

Introduction to Singly Linked List

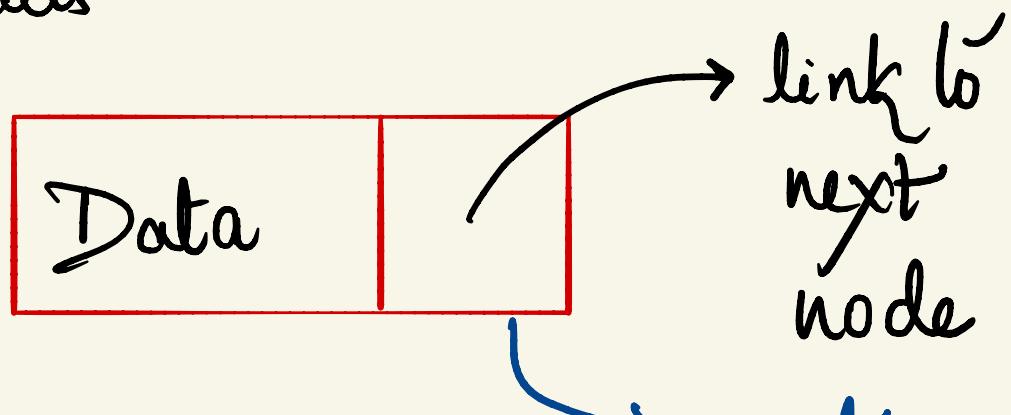
Singly linked list

Linked List ADT:

Collection of objects which are accessed sequentially.

In linked list terminology, each objects are called nodes.

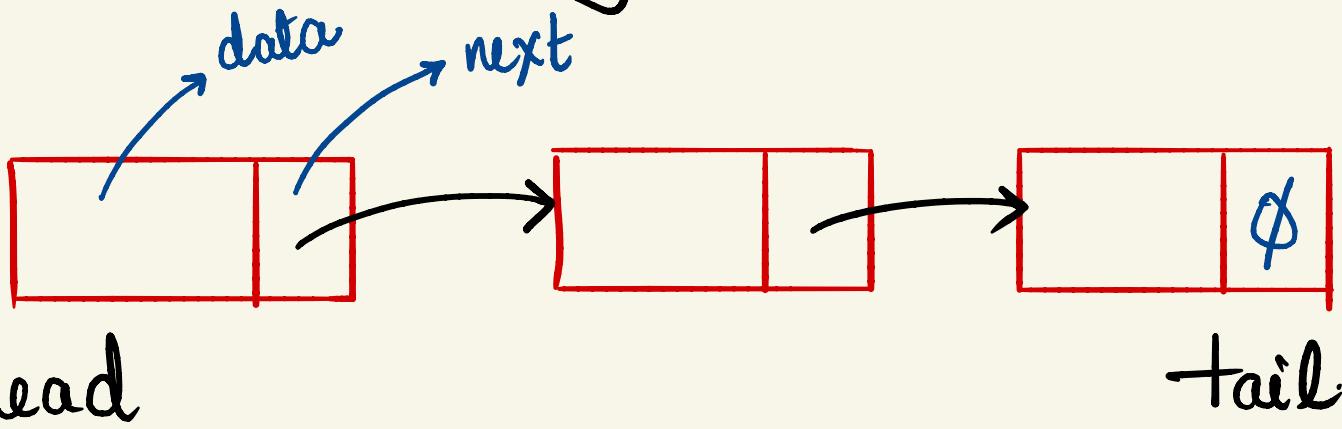
A node of singly linked list consists of two parts:



1. Data
2. Link to the next node

Two special nodes which are identified by head and tail.

Consider a following singly linked list



tail node link section is pointing to nowhere as there are no further nodes in the linked list.

Various Operations on Linked List.

1. addLast(item)

Adding a node with data as item to the end of the list.

2. addFirst(item)

Adding a node with data as item at the start of the linked list.

3. delLast()

Deleting a last node.

4. delFirst()

Deleting the first node.

5. delNode(item)

Delete the node with with data as item from the list.

6. size()

Returns the size or no. of nodes in the list.

7. display()

Display all items inside a list in sequential order.

Pseudo - Codes for the operations .

addLast(sll , item)

// sll : is the given singly linked list.

// item : data of new node

newnode. data \leftarrow item

newnode. next \leftarrow null

if (sll. head is null)

