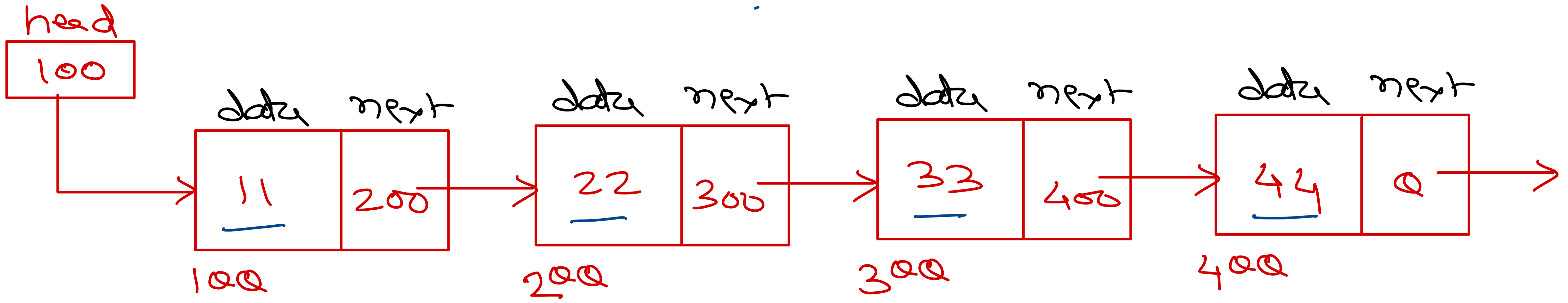
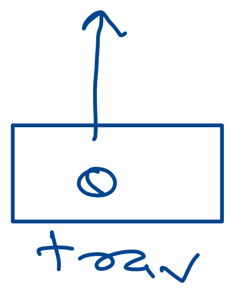
SUNBEAM

Data Structure & Algorithms

Nilesh Ghule

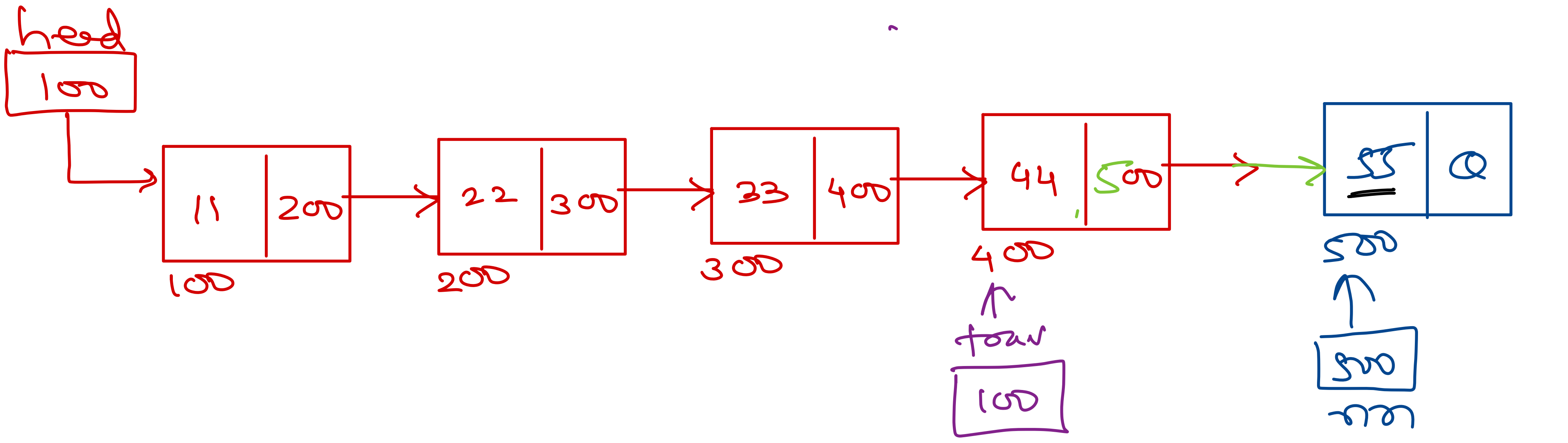
Singly Linear Linked List → display()



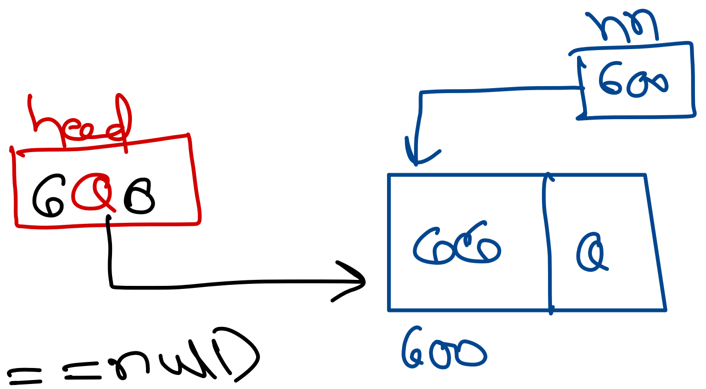


```
trav = head;
while (trav != null)
{
      print(trav.data);
      trav = trav.next;
}
```

