Data Structure & Algorithms

Sunbeam Infotech

Linked List - Singly List

1) display: O(n)

2) addLast: O(n)

3) addFirst: O(1)

4) addAtPos: O(pos)

   avg: O(n)

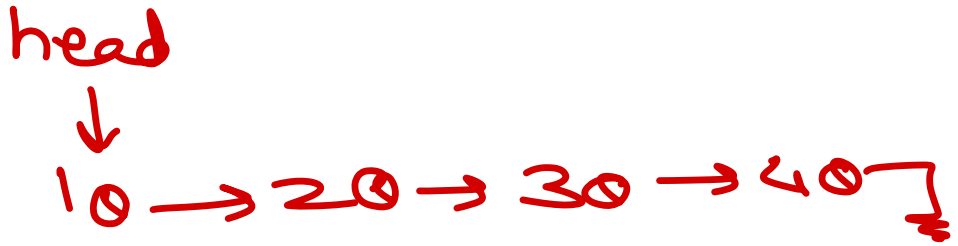
5) delFirst: O(1)

6) delLast: O(n)

7) delAtPos: O(pos)

   avg: O(n)

8) delAll: O(1) <- Java. (GC)



Linked List - Singly List

1) display : O(n)

2) addLast : O(1)

3) addFirst : O(1)

4) addAtPos : O(pos)

avg : O(n)

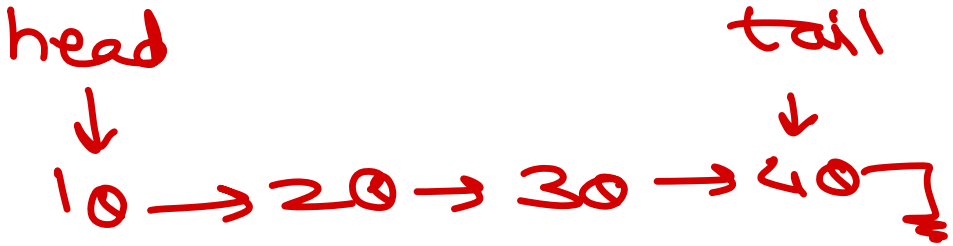
5) delFirst : O(1)

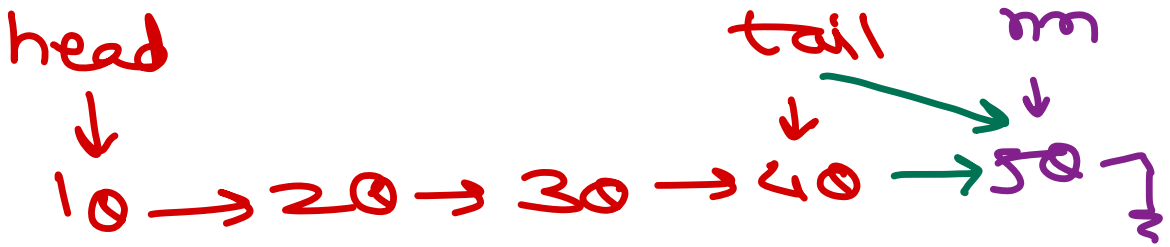
6) delLast : O(n)

7) delAtPos : O(pos)

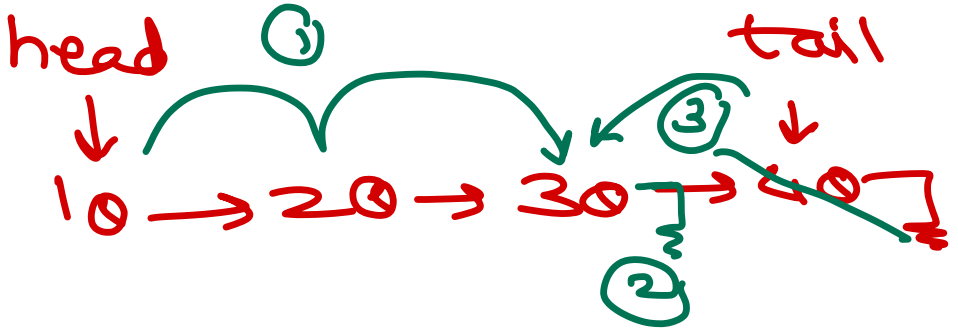
avg : O(n)

8) delAll : O(1) ← Java. (GC)



