SUNBEAM

Sunbeam Institute of Information Technology

Pune and Karad

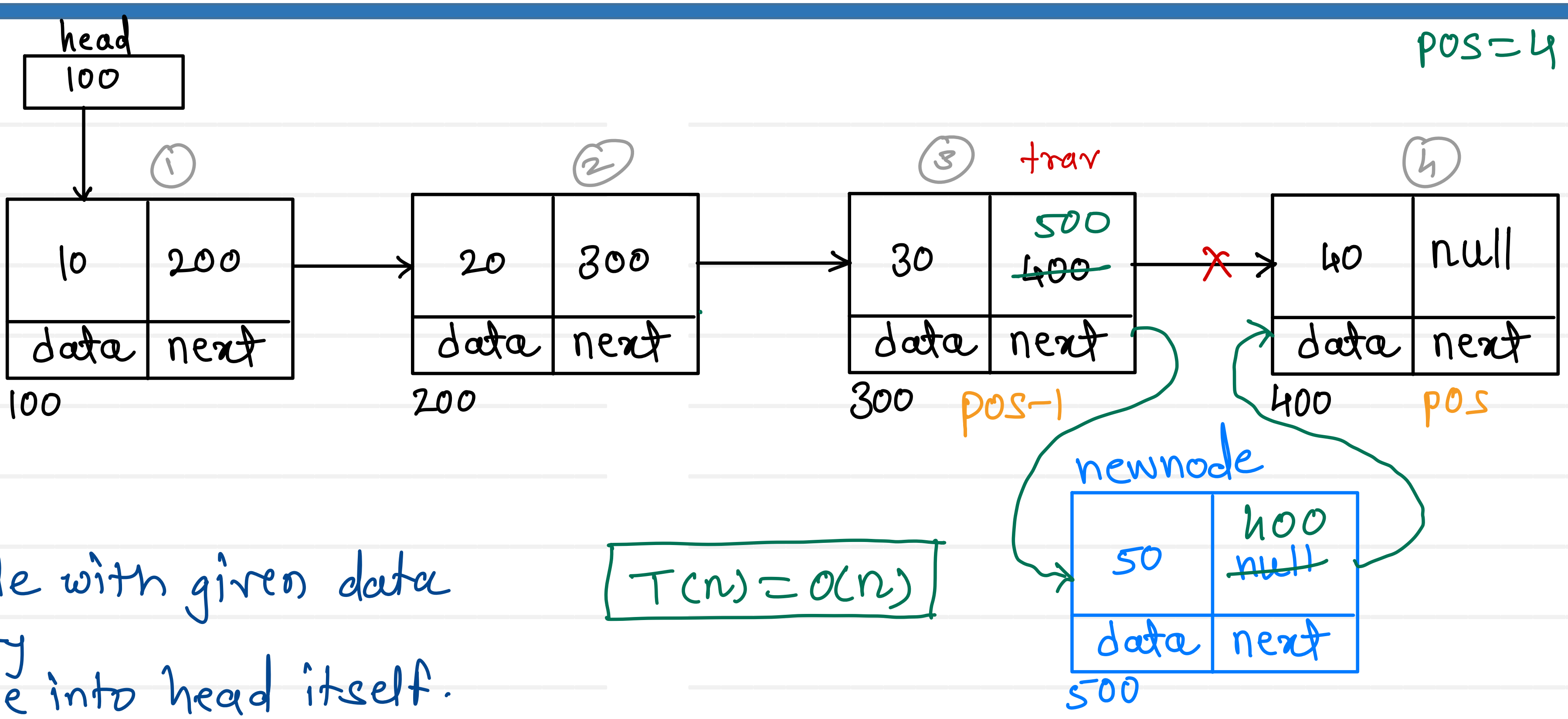
Module – Data Structures and Algorithms

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Singly linear Linked List - Add position