Singly Linear Linked List – add Last

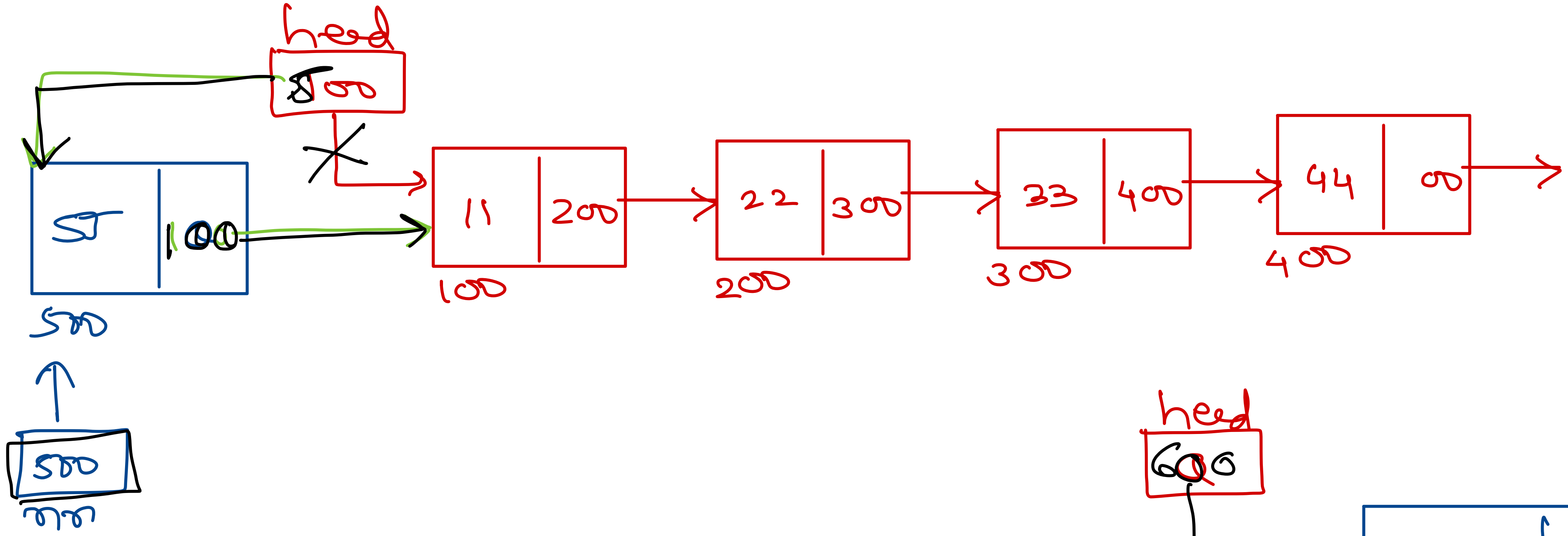


```
① Node nn = new Node(val);
② trav = head;
   while(trav.next != null)
       trav = trav.next;
③ trav.next = nn;
```



```
if(head == null)
    head = nn;
```

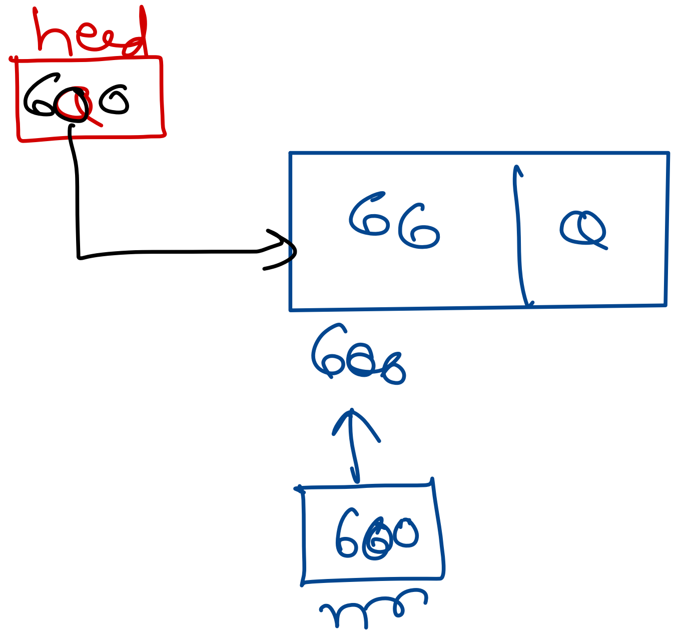
Singly Linear Linked List – add First



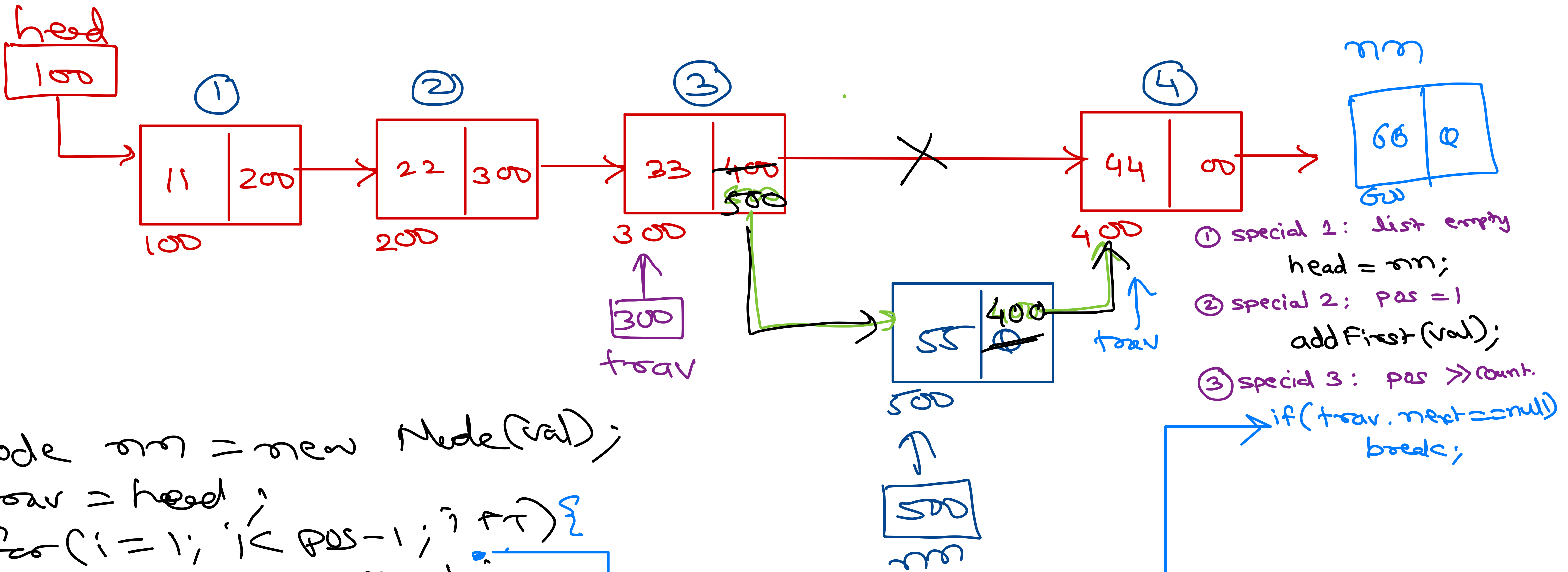
① Node nn = new Node(val);