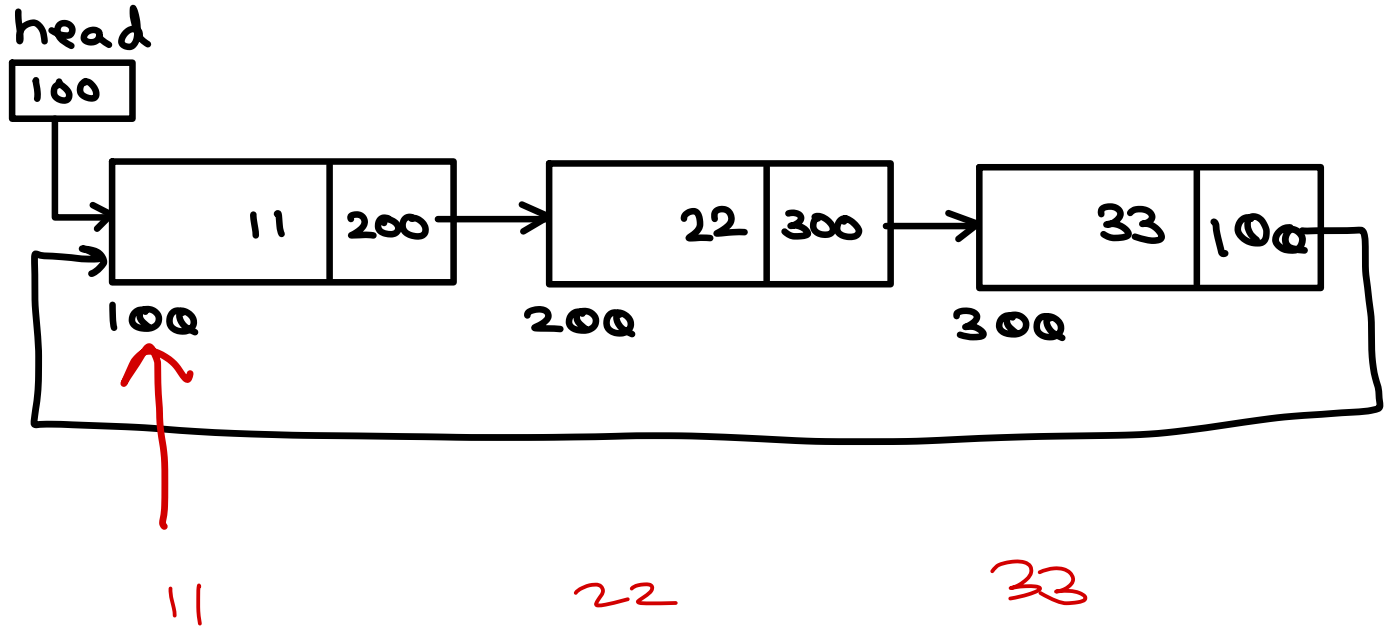



```
tail.next = nn;
tail = nn;
```



Linked List – Singly Circular List.

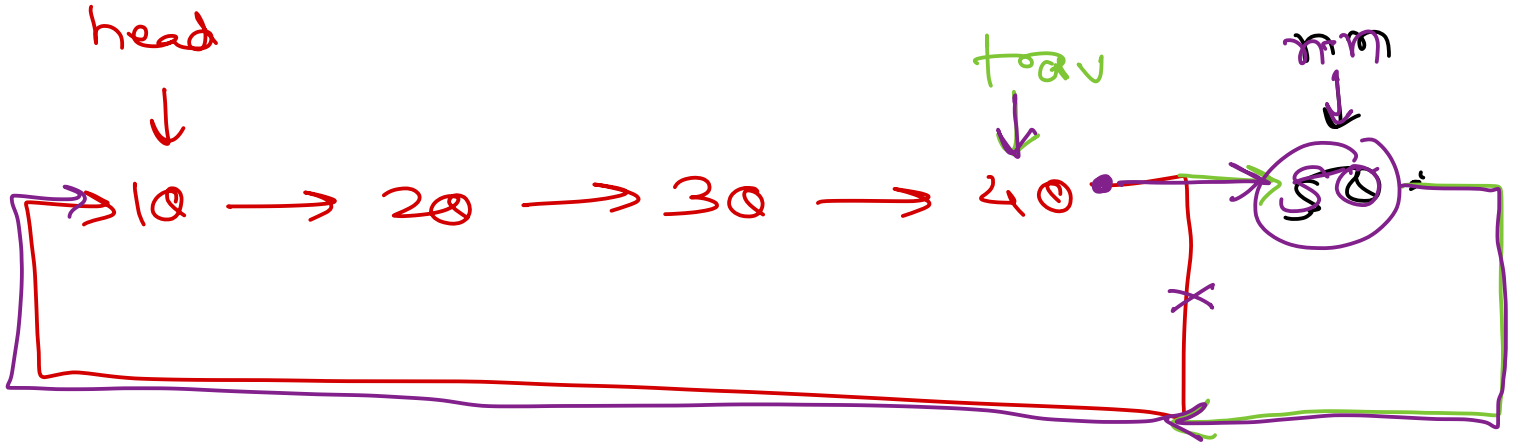
```
trav = head;
do {
    pf(trav.data);
    trav = trav.next;
} while(trav != head)
```



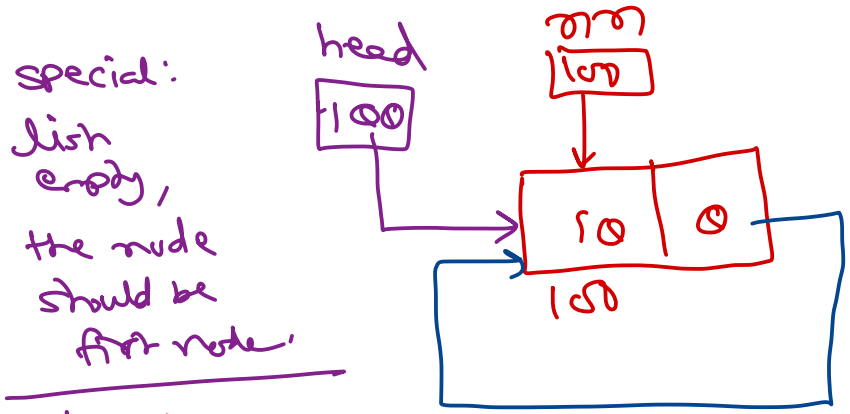
head
0

if list is empty, return;

