Singly linked list.

SLL - search

```
function search ( sll , elem ) is:
        current ← sll.head
        while current ≠ NIL                    dereference with square brackets
                                                because current is
                                                a pointer
            if [current].info == elem then:
                search ← current
            current ← [current].next
        search ← current
```

O(n)

```
subalg insertFirst (sll, elem) is:
    newNode ← allocate()
    [newNode].info ← elem
    [newNode].next ← sll.head
    sll.head ← newNode
end
```

$\Theta(1)$

```
subalg insertPos (sll, p, elem) is:
    if p < 1 then
        @ throw an exception
    else if p = 1 then
        insertFirst (sll, elem)
    else
        currentNode ← sll.head
        currentPos ← 1
        while currentNode ≠ NIL and
              currentPos < p-1 then
            currentNode ← currentNode.next
            currentPos ← currentPos + 1
```

if currentNode ≠ NIL then

newNode = allocate()

[newNode].info ← elem

[newNode].next ← [currentNode].next

[currentNode].next ← newNode

else

throw exception

$\Theta(n)$