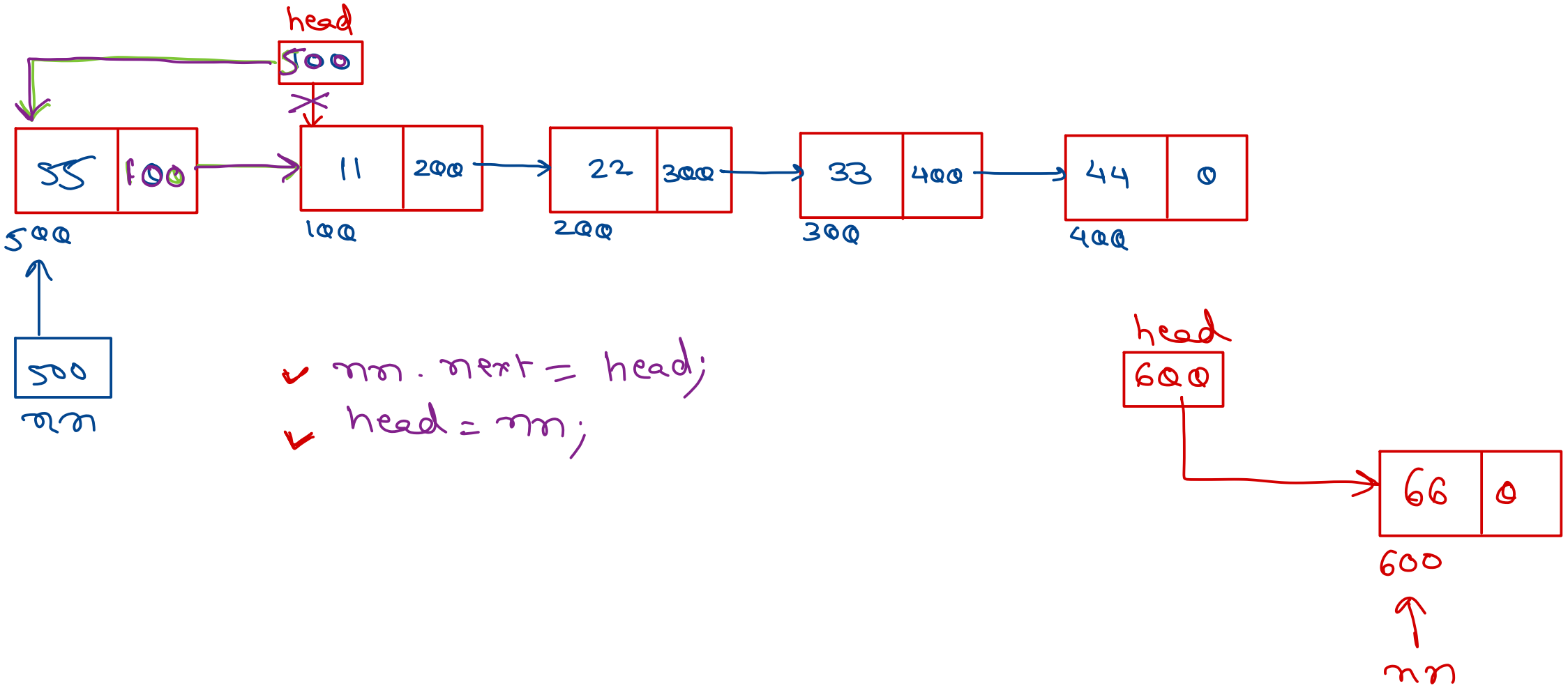
