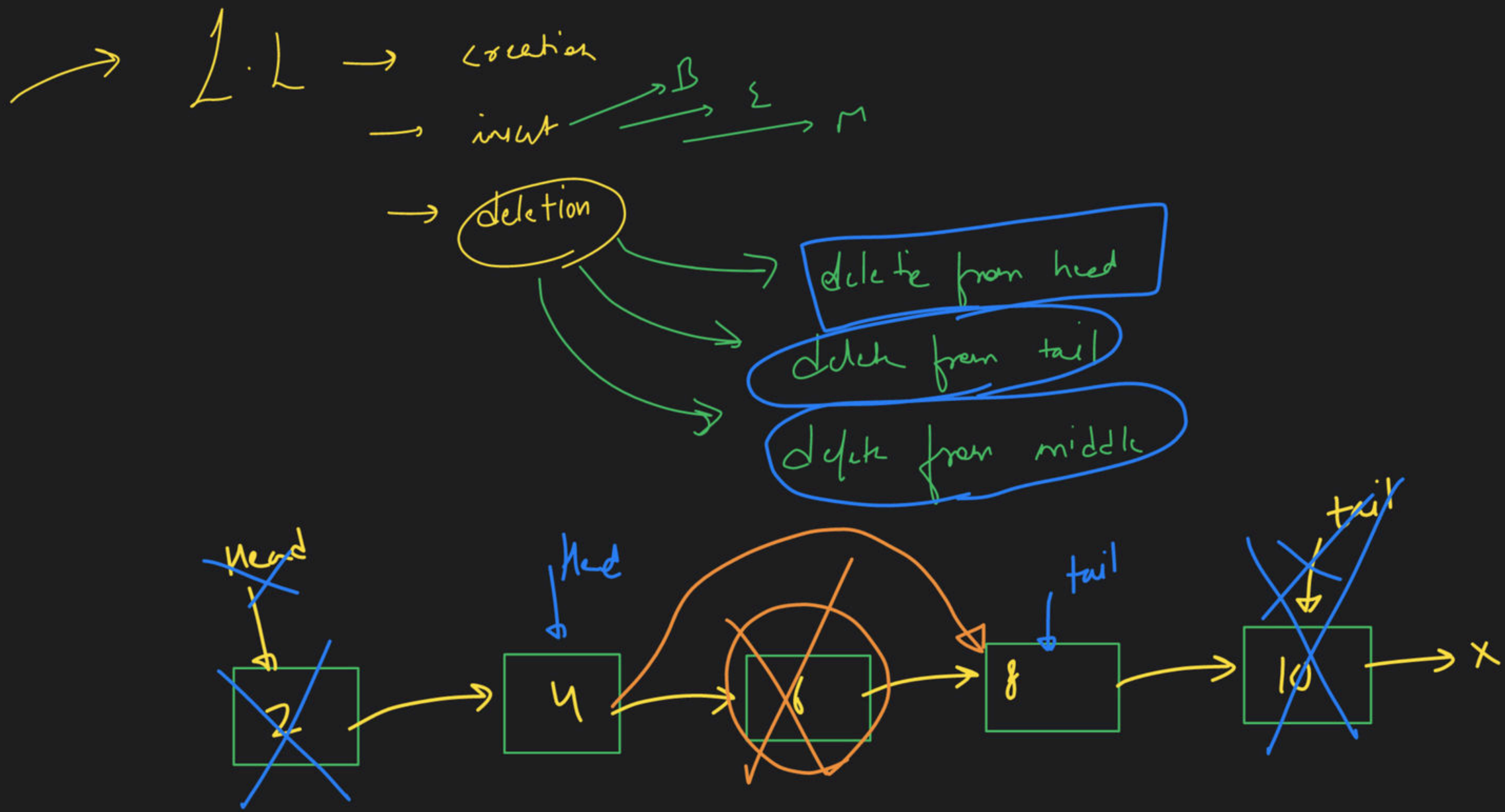
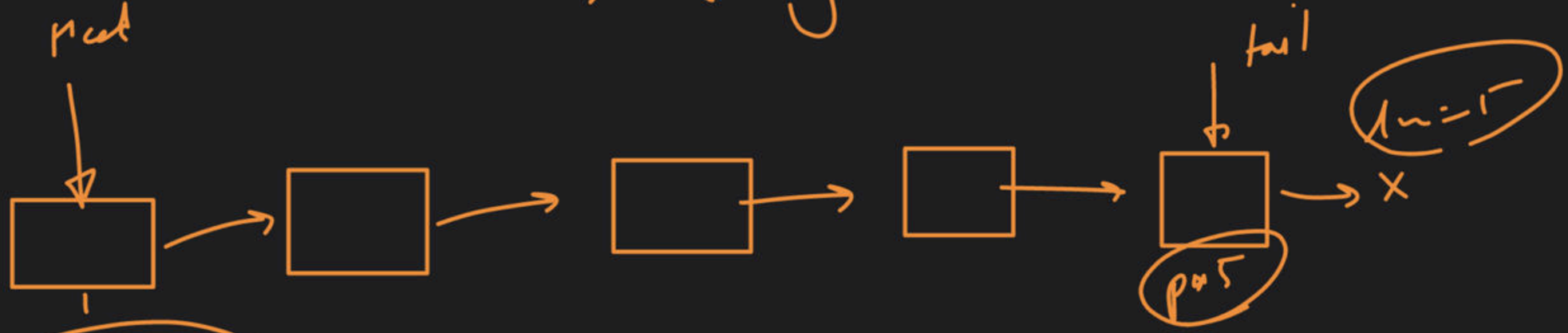
Linked List - Class 2