② nn.next = head;

③ head = nn;



Singly Linear Linked List – addAtPos(55, 4)

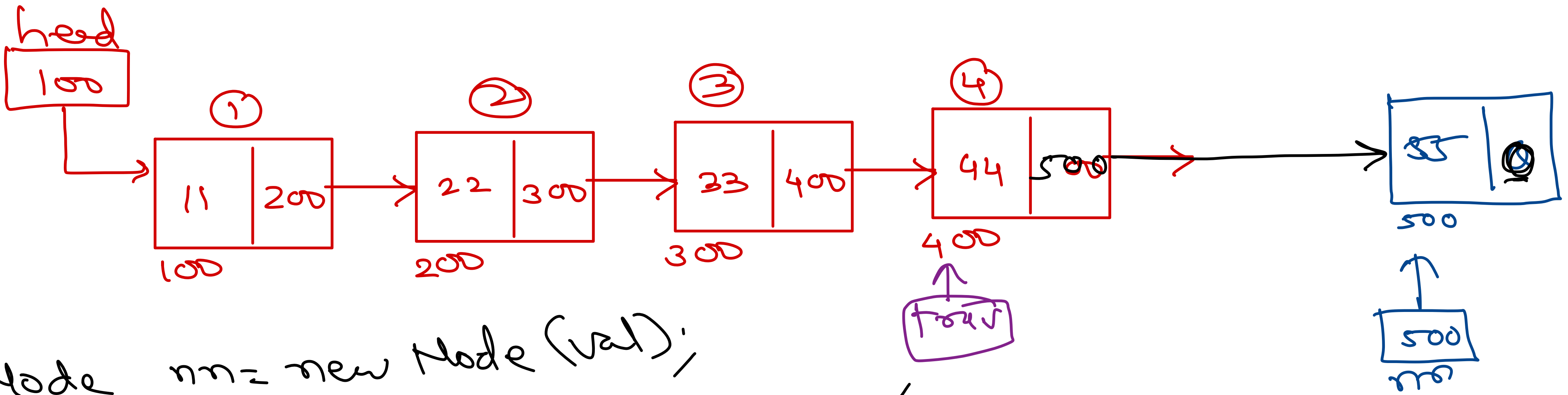


① special 1: list empty
head = nn;
② special 2: pos = 1
addFirst(val);
③ special 3: pos >> count.
if(trav.next==null)
break;

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

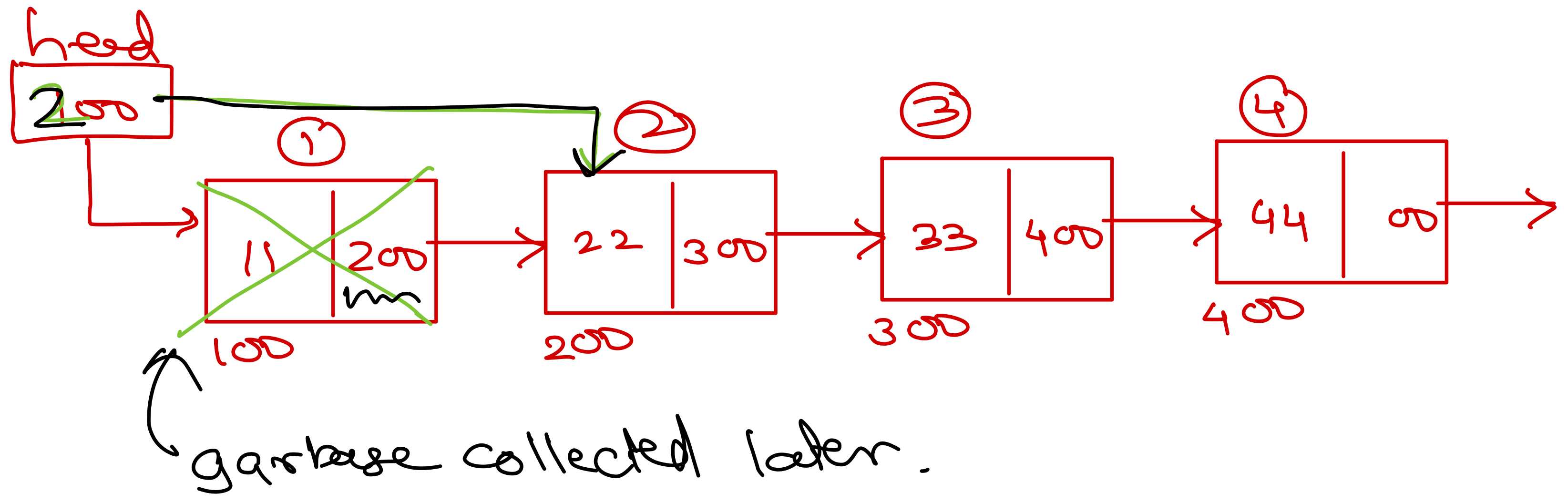
make before break

Singly Linear Linked List – addAtPos(55, 10) ← special 3rd

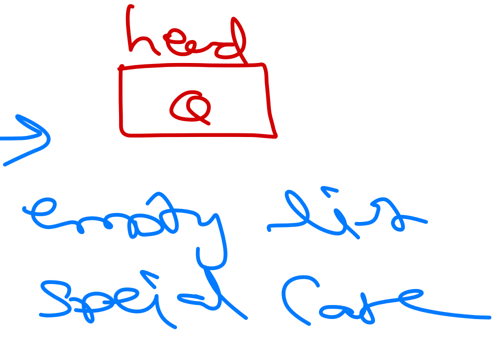


```
Node nn = new Node(val);
Node trav = head;
for(i=1; i<pos-1; i++){
    if(trav.next == null)
        break;
    trav = trav.next;
}
nn.next = trav.next;
trav.next = nn;
```