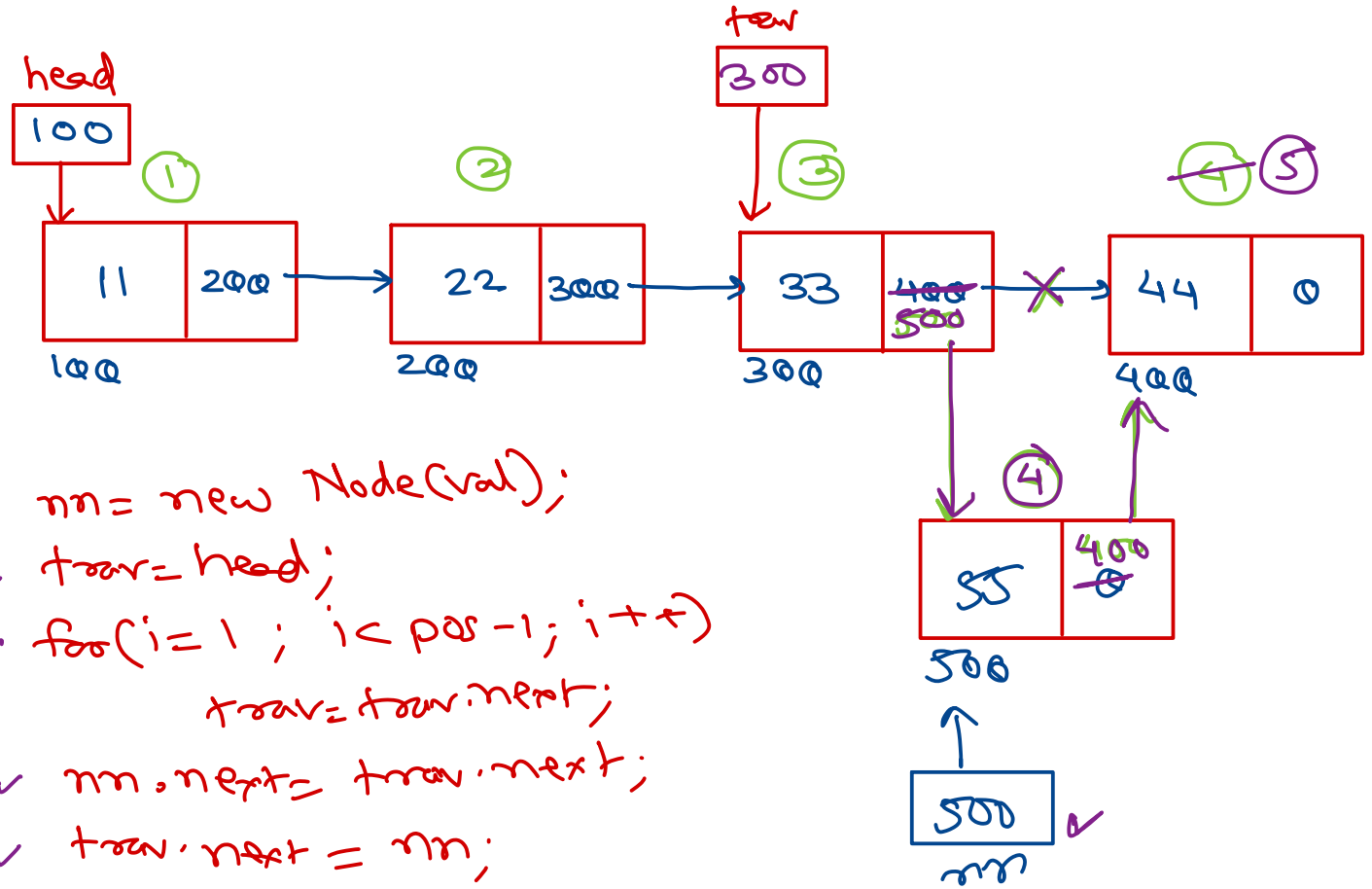
Data Structure & Algorithms

