

Doubly linked lists.

- Subalg nextLast (dll, elem):

newNode = allocate()

[newNode].info \leftarrow elem

[newNode].next \leftarrow NULL

[newNode].prev \leftarrow dll.tail

If dll.head == NULL Then

dll.head \leftarrow newNode

dll.tail \leftarrow newNode

else

(dll.tail).next \leftarrow newNode

dll.tail \leftarrow newNode

→ Subalg insertOnPosition (dll, elem, position):

if position < 1

② throw exception

if position == 1 then

newNode ← allocate()

[newNode].info ← elem

[newNode].prev ← NIL

[newNode].next ← dll.head

if dll.head ≠ NIL:

[dll.head].prev ← newNode

dll.head ← newNode

else:

dll.head ← newNode

dll.tail ← newNode

currentNode ← dll.head

currentPos ← 1

while currentNode \neq Nil and
currentPos < position

currentNode \leftarrow [currentNode].next

currentPos \leftarrow currentPos + 1

if currentPos = position then

newNode \leftarrow allocate()

[newNode].info \leftarrow elem

[newNode].next \leftarrow currentNode

if currentNode \neq Nil

[newNode].prev \leftarrow [currentNode].prev

[currentNode].prev.next \leftarrow newNode

[currentNode].prev \leftarrow newNode

else:

[newNode].prev \leftarrow dll.tail

[dll.tail].next \leftarrow newNode

dll.tail \leftarrow newNode

Subalg

rotate list (sll)

if sll.head \neq Nil and [sll.head].next \neq Nil then:

firstNode = sll.head

sll.head \leftarrow [sll.head].next

currentNode \leftarrow sll.head

while [currentNode].next \neq Nil

currentNode \leftarrow [currentNode].next

[currentNode].next \leftarrow firstNode

[firstNode].next \leftarrow Nil