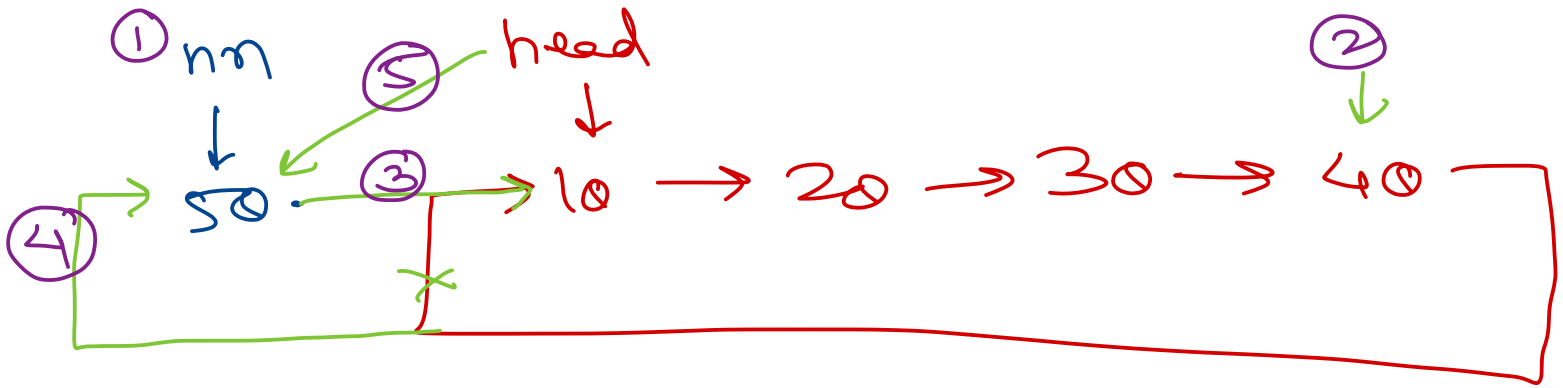
Linked List — addLast()



1. Create nn & init
2. traverse till last node (trav)
3. nn.next = head
4. trav.next = nn

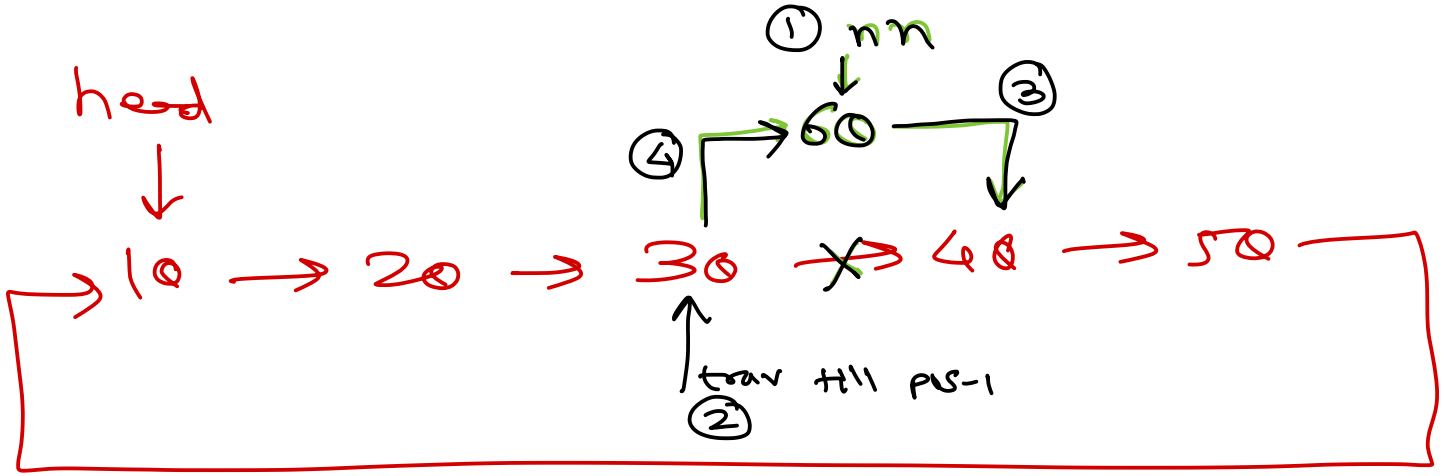


Linked List - addFirst()