Nilesh Ghule

Singly Linear Linked List - addFirst()



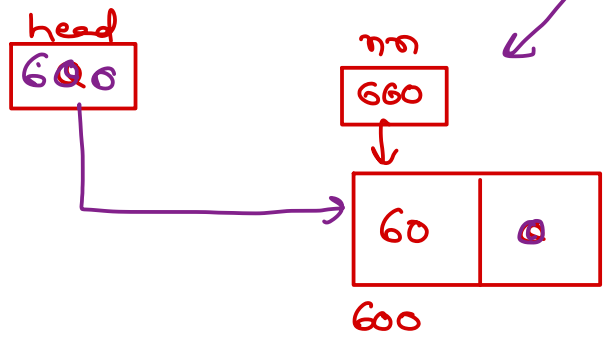
Singly Linear Linked List - addAtPos()



- nn = new Node(val);
- trav = head;
- for(i=1 ; i<pos-1; i++)
  trav = trav.next;
- nn.next = trav.next;
- trav.next = nn;

1. list is empty
2. add node at pos = 1
3. add node at pos < 1

add node at first

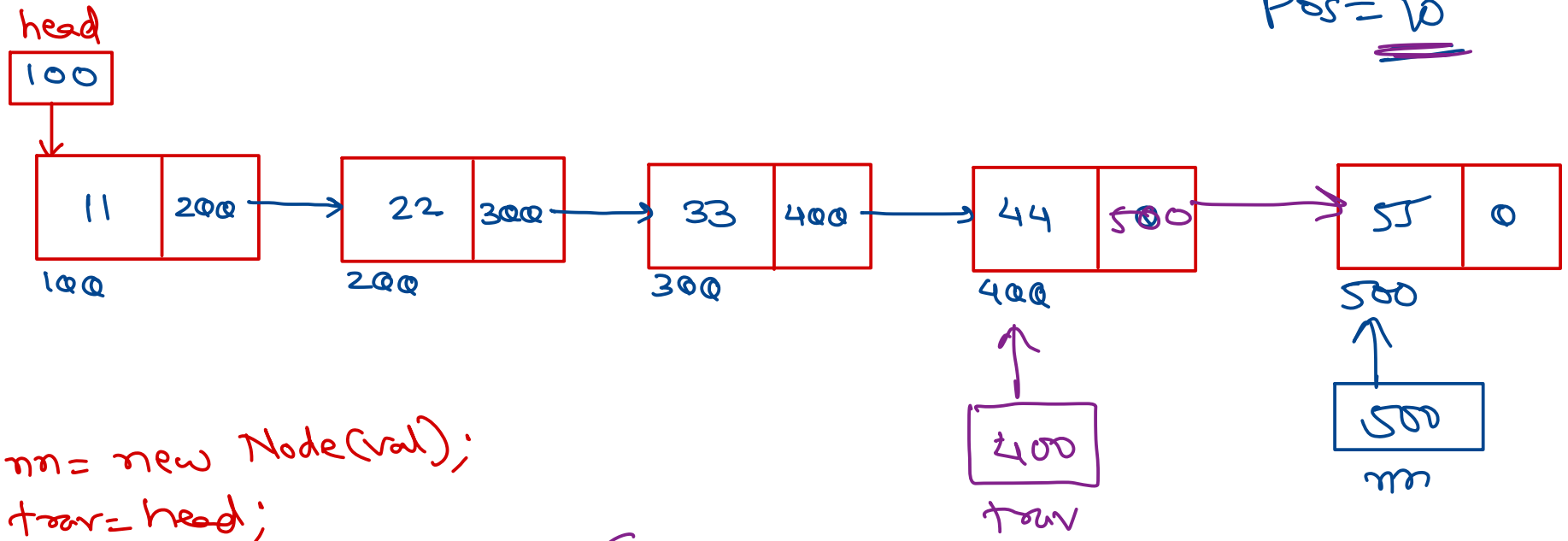


Singly Linear Linked List

- addAtPos()

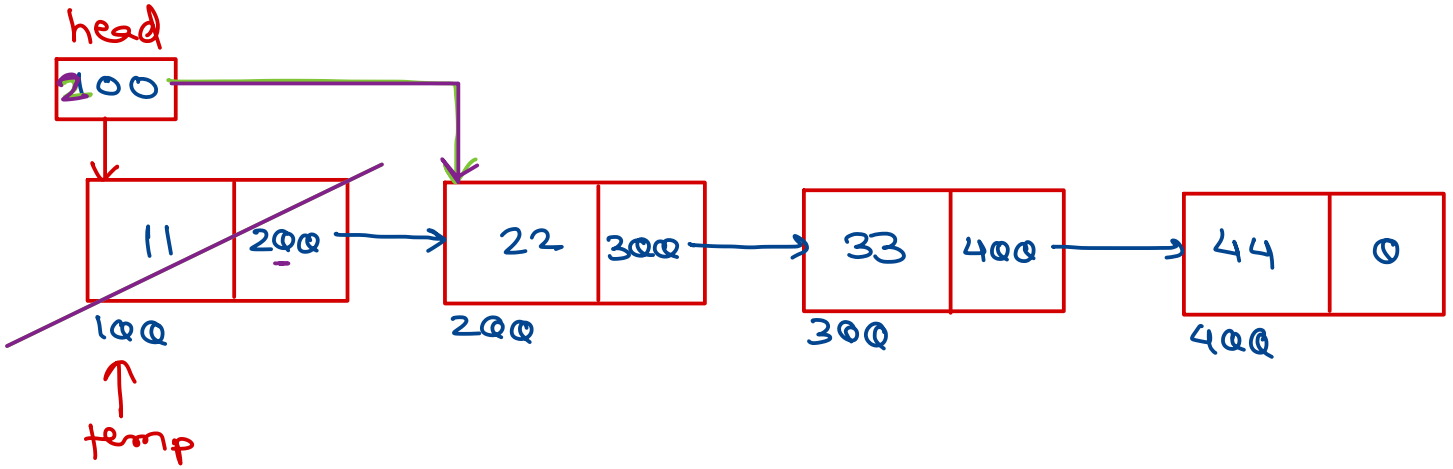
④ add beyond last pos

val = 55
Pos = 10




```
nn = new Node(val);
trav = head;
for(i=1 ; i<pos-1; i++){
    if(trav.next==null)
        break;
    trav = trav.next;
}
nn.next = trav.next;
trav.next = nn;
```

Singly Linear Linked List - delFirst()