sll. head \leftarrow newnode

sll. tail \leftarrow newnode

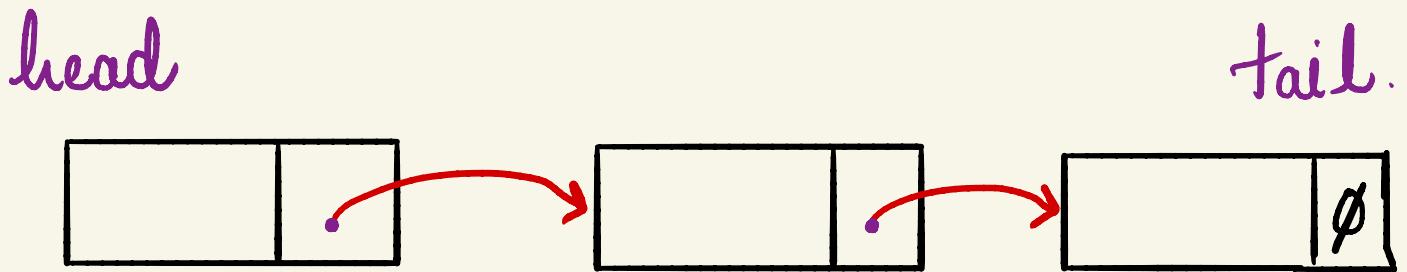
else

sll. tail. next \leftarrow newnode

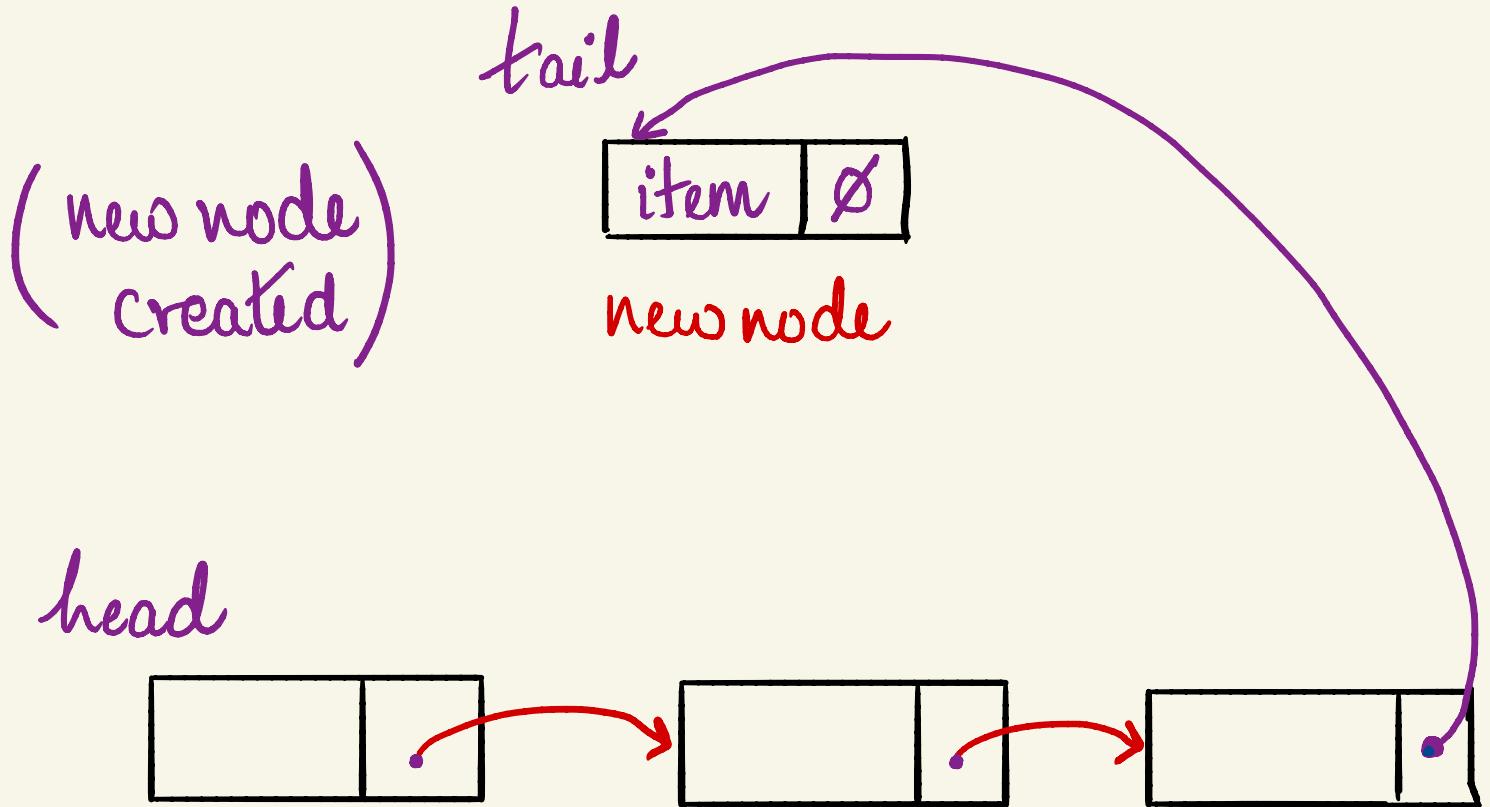
sll. tail \leftarrow newnode

endif

end addLast



sll (Given)



(addLast Explanation)

addfirst(sll , item)

// sll: singly linked list

// item: data of new node

newnode . data \leftarrow item

newnode . next \leftarrow null



if (sll . head == null)

 sll . head \leftarrow newnode

 sll . tail \leftarrow newnode

else

 newnode . next \leftarrow sll . head

 sll . head \leftarrow newnode

endif.

end addfirst

head

tail.



sll (Given)

head



new node

tail.



sll (Given)

delLast (sll)