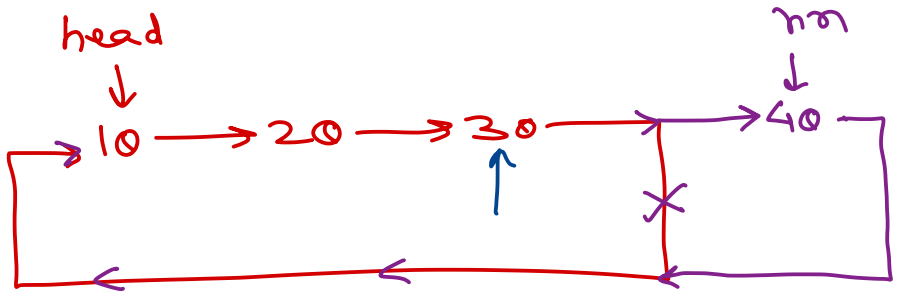


1. create nn & init
2. trav till last node
3. nn next = head
4. last node next = nn
5. head = nn;

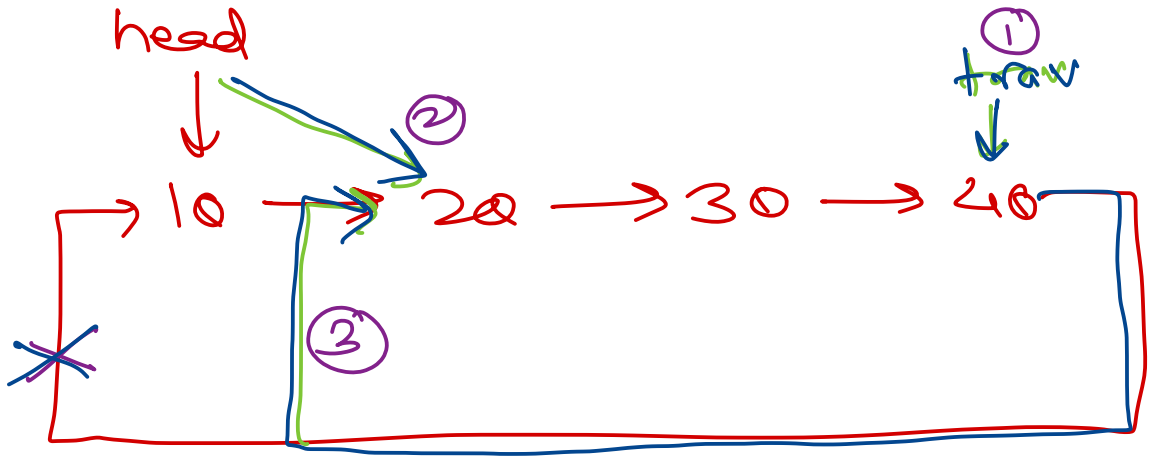
Linked List → add At Pos.



i=3 pos=8



Linked List → delFirst()

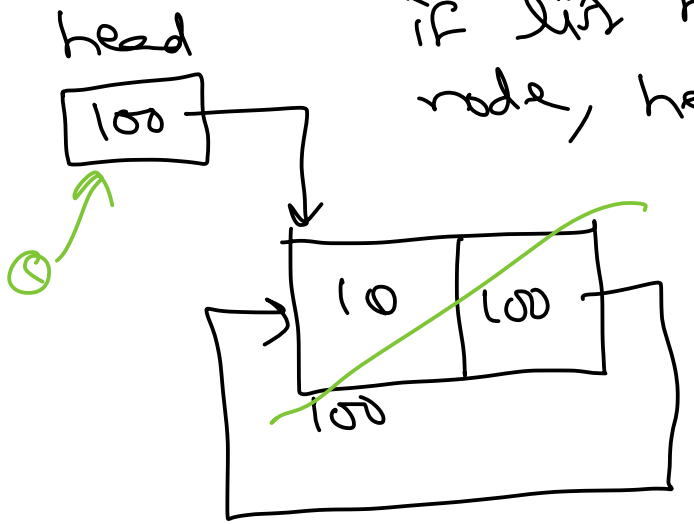


① trav till last node.

② take head to next node (2nd)

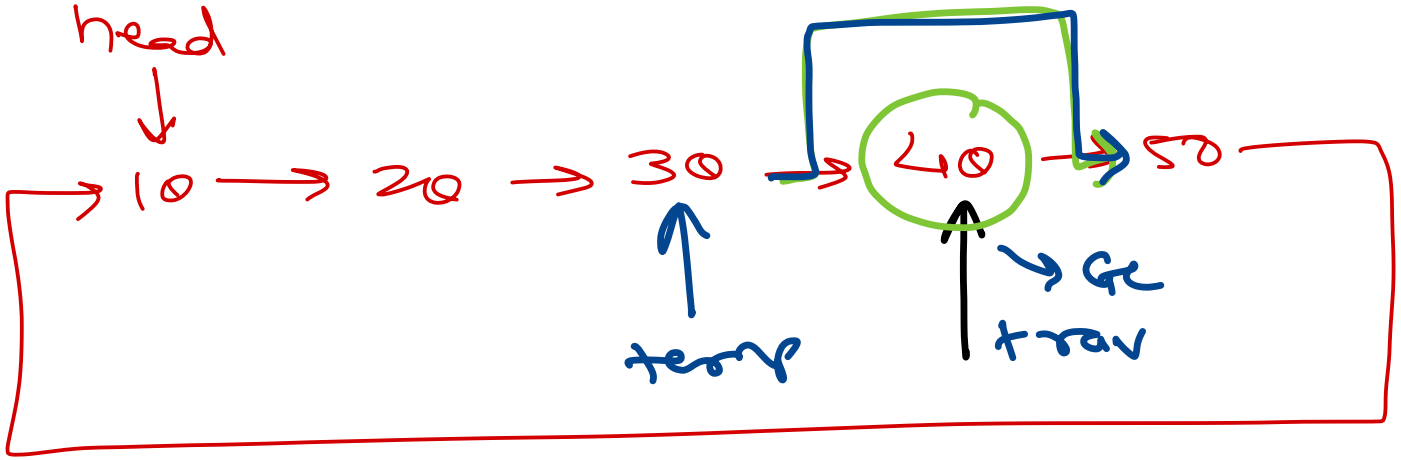
③ trav's next to new head.

if list is empty, throw exception.