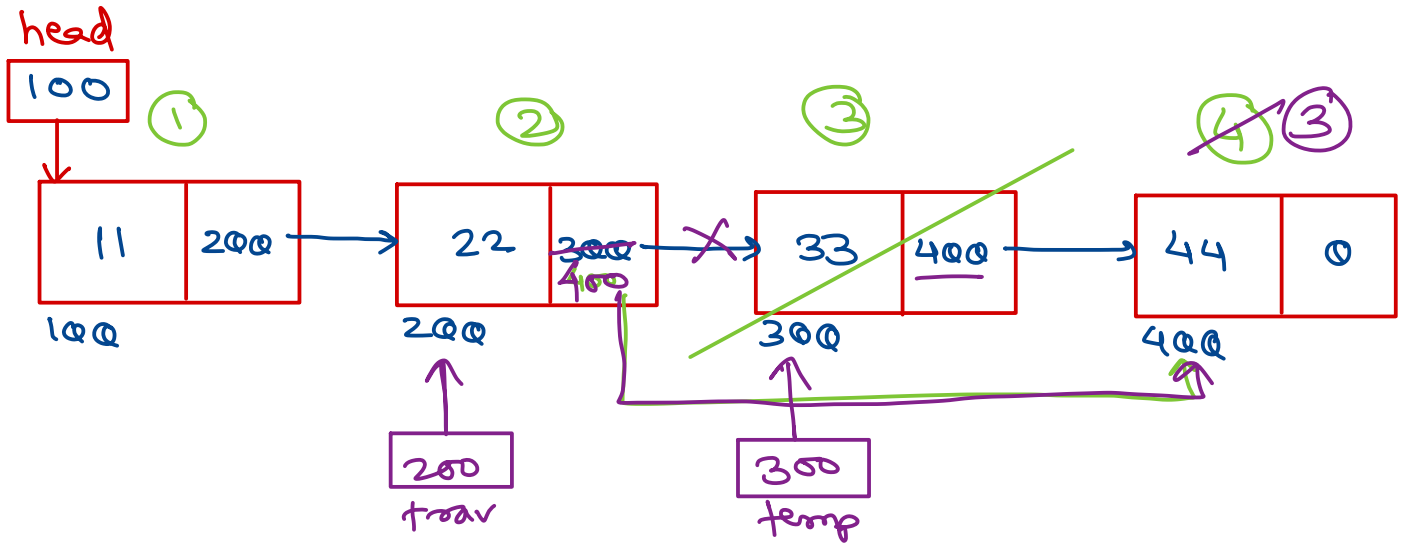

```
temp = head; //x in Java/Python
head = head.next;
delete temp; (G.C)
```

head
0

① if list is empty.

```
if(head != null)
    head = head.next;
```

Singly Linear Linked List - delAtPos()

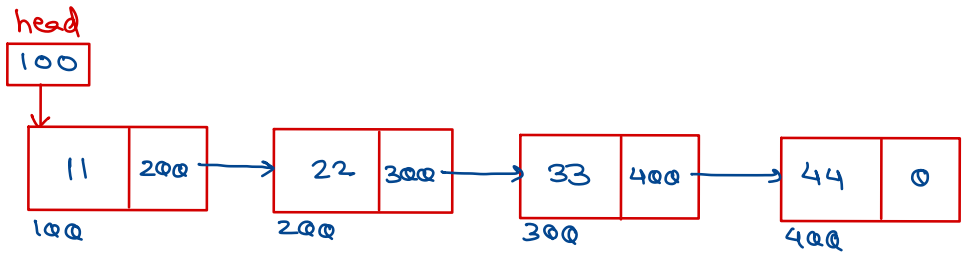


1. list is empty
2. delete at pos 1

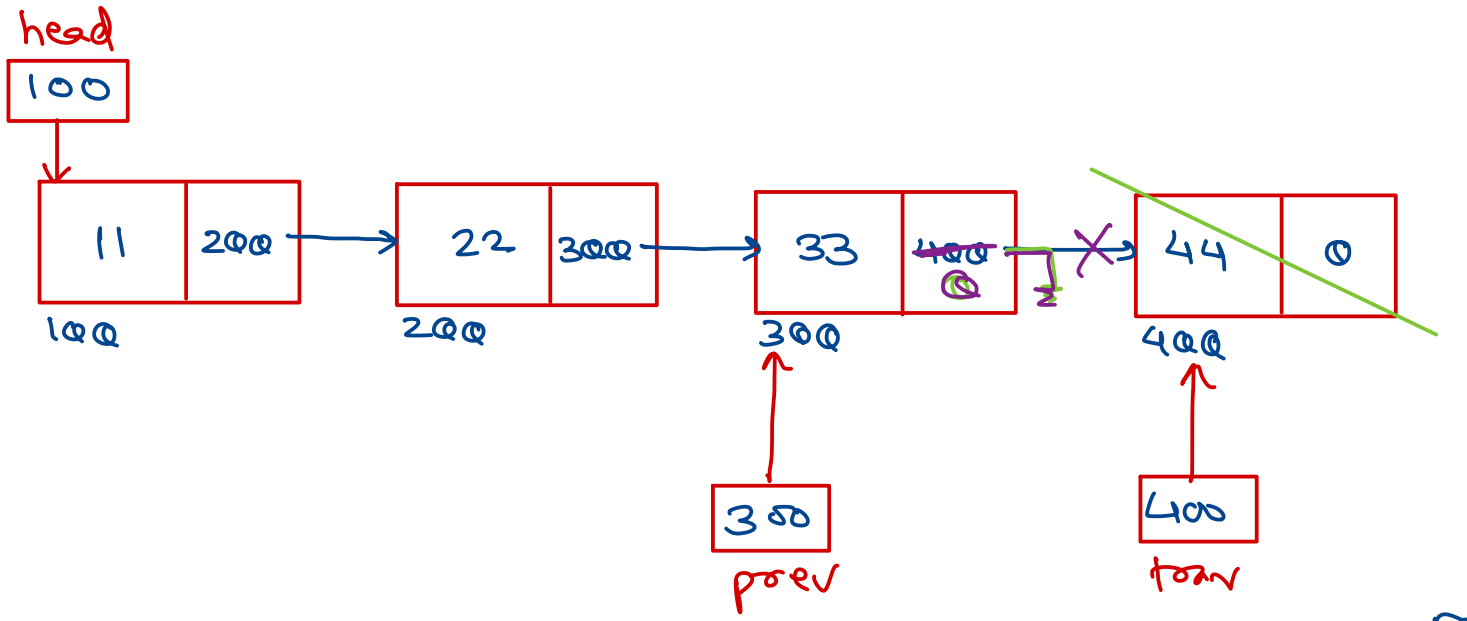
} call delFirst()