Make before break



$T(n) = O(n)$

1. create newnode with given data
2. if list is empty
   add newnode into head itself.
3. if list is not empty
   a. traverse till pos-1 node
   b. add pos node into next newnode
   c. add newnode into next of pos-1

```
trav = head;
for( i=1 ; i < pos-1 ; i++)
    trav = trav.next;
```

pos=4

trav	i	i<3
100	1	T
200	2	T
300	3	F

head

↓

$10 \rightarrow 20 \rightarrow 30 \rightarrow 40$

pos = 1 , value = 50

trav	i	i < 0
100	1	F

head

↓

$10 \rightarrow 50 \rightarrow 20 \rightarrow 30 \rightarrow 40$

pos <= 1

```
Node trav = head
for( i=1; i<pos-1; i++)
    trav = trav.next;
newnode.next = trav.next;
trav.next = newnode.
```

pos = 6

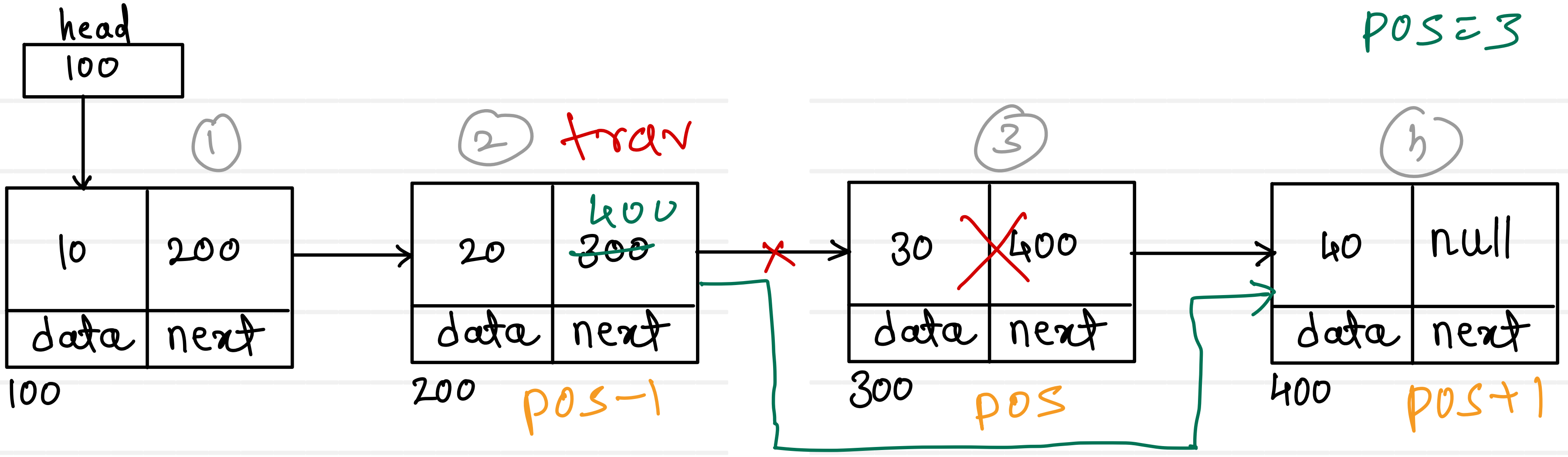
trav	i	i < 5
100	1	T
200	2	T
300	3	T
400	4	T
null	5	F

newnode.next = trav.next

↑

null pointer exception

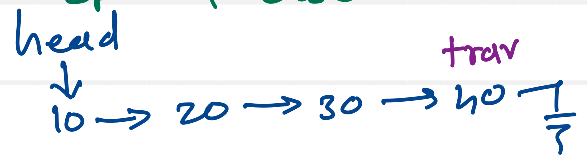
Singly linear Linked List - Delete position



1. if list is empty, return
2. if list has single node, head = null
3. if list has multiple nodes,
   a. traverse till pos-1 node
   b. add pos+1 node into next of pos-1

$$\boxed{T(n) = O(n)}$$

special case 2 :