Special class

Love Babbar • Nov 4, 2023


L.L
creation
insert
B
E
M
deletion
delete from head
delete from tail
delete from middle
head
head
tail
tail
2
4
6
8
10
X

L.L → deletion → head
→ tail
→ middle
→ deleteNode(position)

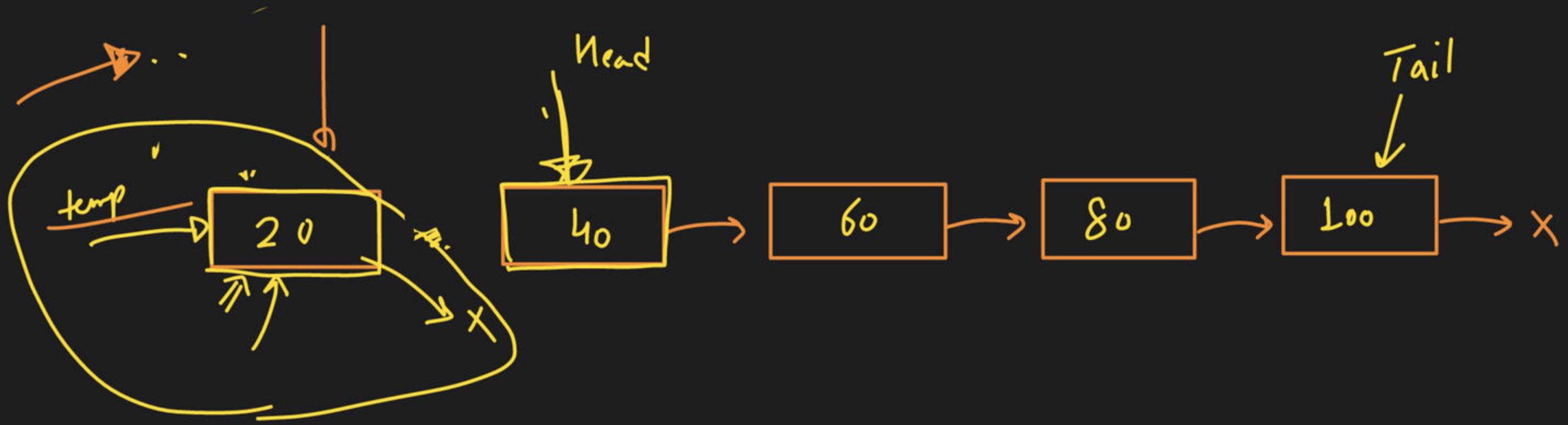


① Empty list → head == NULL

② if (pos == 1) → delete from head

③ if (pos == len) → delete from tail

else → middle

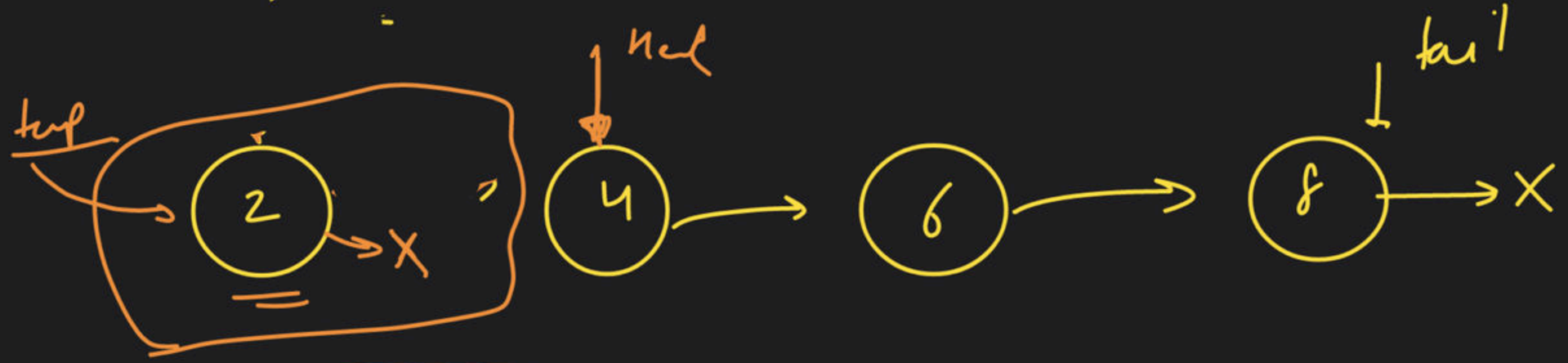


(A) Node * temp = head

(B) head = temp → next

(C) temp → next = NULL

(d) delete temp

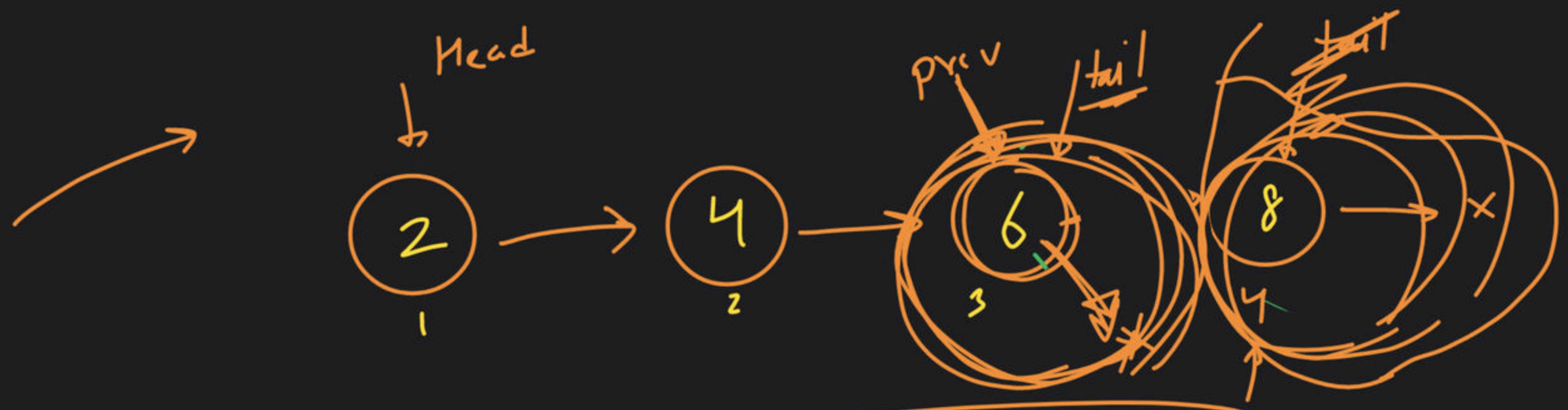


(A) Node * temp = head

(B) head = head -> next;

(C) temp -> next = NULL;