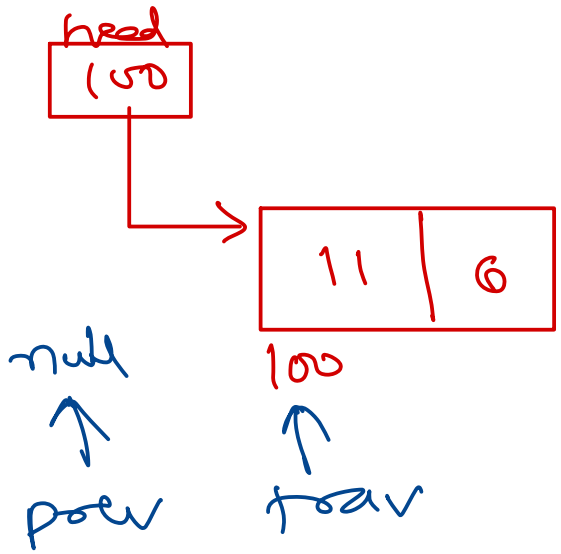
3. delete at pos < 1
   - give error / do nothing
4. delete pos beyond last ele.
   - give error / do nothing

```
trav = head;
for( i=1; i<pos-1; i++) {
    if(trav.next == null)
        return; //do nothing
    trav = trav.next;
}
temp = trav.next;
trav.next = temp.next;
//delete temp; → (GC).
```



Singly Linear Linked List - delLast()





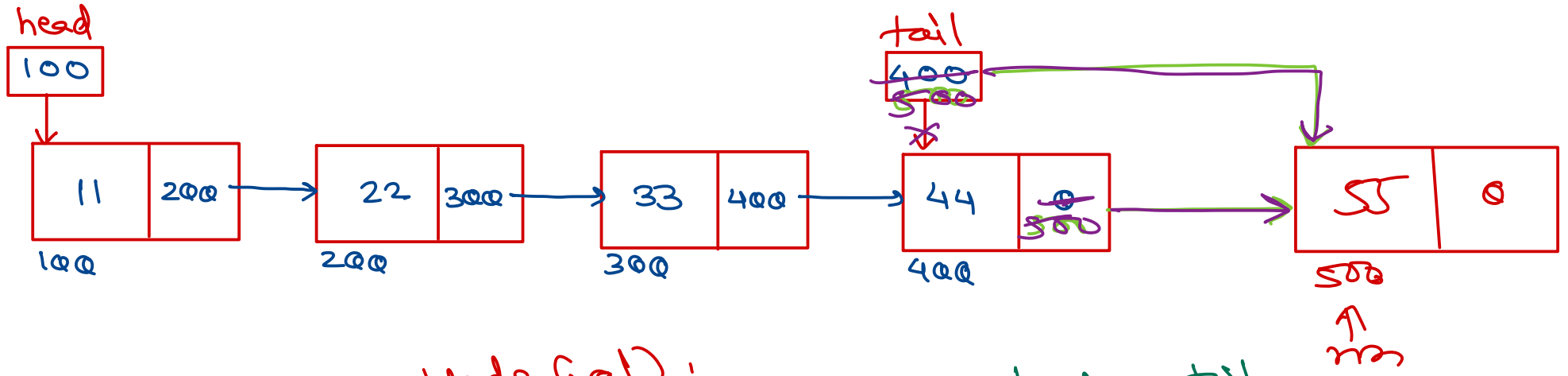
① if list is empty, then do nothing.