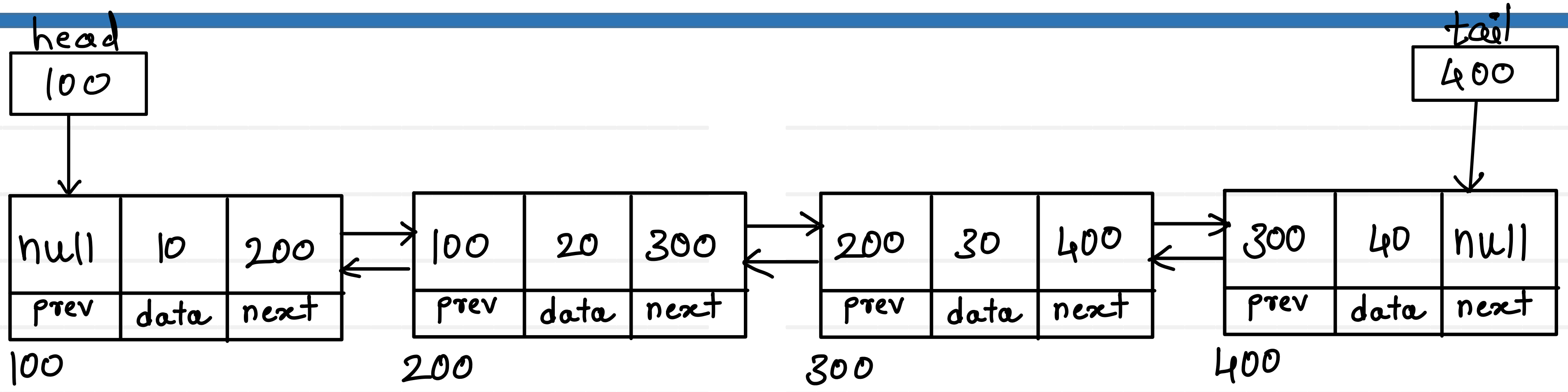
pos = 5

trav	i	i<4
100	1	T
200	2	T
300	3	T
400	4	F

trav
800
trav.next.next => null

trav.next = trav.next.next ;
→ null pointer exception

Doubly Linear Linked List - Display



Forward traversal

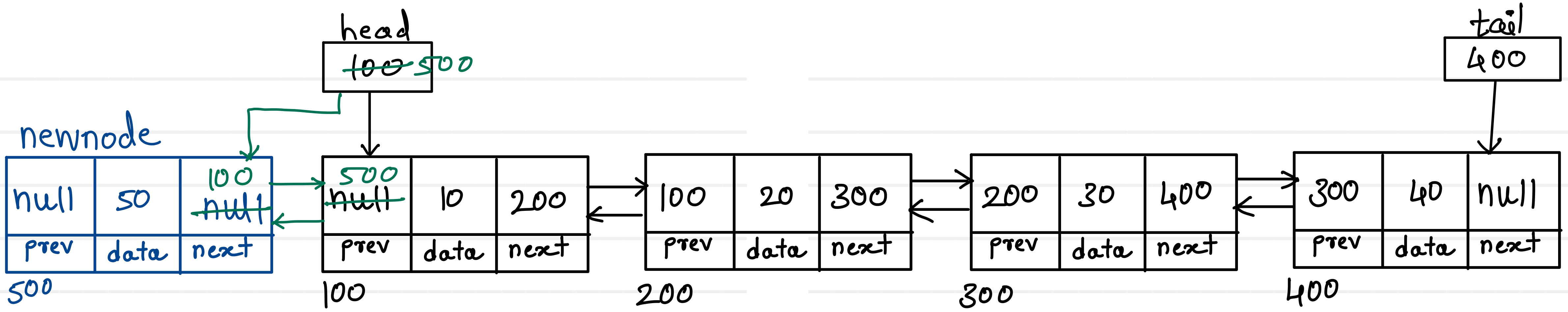
1. create trav & start at head
2. print current node data (trav.data)
3. go on next node (trav.next)
4. repeat above two steps till last node

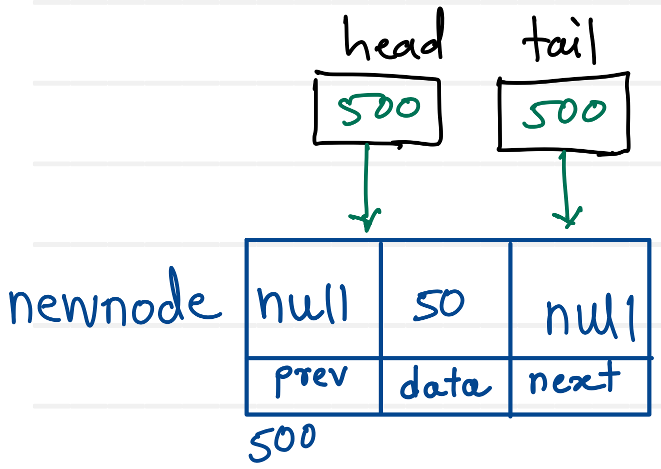
Backword traversal

1. create trav & start at tail
2. print current node data (trav.data)
3. go on prev node (trav.prev)
4. repeat above two steps till first node