(D) delete temp



pos = 4

I delete Node

II node ki prev node → NULL

III tail update

(A) traverse second last node → prev

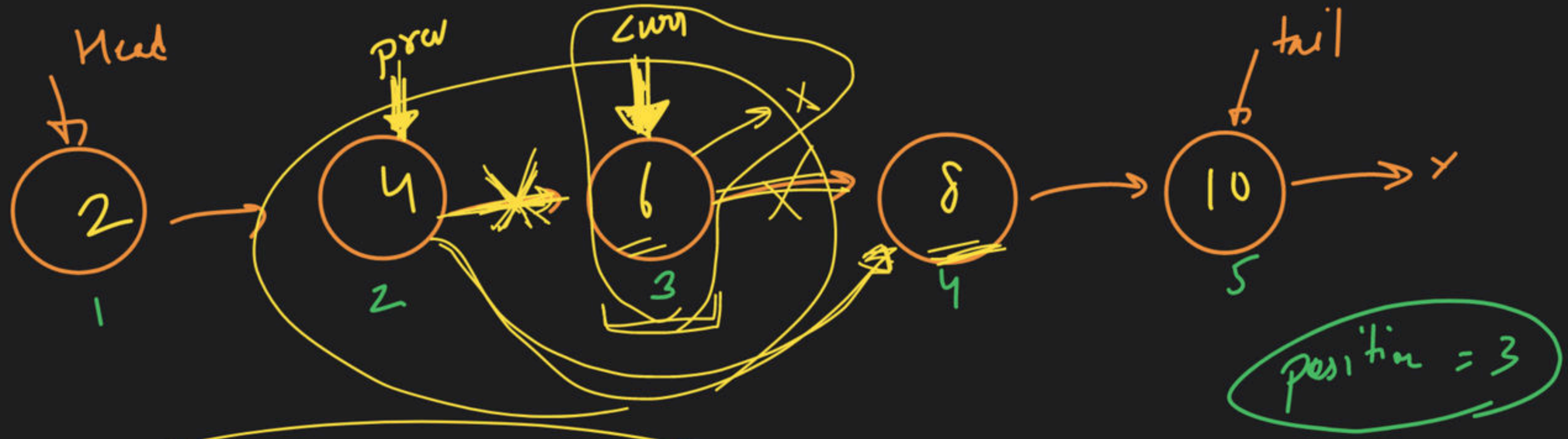
(B) prev → next = NULL

(C) delete tail