② If list has only one element, delete that ele.

```
if(head.next==null)
    head=null;
```

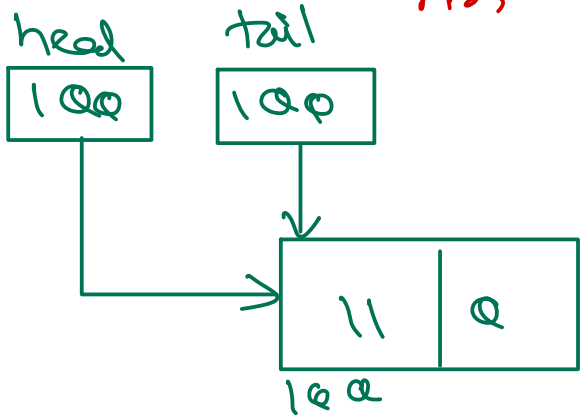
```
prev=null;
trav=head;
while(trav.next!=null){
    prev=trav;
    trav=trav.next;
}
prev.next=null;
```

Singly Linear Linked List – addLast() → O(1)

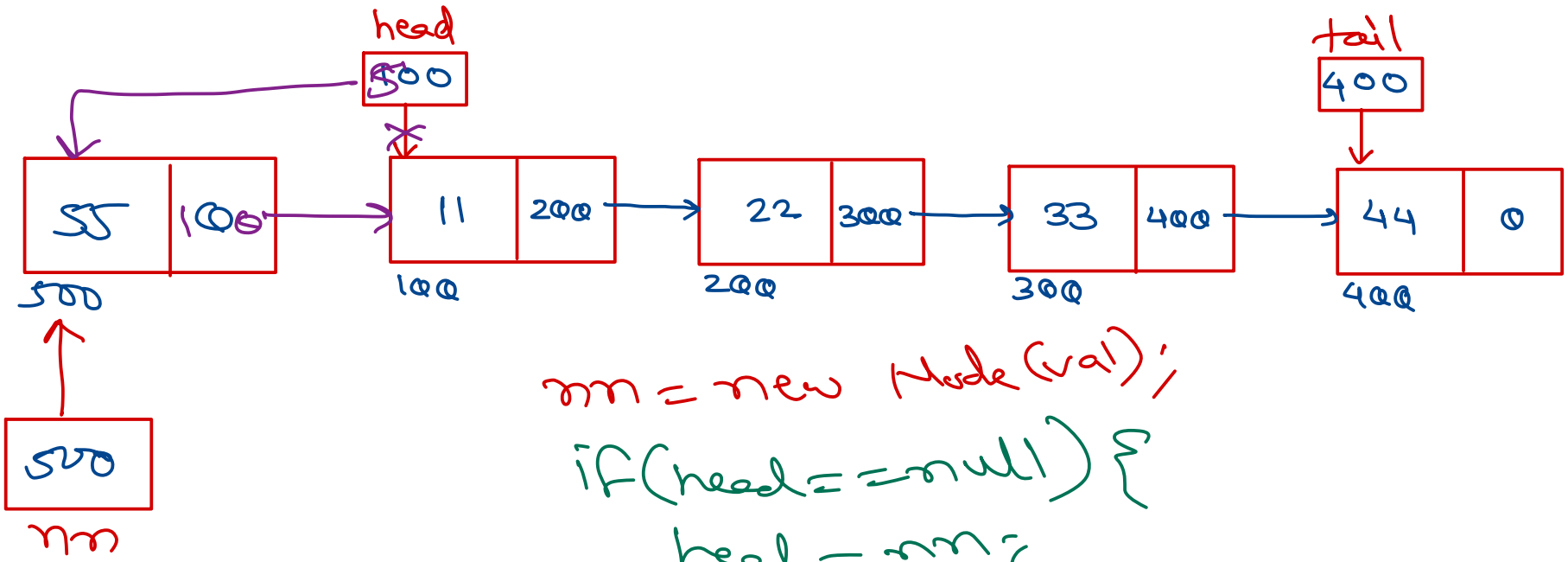



```
class List {
    Node head;
    Node tail;
    =
}
```