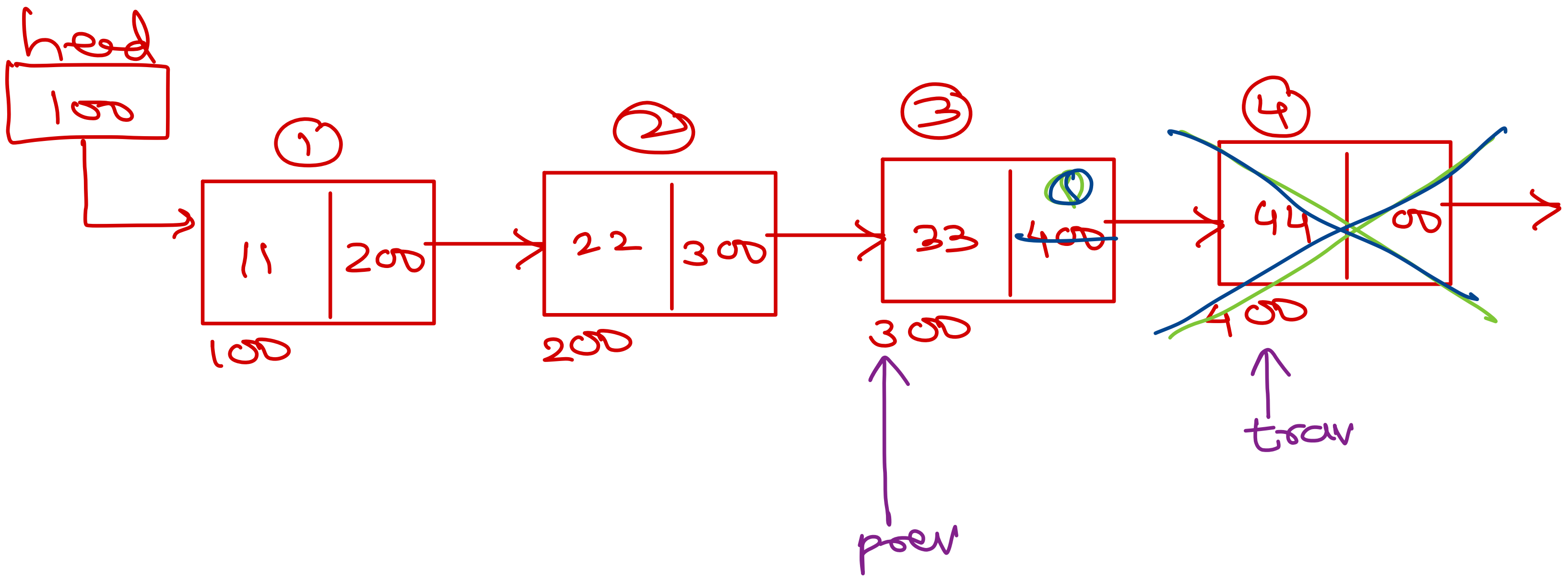
Singly Linear Linked List - delFirst()



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



Singly Linear Linked List – del Last( )

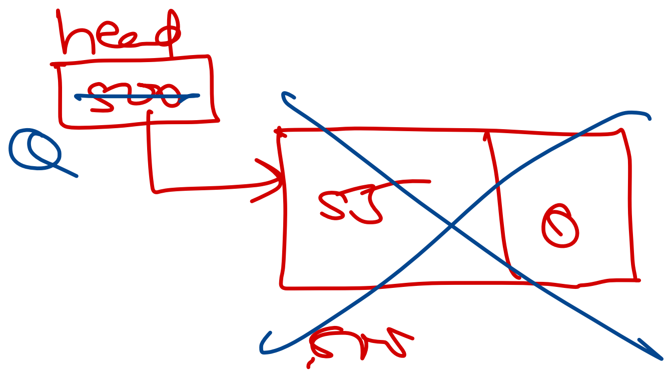


special 1: list empty
- do nothing.