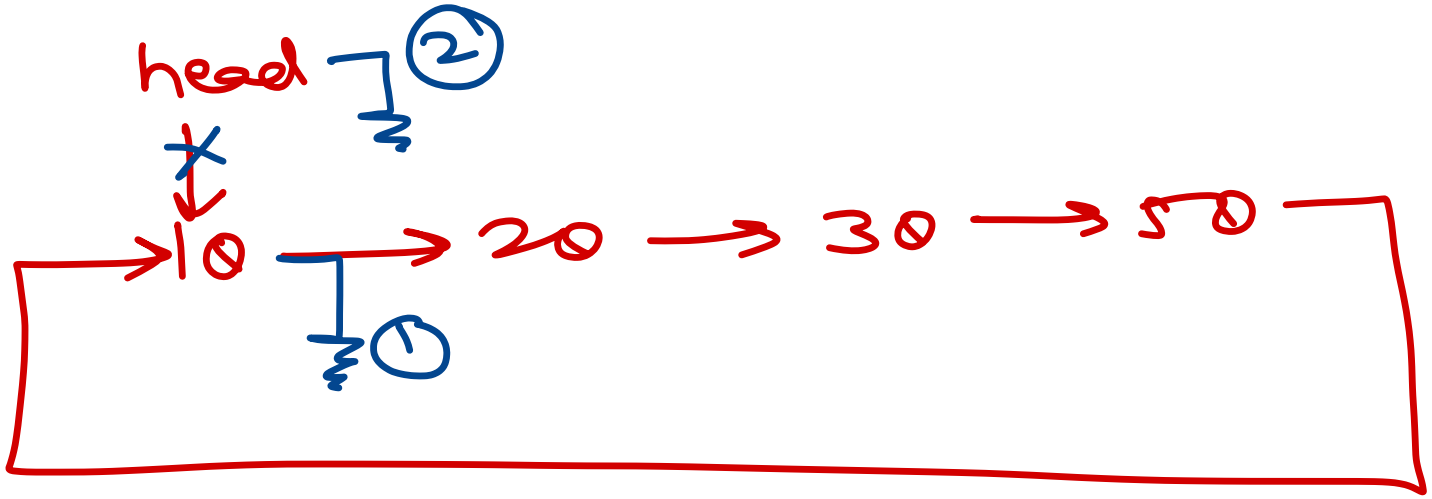


if list has single node, head = null;

Linked List → delAtPos()



Linked List



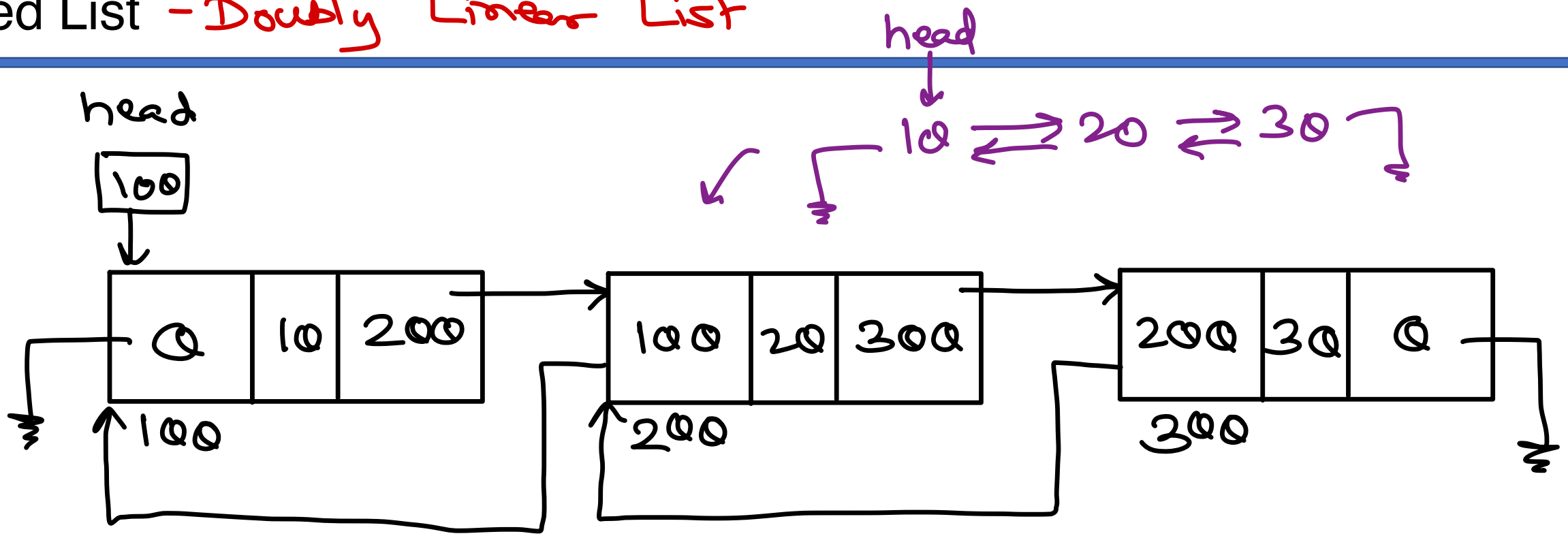
① Convert cir list into singly lin list.

```
head.next = null;
```