(D) tail = prev

tail = prev

delete tail

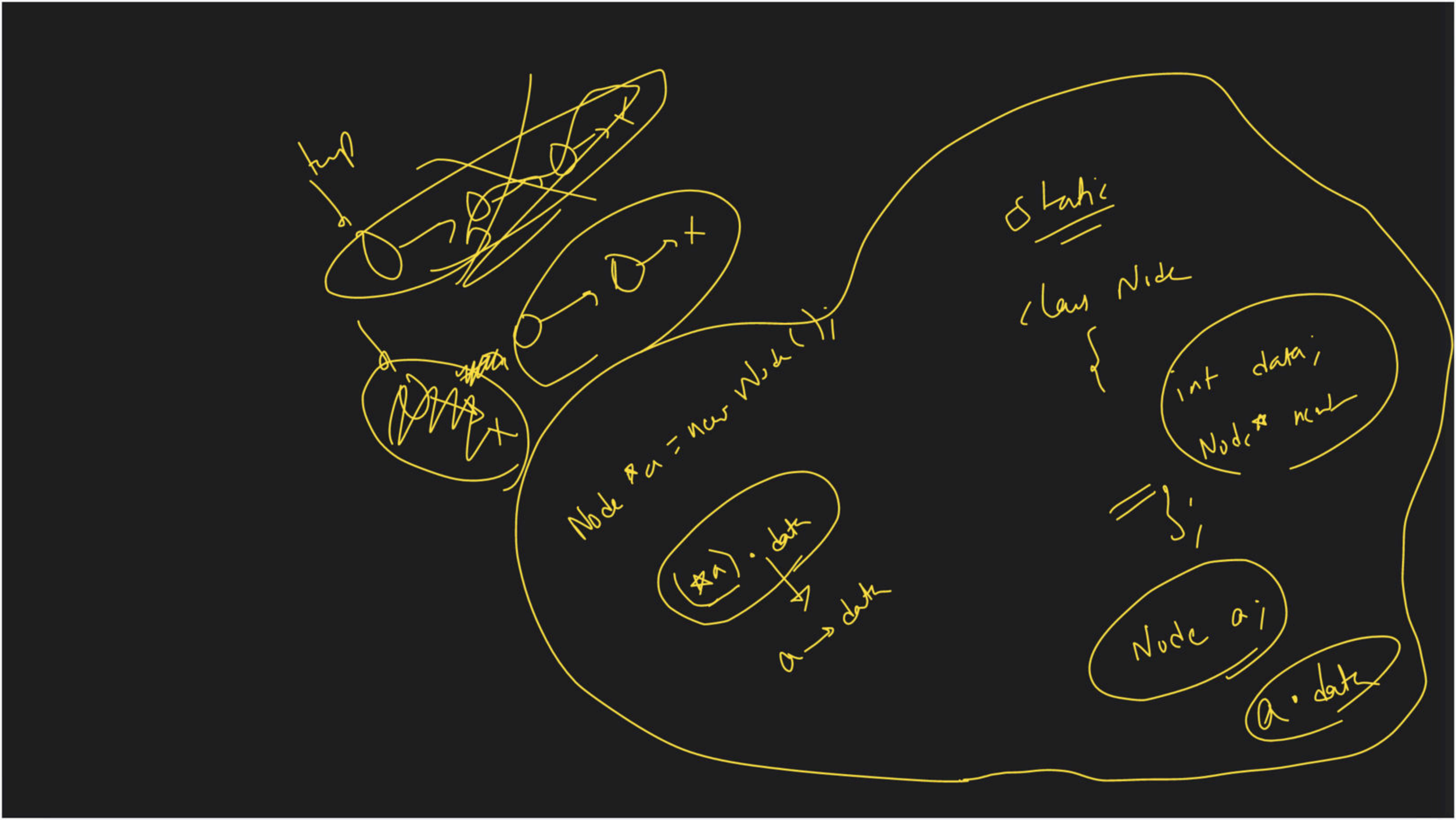


(A) traverse LL for prev / curr

(2) prev -> next = curr -> next

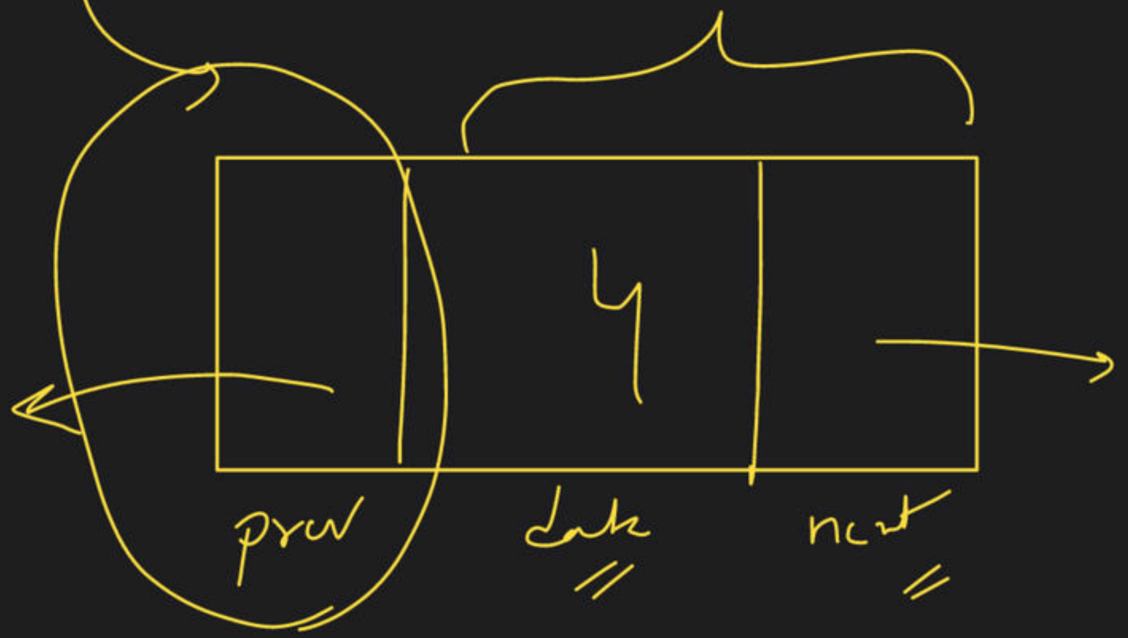
(3) curr -> next = null