$$\boxed{T(n) = O(n)}$$

Doubly Linear Linked List - Add first

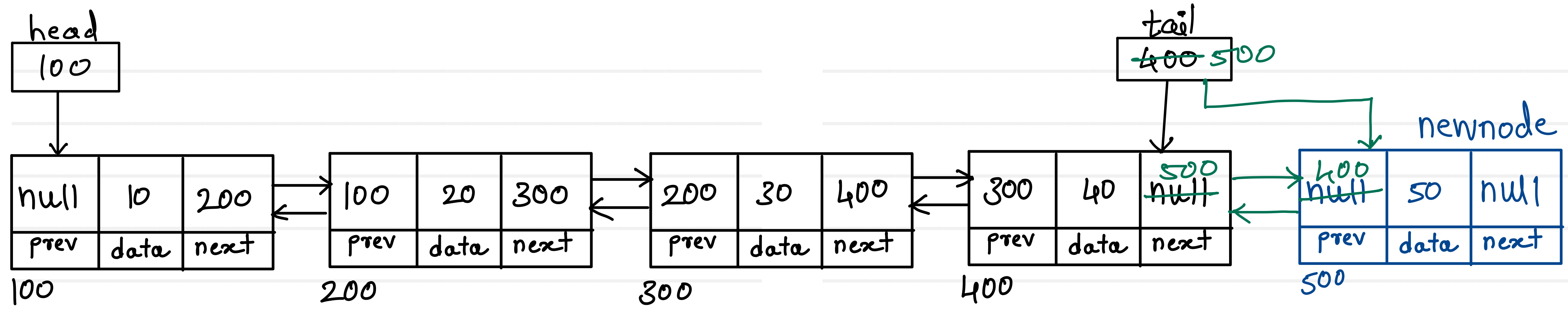


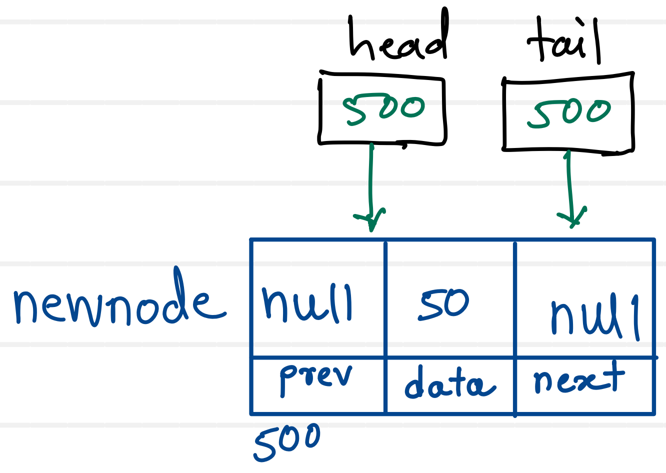


$\boxed{T(n) = O(1)}$

1. create node with data
2. if list is empty
   add newnode into head & tail
3. if list is not empty
   a. add first node into next of newnode
   b. add newnode into prev of first node
   c. move head on newnode