② make head null

```
head = null;
```

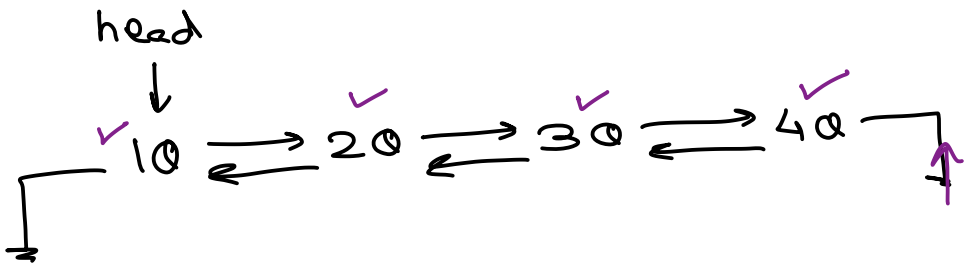
Linked List - Doubly Linear List



next → addr of next node

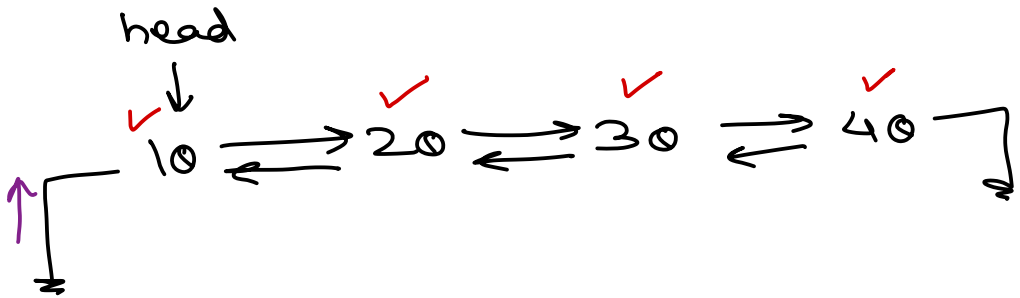
prev → addr of prev node.

Linked List – display()




```
trav = head;
while(trav != null)
{
    pf(trav.data);
    trav = trav.next;
}
```

forward display



① traverse till last node.

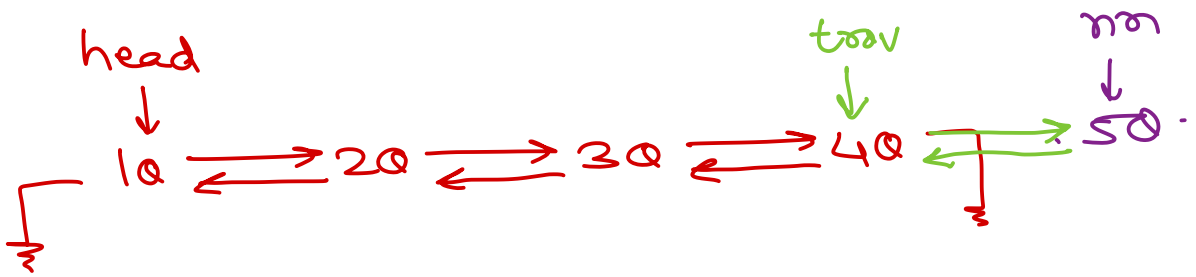
```
trav = head;
while(trav.next != null)
    trav = trav.next;
```

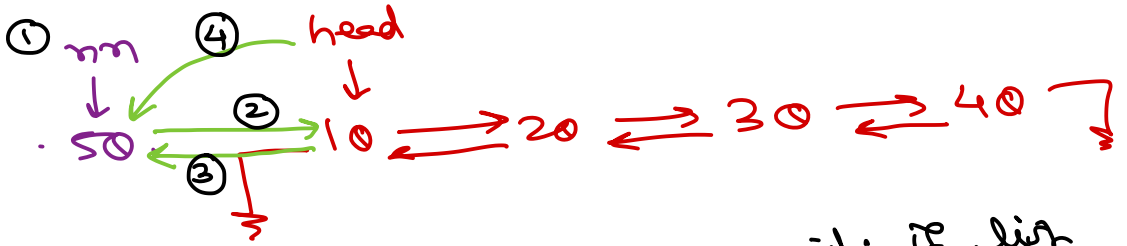
② trav & print each node in rev dir.