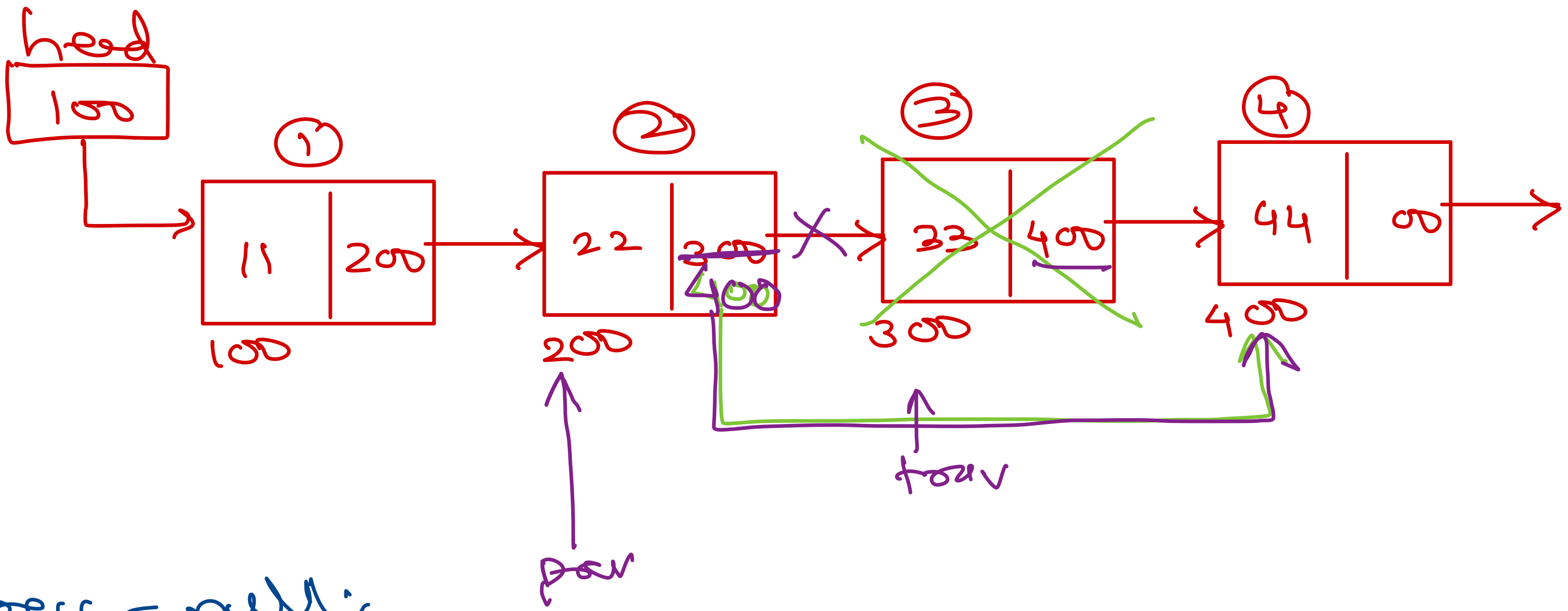
special 2: only one node

```
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 - delAtPos(3)



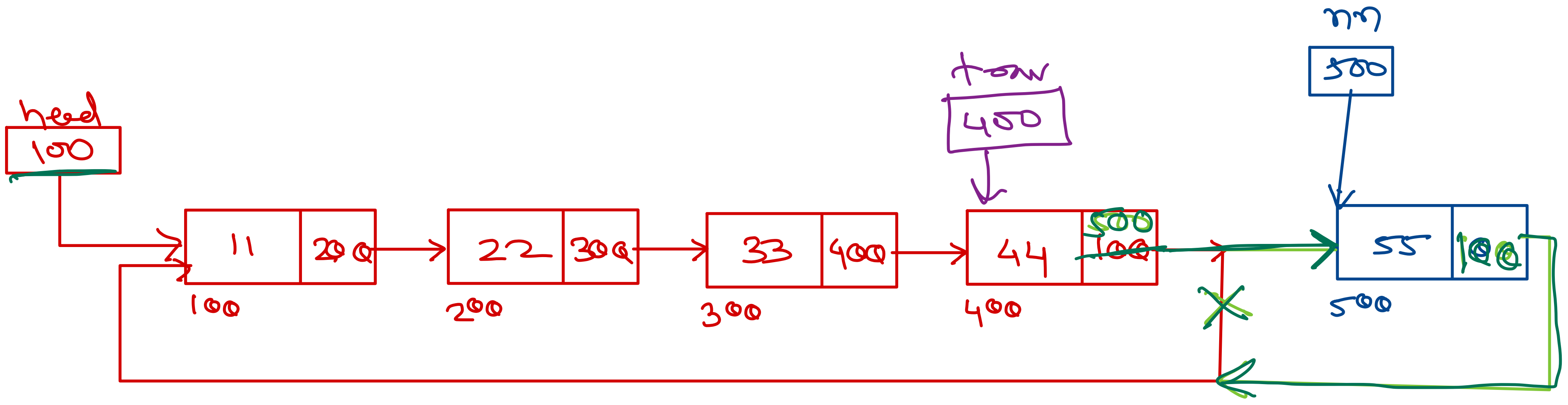
special 1 : list empty
do nothing.

special 2 : pos == 1
delFirst()

special 3 : pos >> count
if(trav == null)
return;

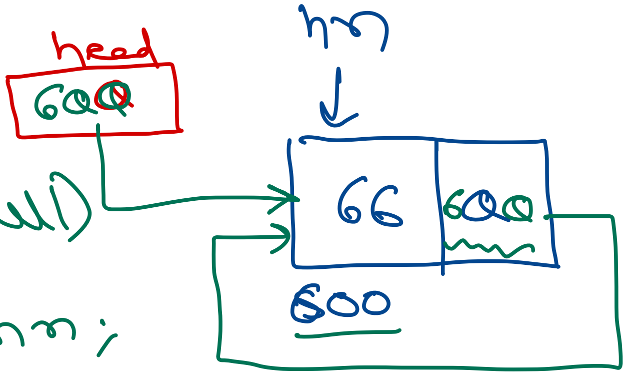
```
prev = null;
trav = head;
for( i=1; i< pos ; i++ ){
    prev = trav;
    trav = trav.next;
}
prev.next = trav.next;
```

Singly Circular Linked List - add Last( )

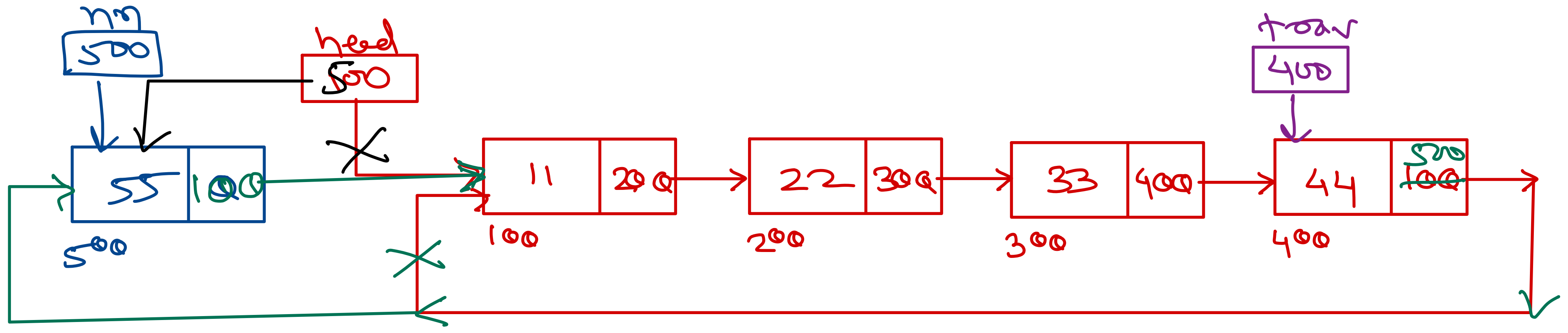