Doubly Linear Linked List - Add last

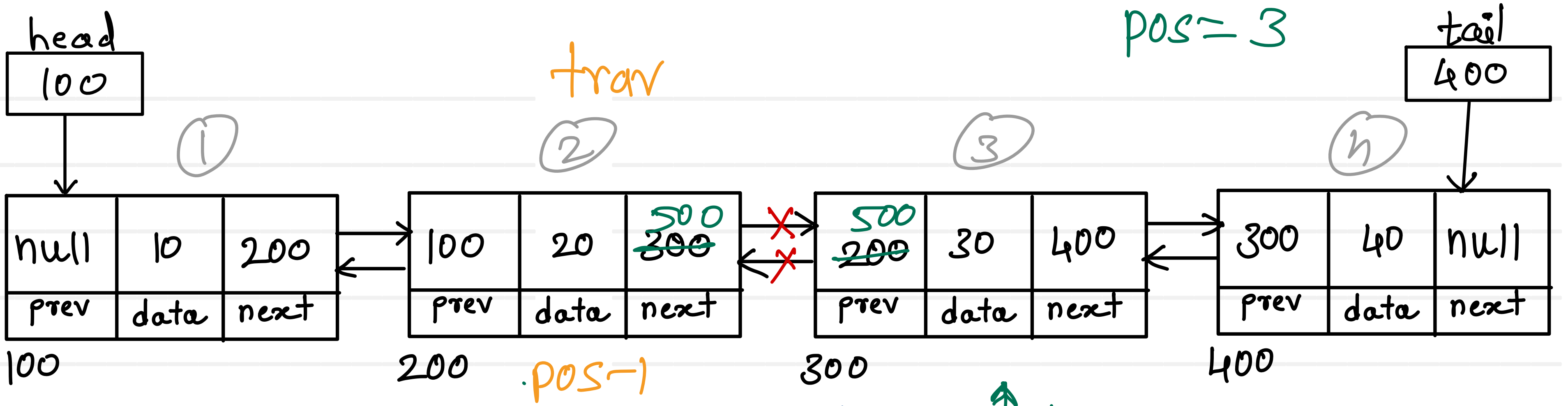


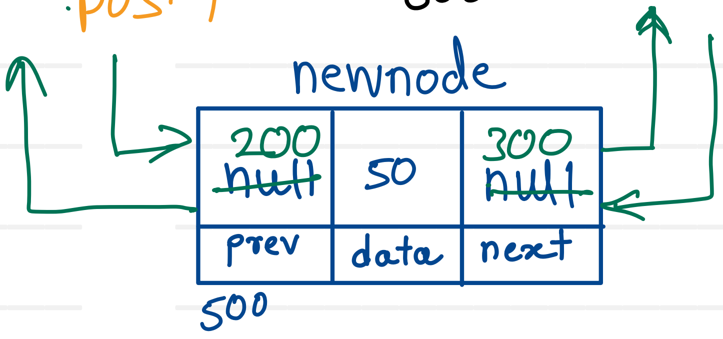


1. Create node with data
2. if list is empty,
   add newnode into head & tail
3. if list is not empty
   a. add last node into prev of newnode
   b. add newnode into next of last node
   c. move tail on newnode

$\boxed{T(n) = O(1)}$

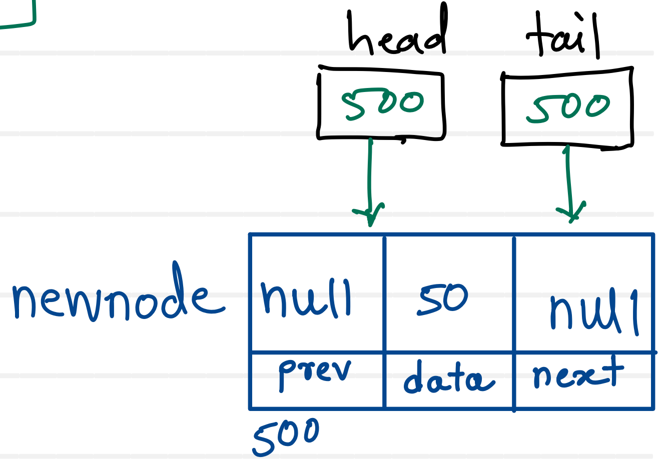
Doubly Linear Linked List - Add position

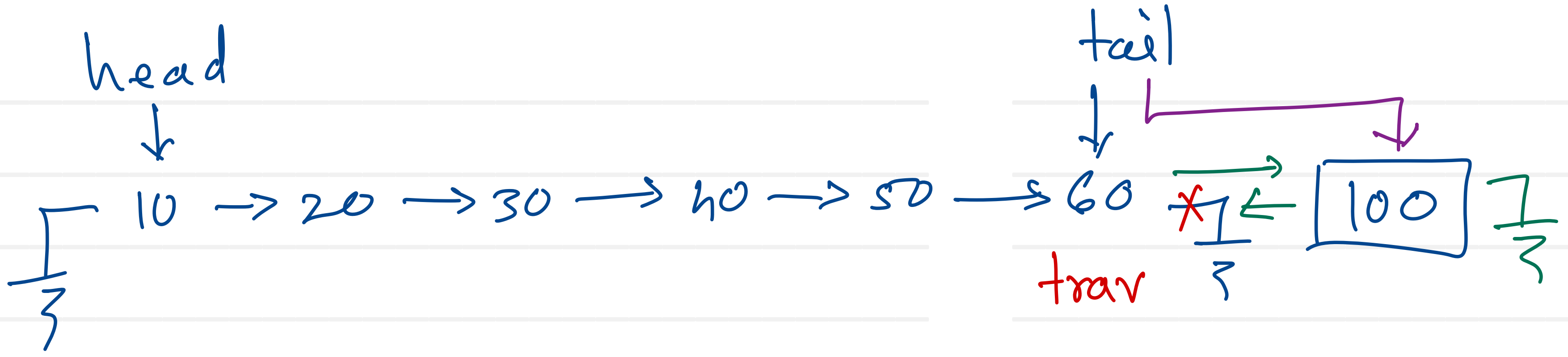




1. create node for given value
2. if list is empty, then add newnode into head & tail
3. if list is not empty
   a. traverse till pos-1 node
   b. add pos node into next of newnode
   c. add pos-1 node into prev of newnode
   d. add newnode into prev of pos node
   e. add newnode into next of pos-1 node