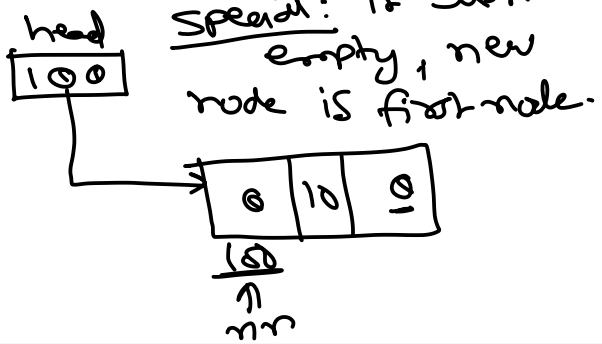
```
while(trav != null)
{   print(trav.data)
    trav = trav.prev;
}
```

reverse display

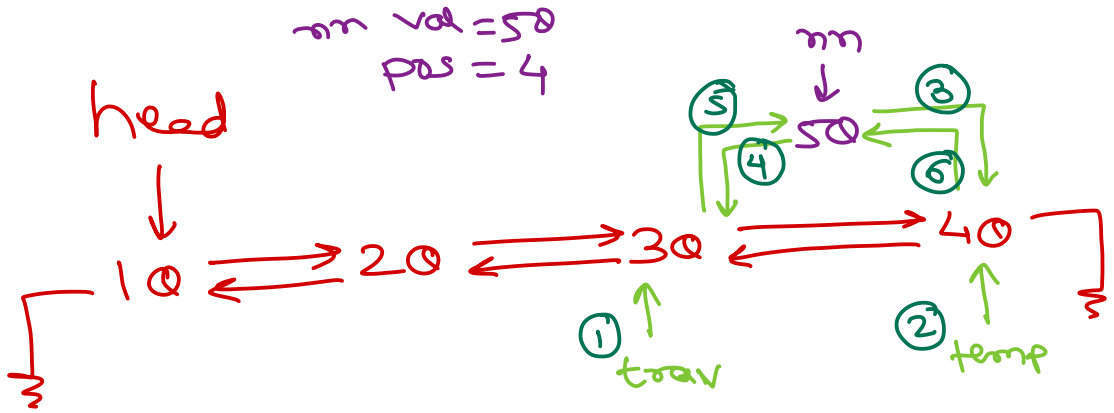
Linked List – addLast() /addFirst()