```
Node nn = new Node(val);
trav = head;
while(trav.next != head)
    trav = trav.next;
nn.next = head;
trav.next = nn;
```

```
if(head == null)
{
    head = nn;
    nn.next = head;
}
```

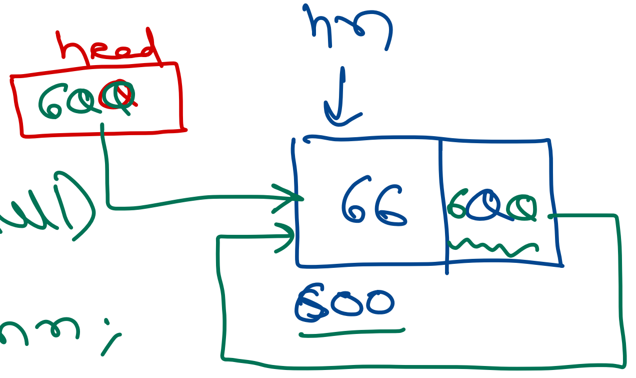


Singly Circular Linked List - addFirst()

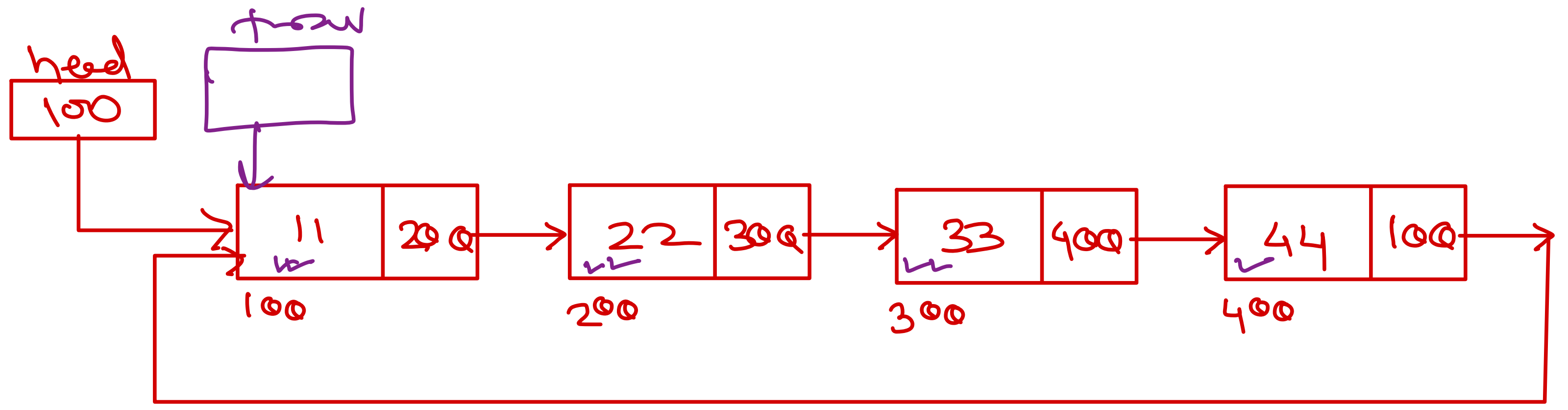


```
① Node nn = new Node(val);
② trav = head;
   while(trav.next != head)
        trav = trav.next;
③ nn.next = head;
④ trav.next = nn;
⑤ head = nn; <-
```

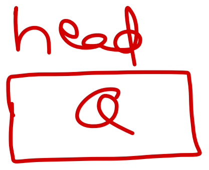
```
if(head==null)
{
     head = nn;
     nn.next = head;
}
```



Singly Circular Linked List - display( )

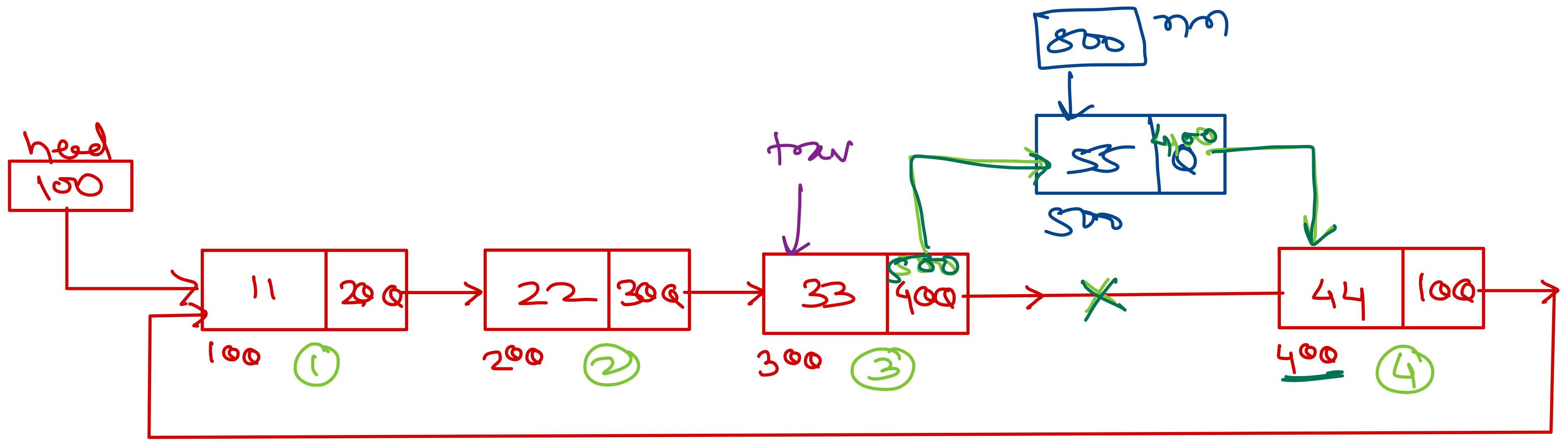