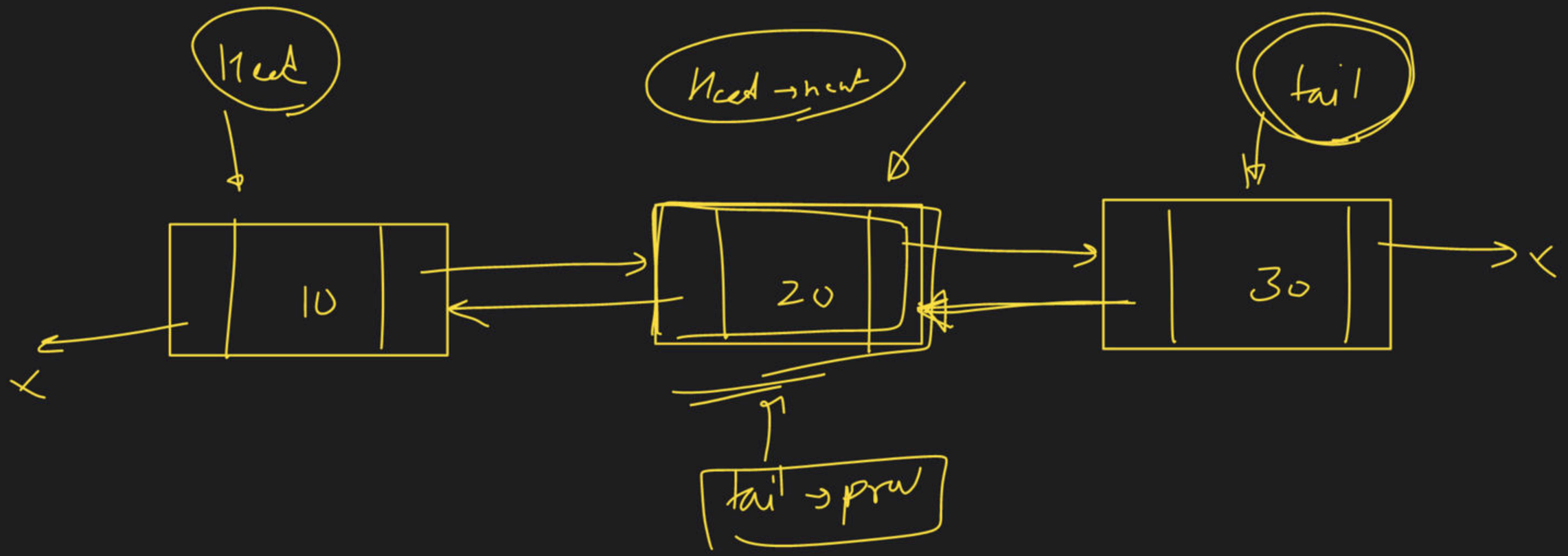
(4) delete curr

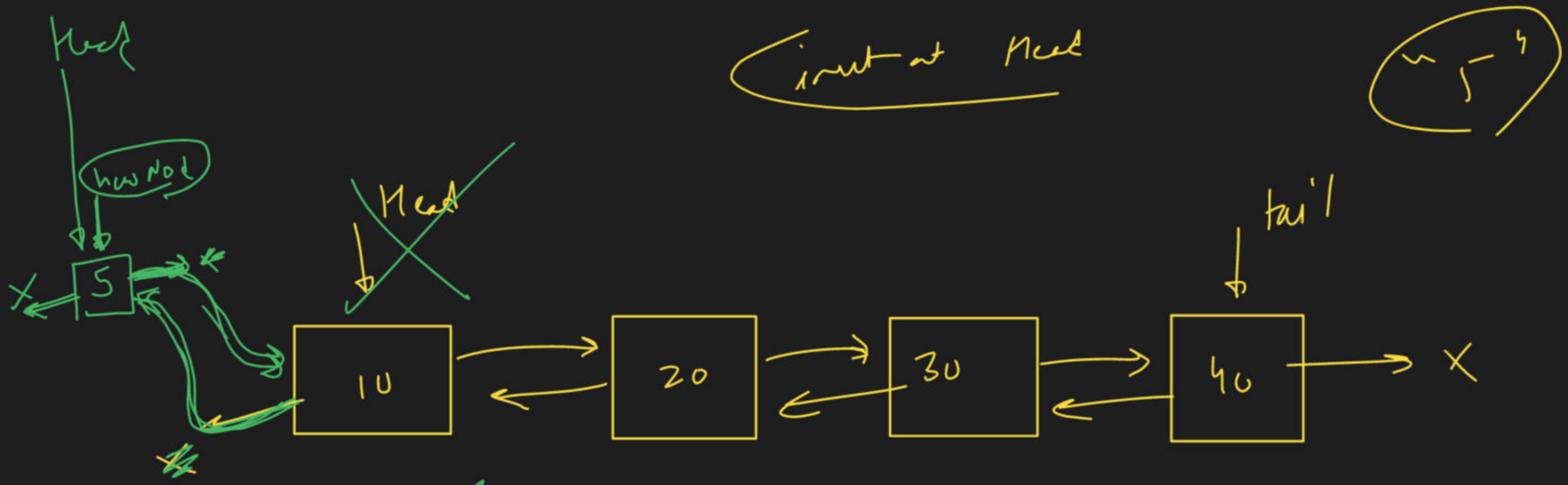

Node *a = new Node();
(*a).data
a->data
static
class Node
{
int data;
Node* next
};
Node a;
a.data