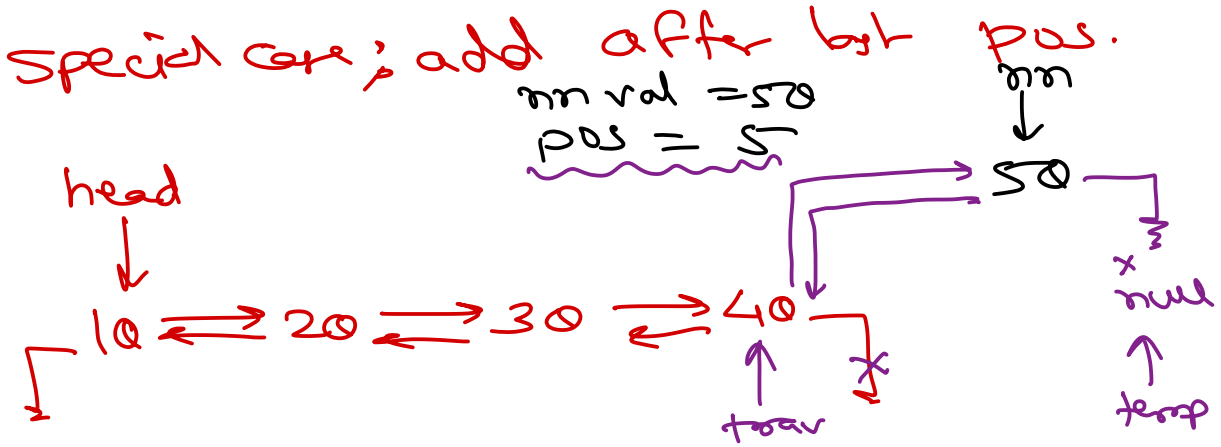




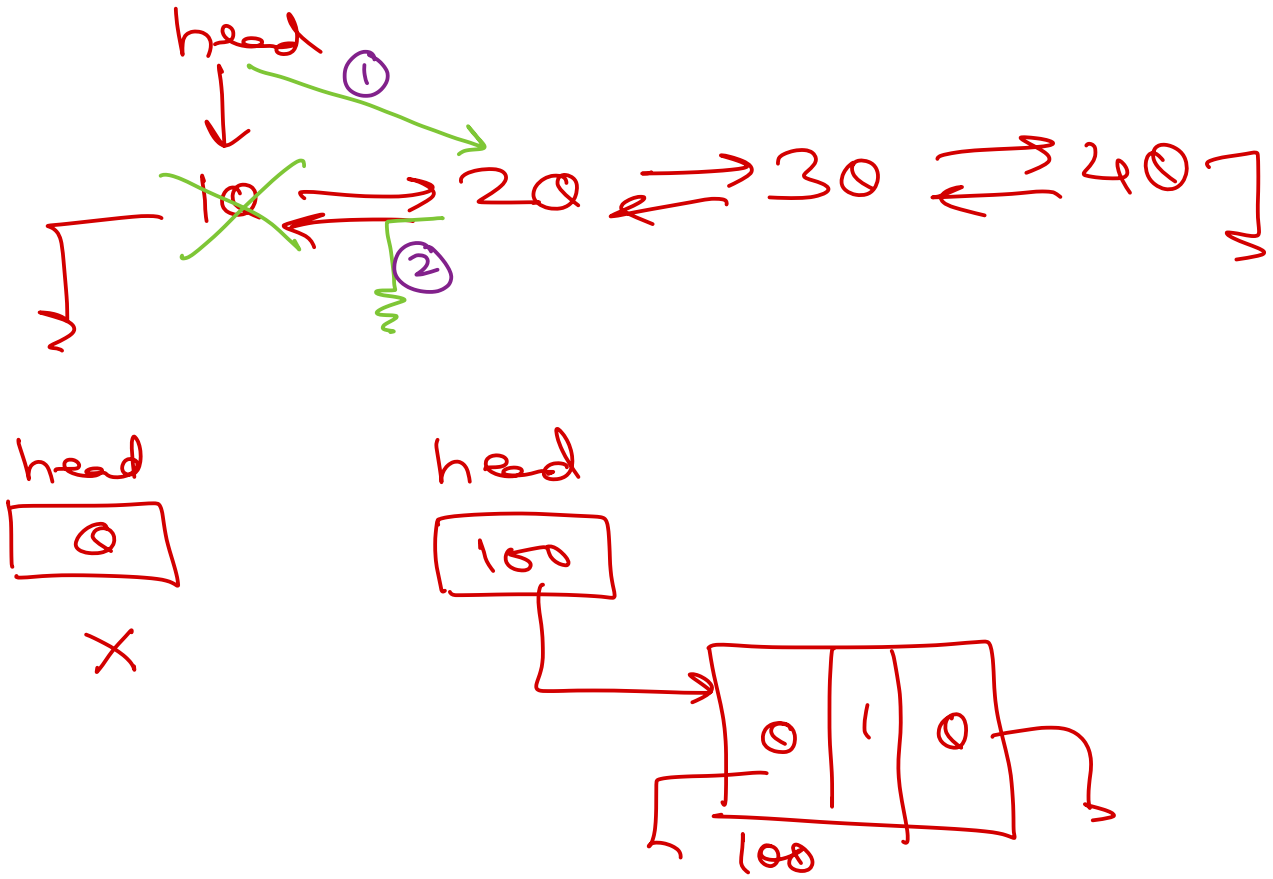


Linked List

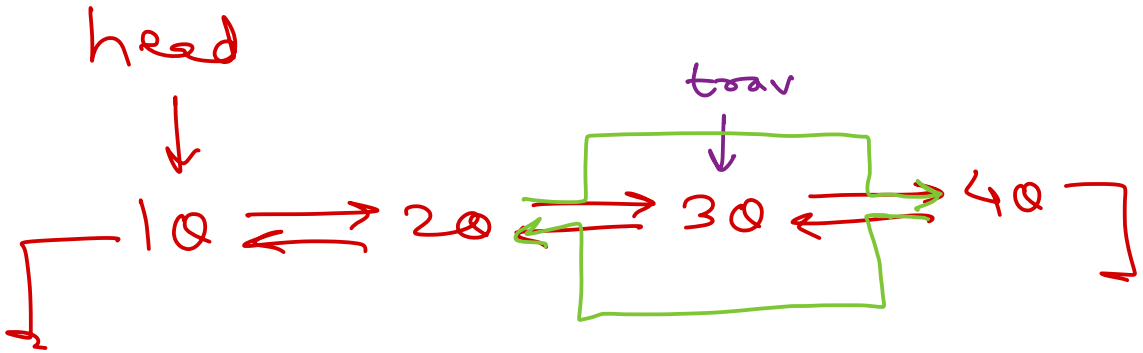

nn val = 50
pos = 4
nn
head
10
20
30
40
50
1
2
3
4
5
6
trav
temp


special care; add after last pos.
nn val = 50
pos = 5
nn
head
10
20
30
40
50
trav
temp
null

Linked List – del first()



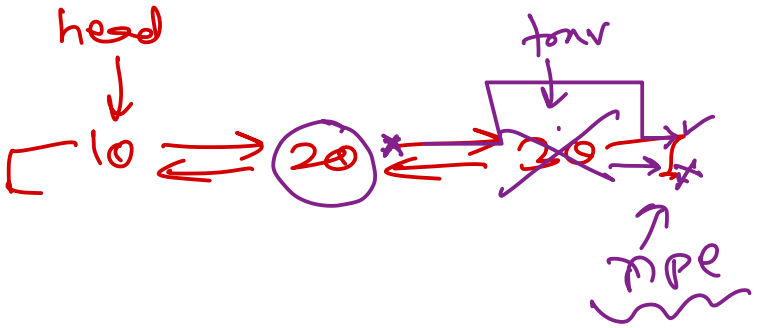
Linked List – del 'last'



① traverse till pos.

② previous node's next to trav's next;

③ next node's prev to trav's prev node



Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>