```
trav = head;
do
{
        print(trav.data);
        trav = trav.next;
} while(trav != head);
```



if list is empty,
do nothing

Singly Circular Linked List - add At Pos(55, 4)

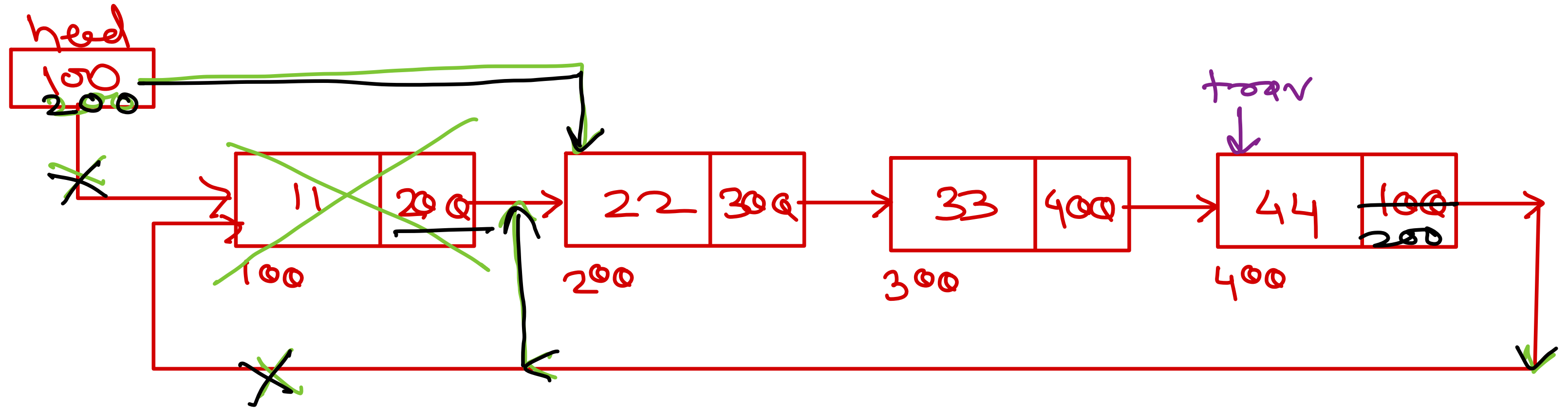


```
Node nn = new Node(val);
trav = head;
for(i = 1; i < pos-1; i++) {
      trav = trav.next;
3
nn.next = trav.next;
trav.next = nn;
```

① special 1 : list empty
    head = nn;
② special 2 ; pos = 1
    addFirst(val);
③ special 3 : pos >> count.

```
if(trav.next == head)
    break;
```

Singly Circular Linked List - delfirst()

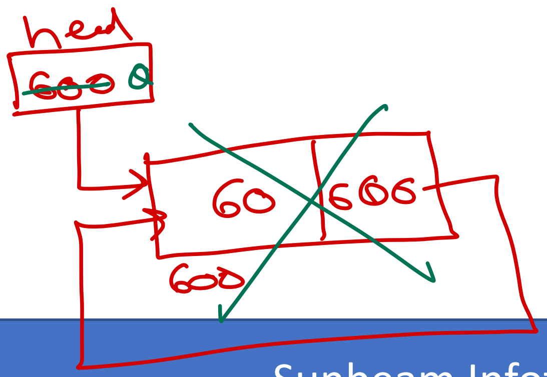


```
① trav = head;
   while(trav.next != head)
       trav = trav.next;
② head = head.next;
③ trav.next = head;
```

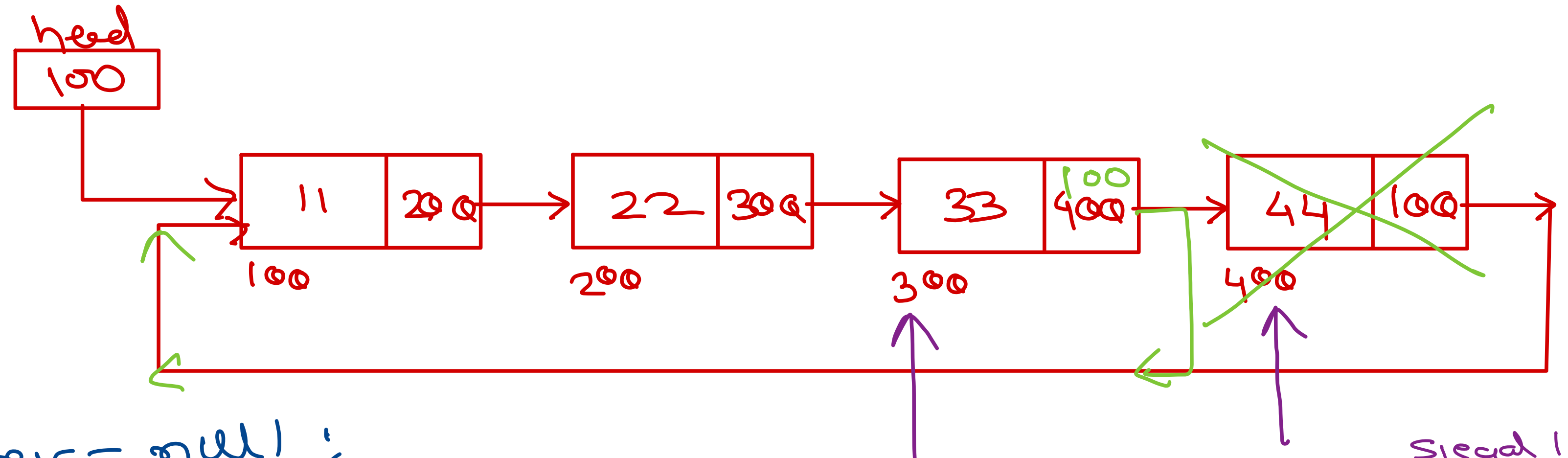
Special 1: list empty, do nothing,

Special 2: list has single ele.

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



Singly Circular Linked List - del Last( )

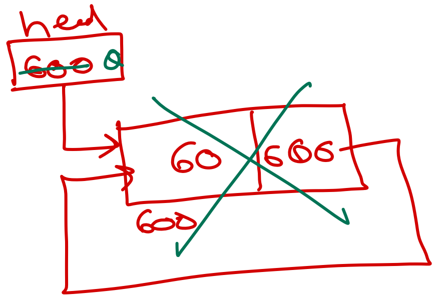