Doubly Linked List

Node




head
head->next
tail
10
20
30
tail->prev



insert at head

head = newNode

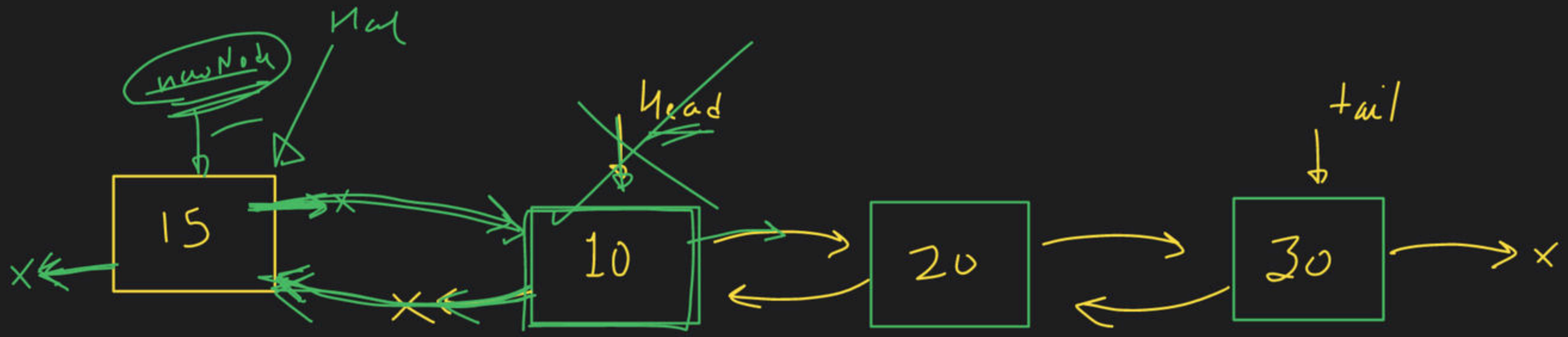
head = head → prev

(A) create node

(B) head → prev = newNode

(C) newNode → next = head

(D) Head = newNode

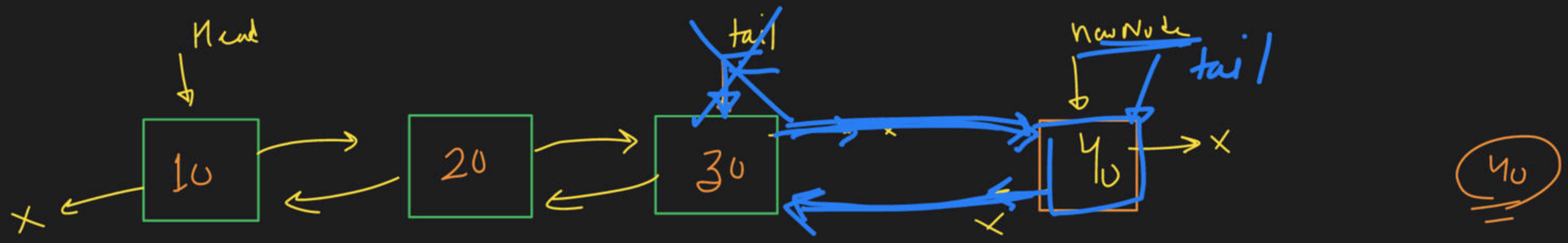


(A) Create Node → Node* newNode = new Node(15);

(B) newNode → next = head;