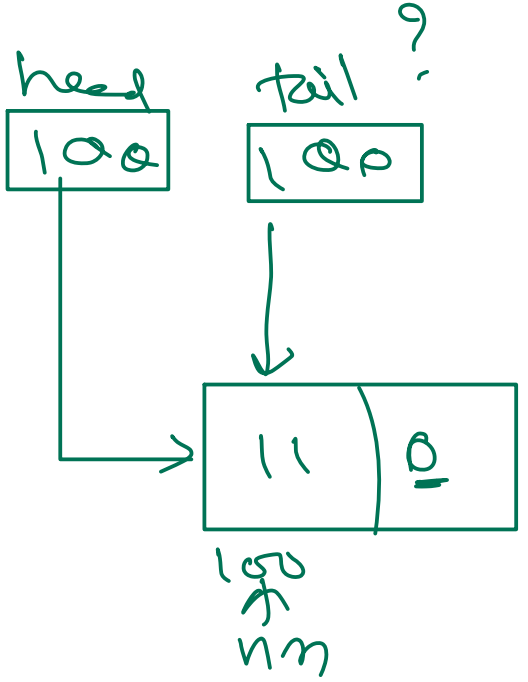
```
nn = new Node(val);
if(head==null) {
    head=nn;
    tail=nn;
}else {
    tail.next=nn;
    tail=nn;
}
```



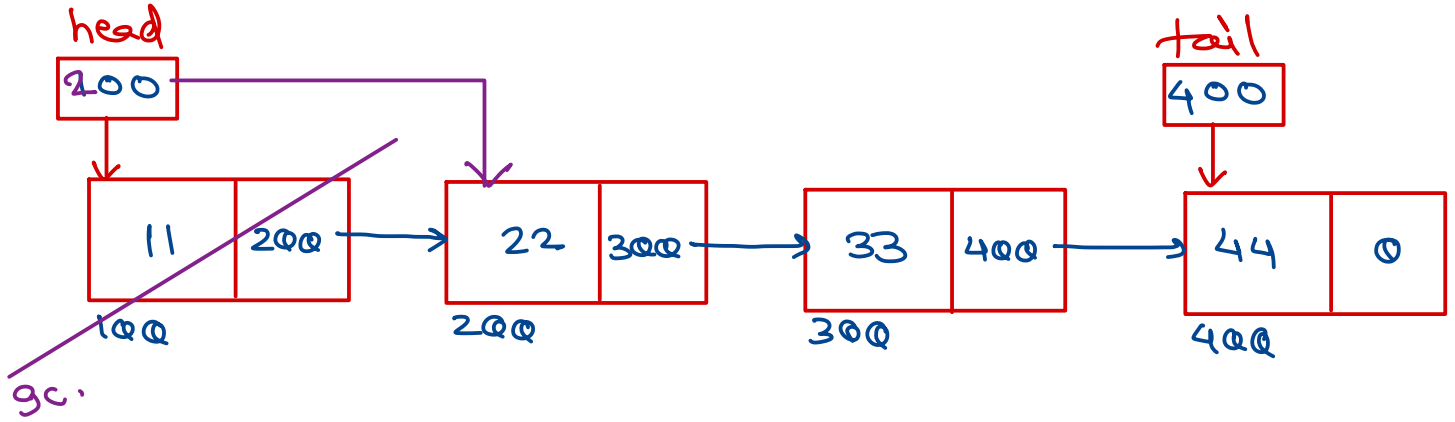
Singly Linear Linked List – addFirst() → O(1)

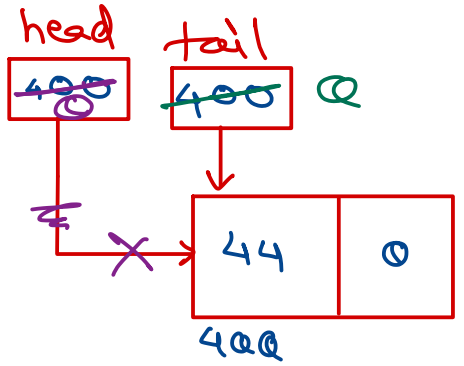



```
nn = new Node(val);
if(head==null){
    head = nn;
    tail = nn;
}else{
    nn.next = head;
    head = nn;
}
```

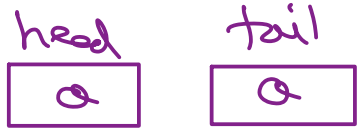


Singly Linear Linked List - delFirst() - O(1)



