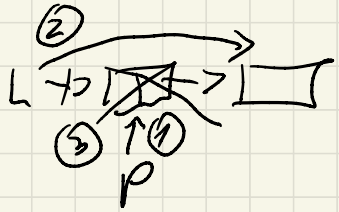
...

q := element (L , i) ; print (q -> key)

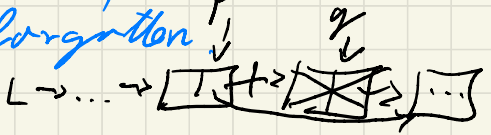
x

delete_first(&L : E1*)

```
p := L
L := L -> next
delete p
```



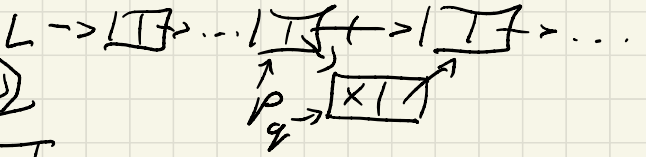
MEMORY LEAKING, if forgotten



L ≠ ∅

delete_next(p : E1*)

```
q := p -> next
p -> next := q -> next
delete q
```

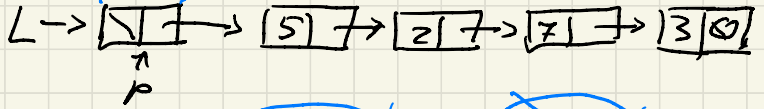


follow(p : E1* ; x : N)

```
q := new E1
q -> key := x
q -> next := p -> next
p -> next := q
```

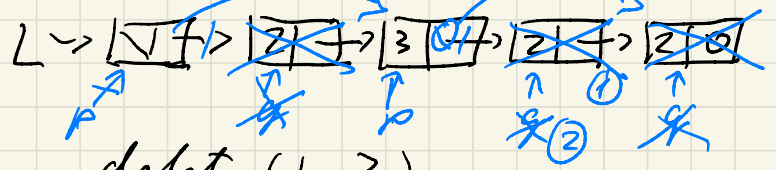
(header)

One-way lists with sentinel (head) (H1L)

(5, 2, 7, 3) => represented: 

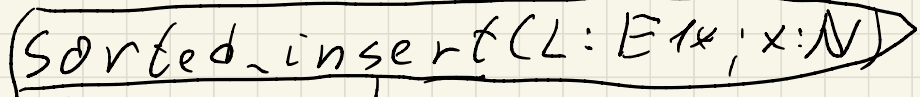
delete_all(L : E1* ; x : N)

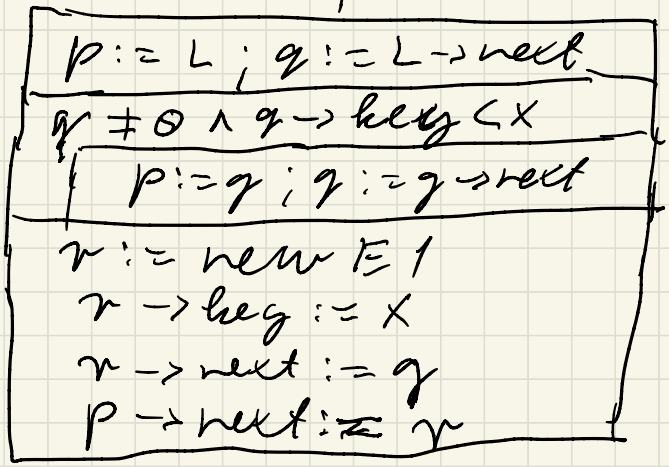
```
p := L
p -> next ≠ Ø
    p -> next -> key = x
        delete_next(p)      p := p -> next
```

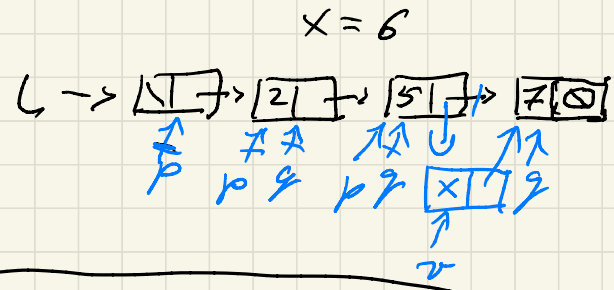


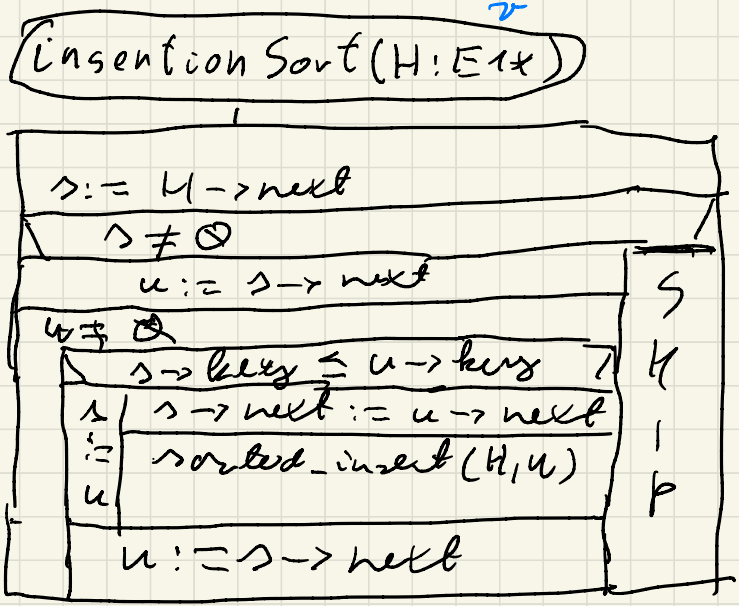
delete (L, 2)