head → prev = newNode

head = newNode

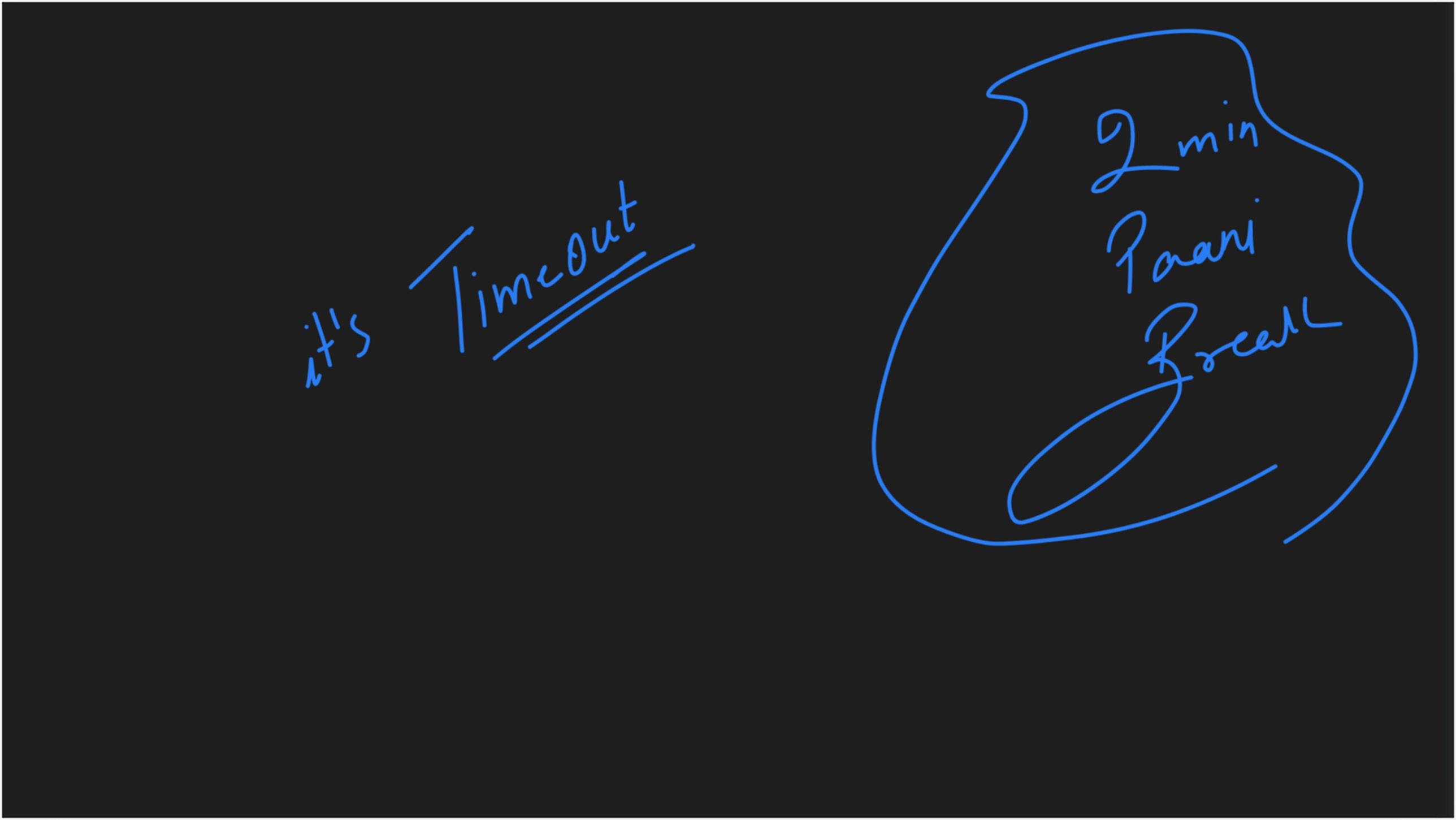


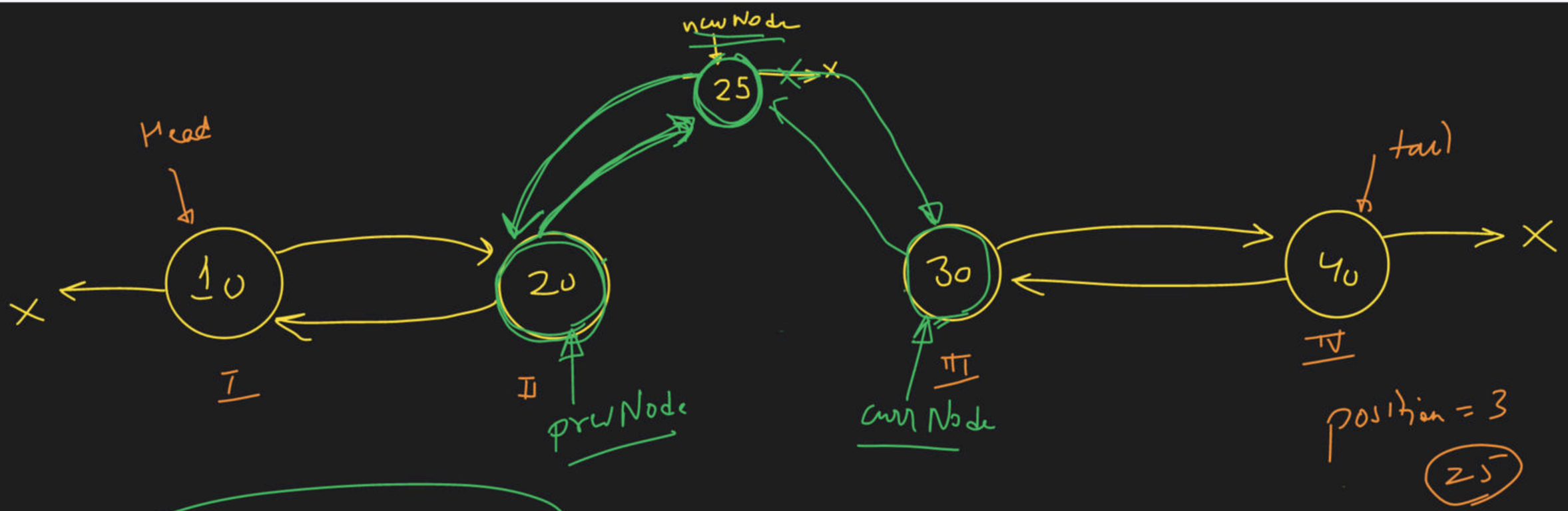
(A) create Node

tail -> next = newNode

newNode -> prev = tail