$$\boxed{T(n) = O(n)}$$





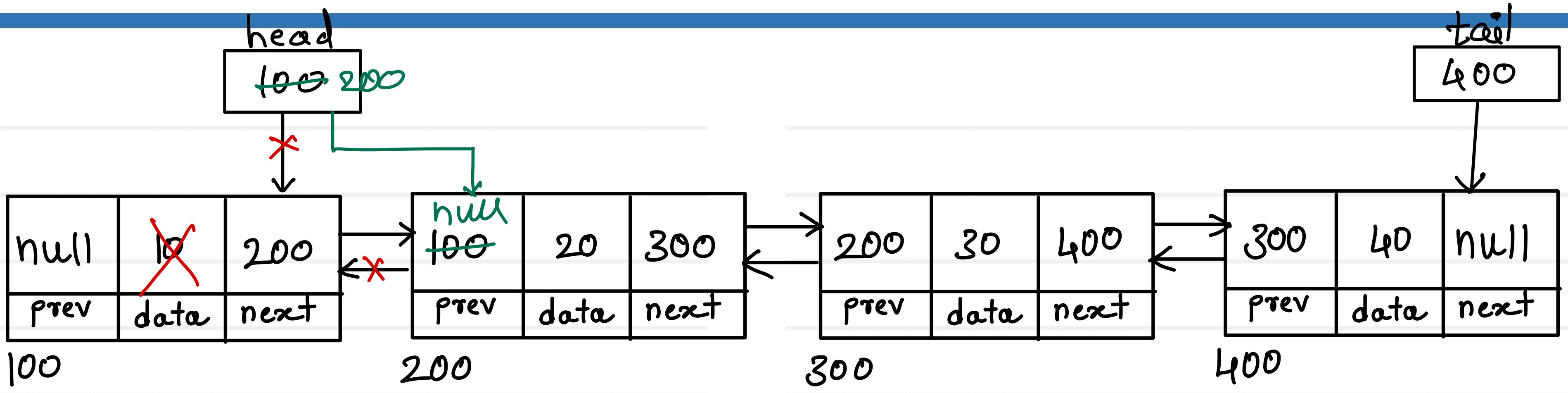
pos = 7

newnode. next = trav. next : f100 . next = null
newnode . prev = trav : f100 . prev = f60
trav . next . prev = newnode : f60 . null . next = f100
trav . next = newnode

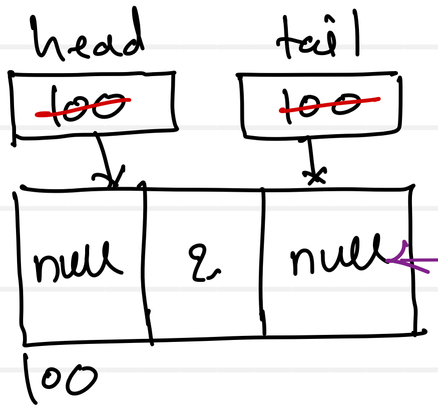
↑null pointer exception

if (trav == tail)
tail = newnode

Doubly Linear Linked List - Delete first



1. if list is empy , return
2. if list has single node , head = tail = null
3. if list has multiple nodes
   a. make prev of second node null
   b. move head on second node