```
prev = null;
trav = head;
while(trav.next != head) {
        prev = trav;
        trav = trav.next;
}
prev.next = head;
```

special 1 : list empty,
do nothing.

special 2 : list has single ele.

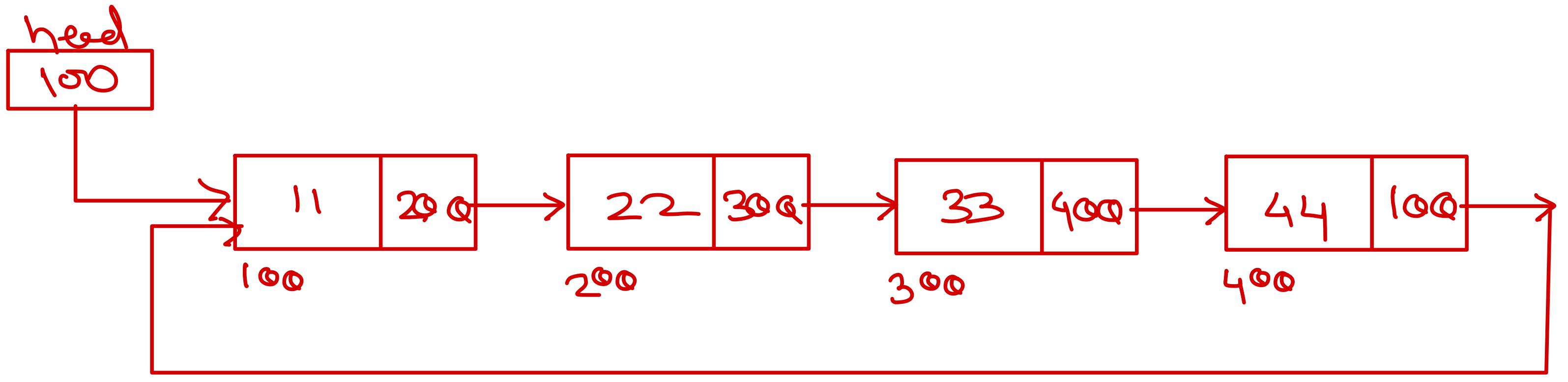
```
if (head.next == head)
    head = null;
```



Singly Circular Linked List

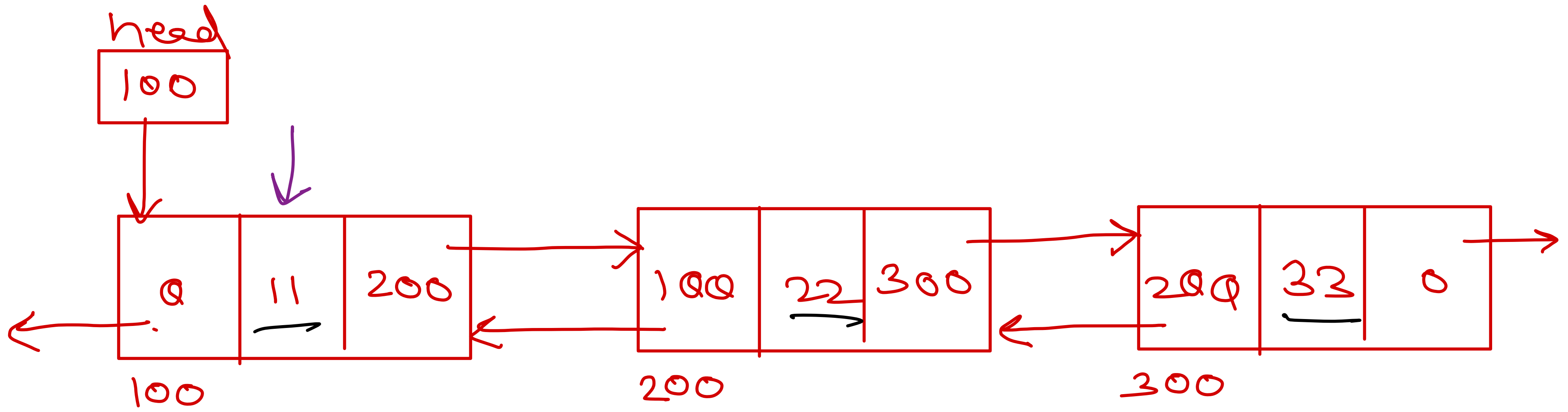
- delAtPos(3)

- Homework.



Doubly Linear Linked List

→ bidirectional traversal. – displayFwd()
displayRev()



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

```
① trav = head;
   while (trav.next != null)
       trav = trav.next;

② while (trav != null)
   {   print (trav.data);
       trav = trav.prev;
   }
```

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>