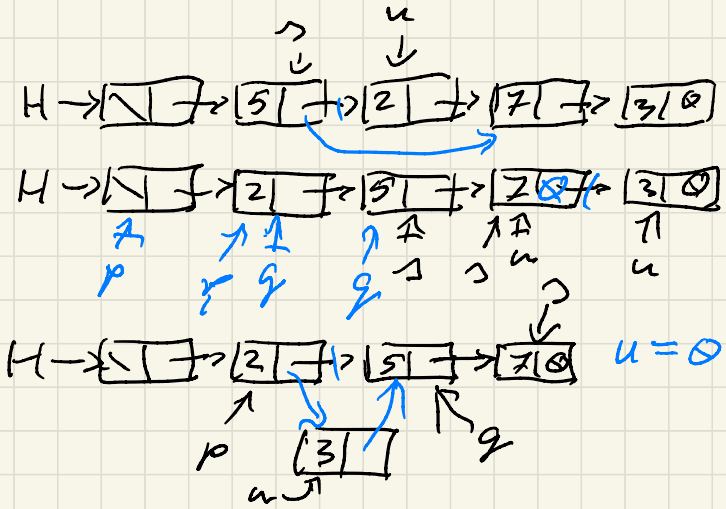
Empty list: L -> |∅|





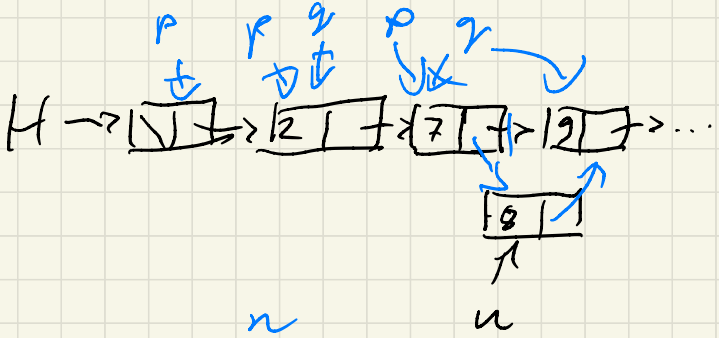






$AT(n), MT(n) \in \Theta(n^2)$

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



sorted_insert(H, u: E1*

```
p := H; q := H->next
q -> key ≤ u -> key
    p := q; q := q->next
u->next := q
p->next := u
```

$$MT_{IS}(n) = \overbrace{1}^{} + \overbrace{n-1}^{n} + n-1 + $$

$$+ 0+1+2+3+\dots+(n-2)$$

$$= \sum_{i=0}^{n} i = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n \in \Theta(n^2)$$

$$AT_{IS}(u) = \overbrace{n + (n-1)}^{2n-1} + \frac{0+1+2+\dots+(n-2)}{2} = n + (n-1) + \frac{(n-1)(n-2)}{4} =$$

$$= \frac{1}{4}n^2 + 1\frac{1}{4}n - \frac{1}{2} \in \Theta(n^2)$$

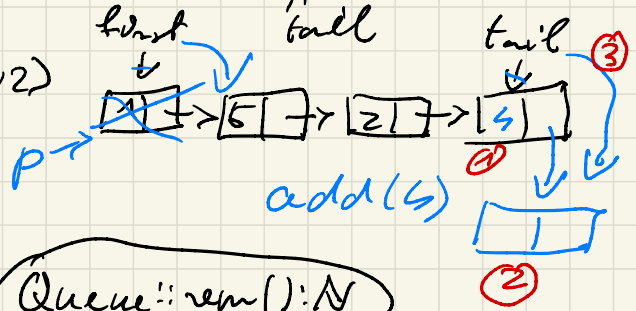
Queue

- n: ℕ
- first, tail: ↑E1

+Queue() {first := tail := new E1; n := 0}
+add(x: ℕ) {tail→key := x; tail := tail→next := new E1; n++}
+rem(): ℕ
+length(): ℕ { return n }
+first(): ℕ

Empty queue: first → [ ] sentinel tail

queue ⟨1, 5, 2⟩



Queue::rem(): ℕ

first = tail

Queue-underflow

x := first→key ; n--
p := first
first := first→next
delete p ; return x

Empty Queue: first = ∅

Queue: ⟨1, 5, 2⟩ [1|•] → [5|•] → [2|∅]

first, tail