XORNode:
  info: TELem
  link: ↑ XORNode

XORList:
  head: ↑ XORNode
  tail: ↑ XORNode

subalgorithm printListForward(xorl) **is:**
//pre: xorl is a XORList
//post: true (the content of the list was printed)
  prevNode ← NIL
  currentNode ← xorl.head
  while currentNode ≠ NIL **execute**
    write [currentNode].info
    nextNode ← prevNode XOR [currentNode].link
    prevNode ← currentNode
    currentNode ← nextNode
  end-while
end-subalgorithm

- Complexity: $\Theta(n)$

subalgorithm addToBeginning(xorl, elem) **is:**
//pre: xorl is a XORList
//post: a node with info elem was added to the beginning of the list
  newNode ← allocate()
  [newNode].info ← elem
  [newNode].link ← xorl.head
  if xorl.head = NIL **then**
    xorl.head ← newNode
    xorl.tail ← newNode
  else
    [xorl.head].link ← [xorl.head].link XOR newNode
    xorl.head ← newNode
  end-if
end-subalgorithm

- Complexity: $\Theta(1)$