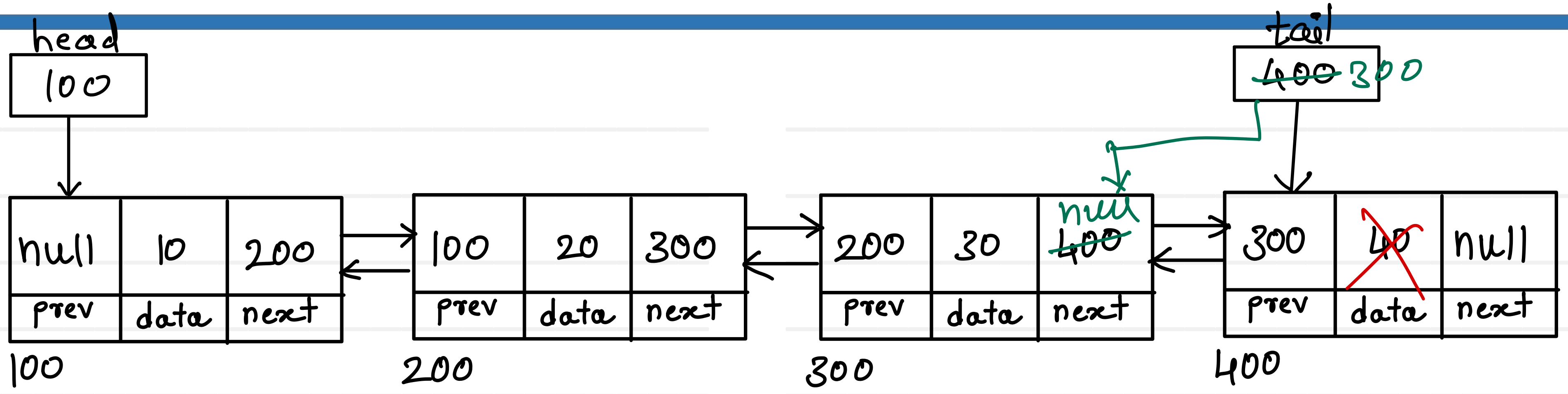


head.next.prev = null

head = head.next

$\boxed{T(n) = O(1)}$

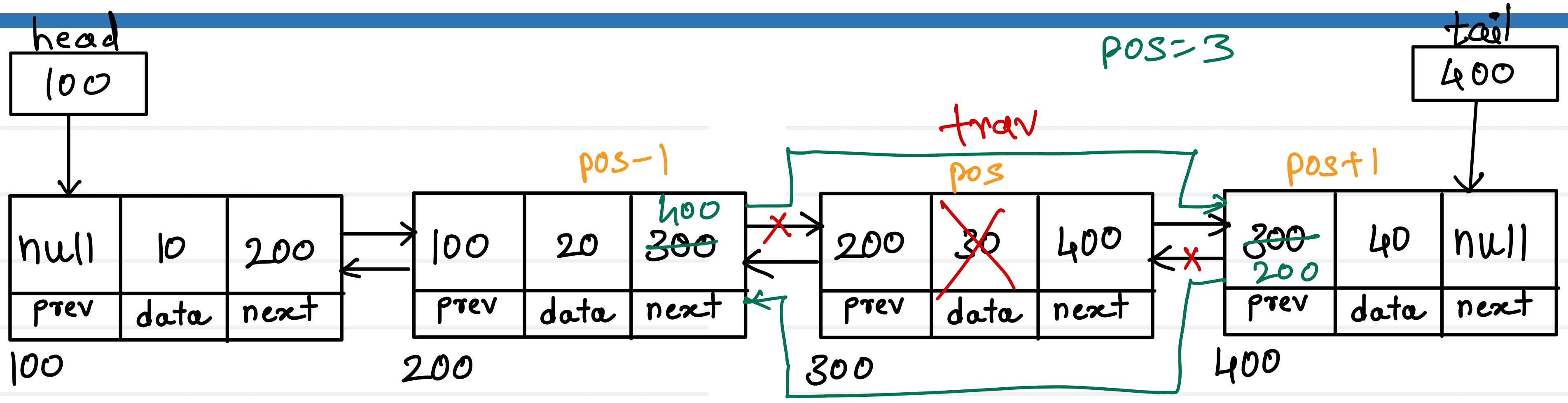
Doubly Linear Linked List - Delete last



1. if list is empty, return
2. if list has single node
   head = tail = null
3. if list has multiple nodes
   a. make next of second last node equal to null
   b. move tail on second last node

$\boxed{T(n) = O(1)}$

Doubly Linear Linked List - Delete Position



1. if list is empty, return
2. if list has single node, head = tail = null
3. if list has multiple nodes,
   a. traverse till pos node
   b. add pos+1 node into next of pos-1 node
   c. add pos-1 node into prev of pos+1 node

$\boxed{T(n) = O(n)}$

SUNBEAM

Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com