// sll : given singly linked list

ddata $\leftarrow \infty$

if (sll.head != null)

 if (sll.head == sll.tail)

 ddata \leftarrow sll.tail.data

 sll.head \leftarrow null

 sll.tail \leftarrow null

 else

 temp \leftarrow sll.head

 while (temp.next != tail)

 temp \leftarrow temp.next

 end while

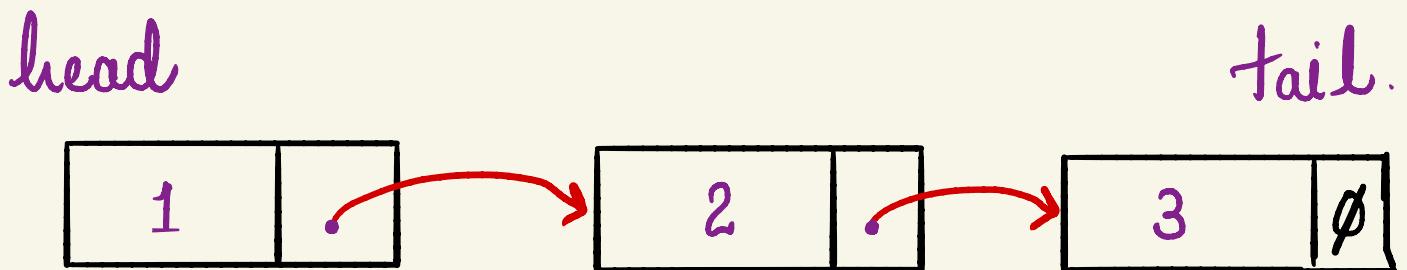
 ddata \leftarrow sll.tail.data

 temp.next \leftarrow null

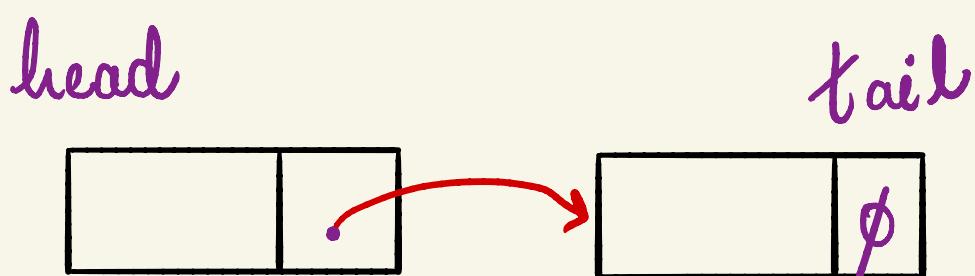
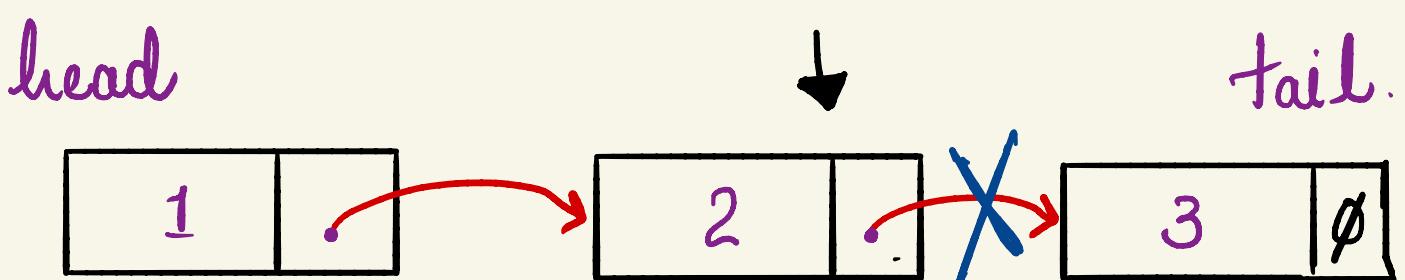
 sll.tail \leftarrow temp

 endif.

end delLast(.



sell (Given)



~~delFirst(sll)~~

// sll : given singly linked list.

ddata $\leftarrow \infty$ (indicator value)

if (sll.head != null)

 if (sll.head == sll.tail)

 ddata \leftarrow sll.head.data

 sll.head \leftarrow null

 sll.tail \leftarrow null

 else

 ddata \leftarrow sll.head.data

 sll.head \leftarrow sll.head.next

 endif

endif

return ddata ;

head

tail



head

tail



delNode (SLL, item)

Homework.

size (sll) Count no. of elements
in the linked list.

//sll : given singly linked list.

Count \leftarrow 0

temp \leftarrow sll.head

while (temp != null)

 Count \leftarrow Count +1;

 temp \leftarrow temp.next

end while

return count

end size

head

tail



(SLL)

temp = head

Count = 1

temp = temp.next

if temp != null

Count = 2

temp = temp.next

if temp != null

Count = 3

temp = temp.next

temp == null

stop.

~~display(sll)~~

//sll : given singly linked list.

temp \leftarrow sll.head

while (temp != null)

 print (temp.data)

 temp \leftarrow temp.next

end while

end display