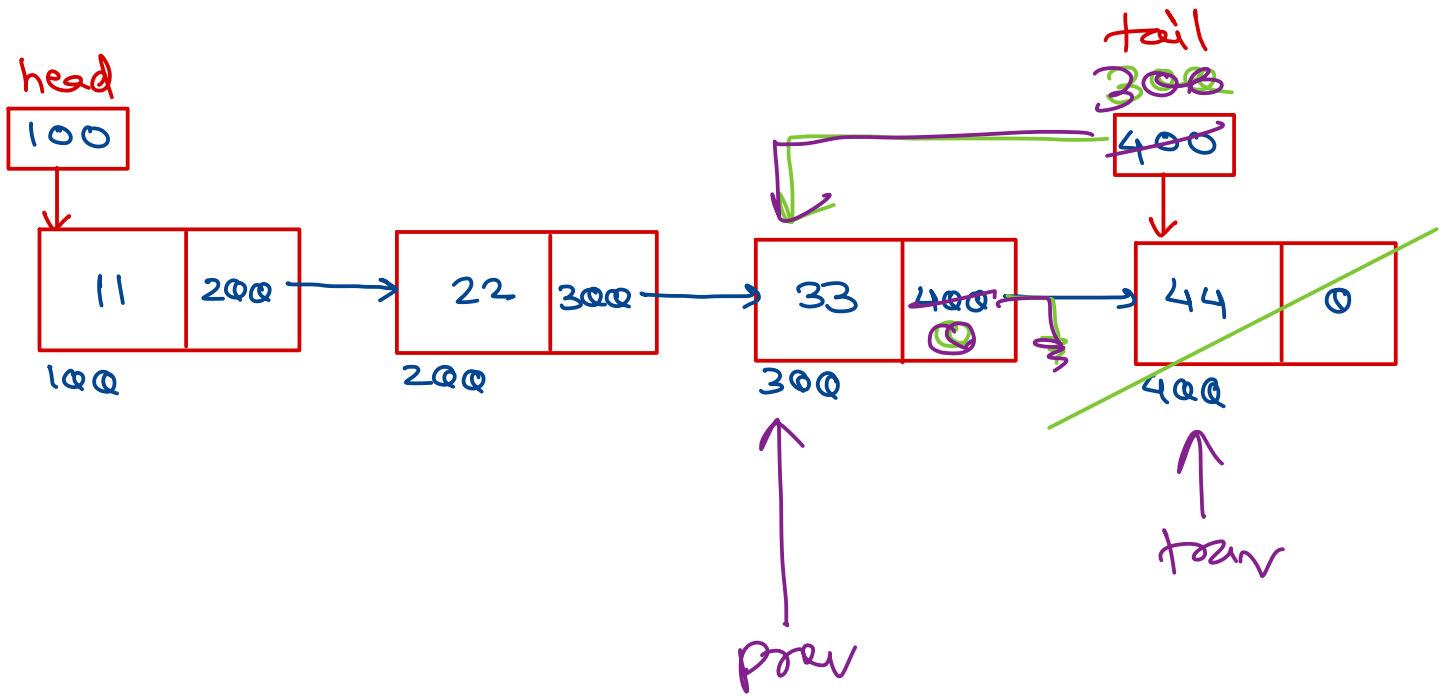



```
if(head != null)
    head = head.next;
if(head==null)
    tail = null;
```



deleted node was the last node.

Singly Linear Linked List - del Last() - O(n)



Stack / Queue using Linked List

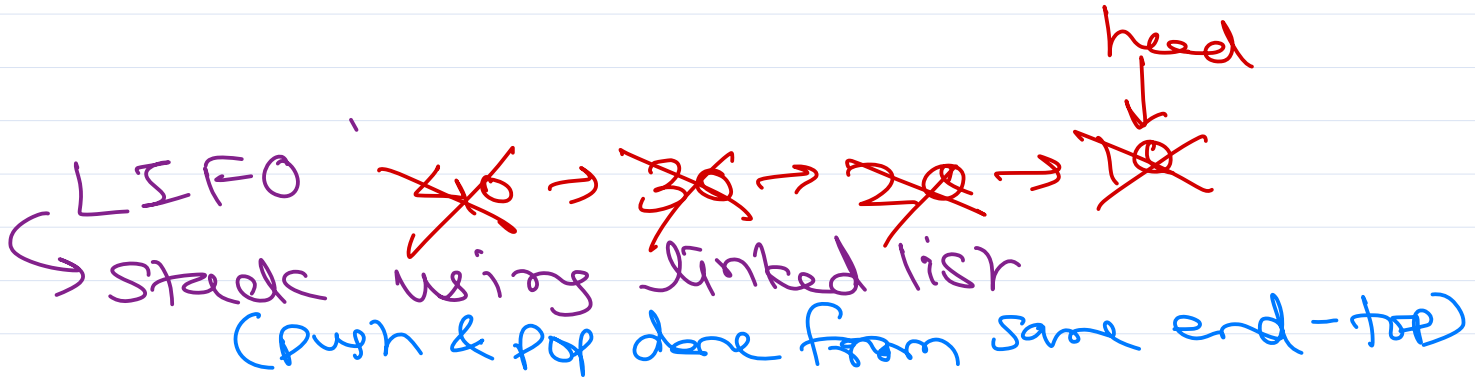
- Stack can be implemented using linked list.
  - add first
  - delete first
  - is empty

- Queue can be implemented using linked list.
  - add last
  - delete first
  - is empty

Singly linear linked list with head & tail pointer

addFirst() → O(1)
addLast() → O(1)
delFirst() → O(1)
delLast() → O(n)

addFirst(): 10, 20, 30, 40 → O(1)
delFirst(): → O(1)



addLast(): 10, 20, 30, 40 → O(1)
delFirst(): → O(1)

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>