tail = newNode


it's Timeout
2 min
Paani
Break



(A) Create a Node

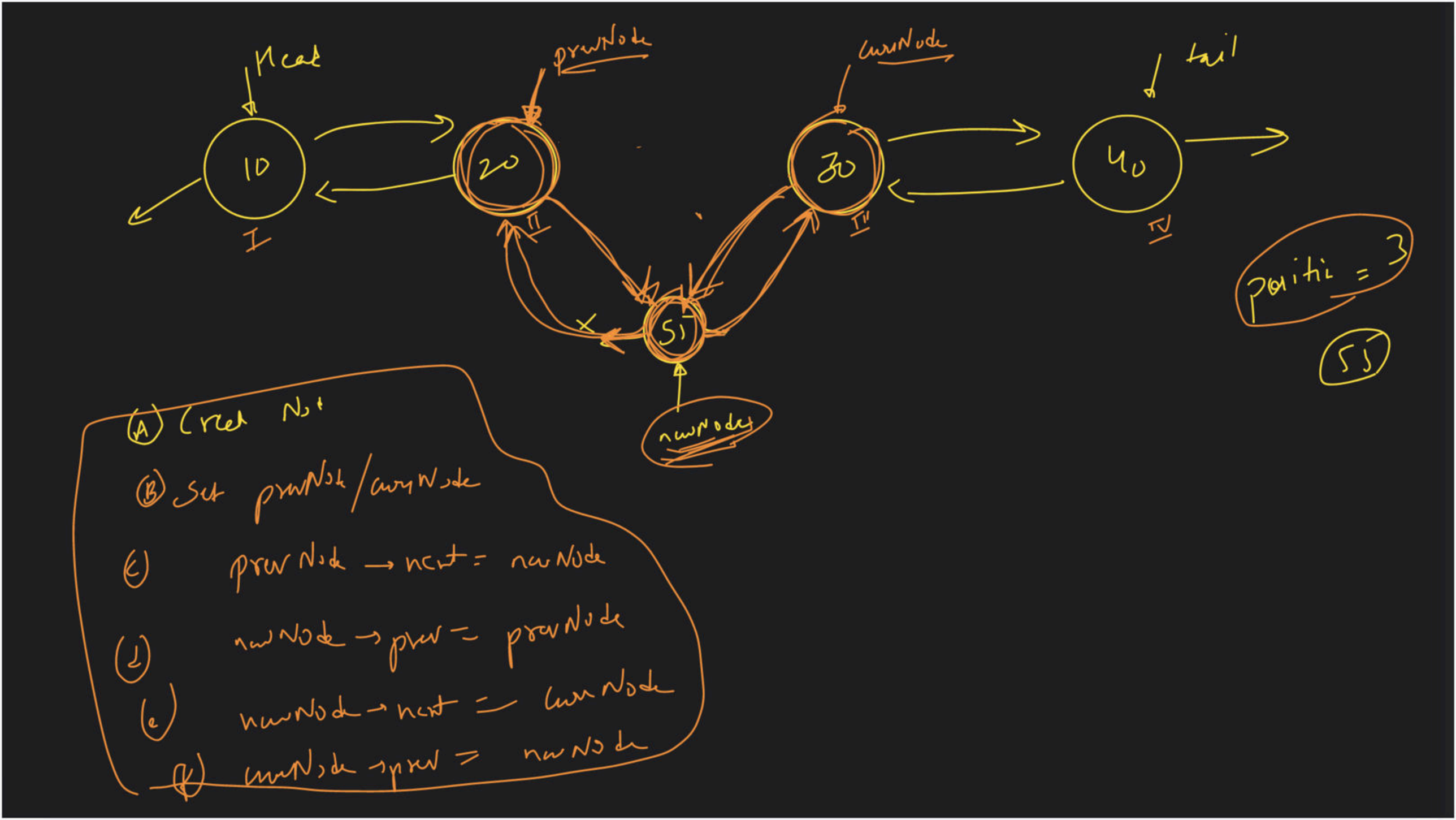
(B) Set prev/curr

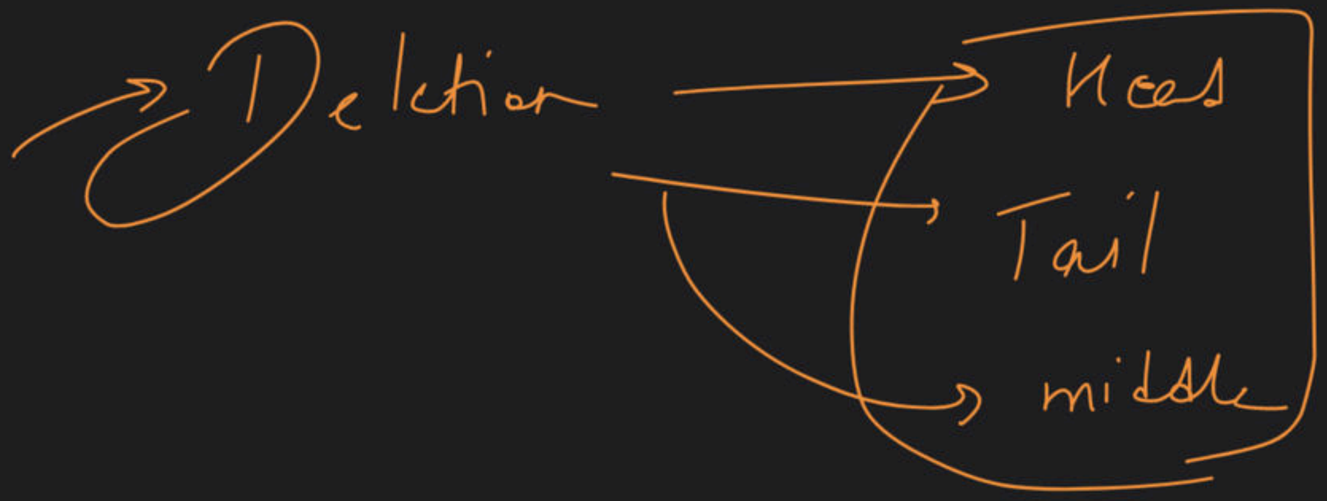
(C) prevNode → next = newNode

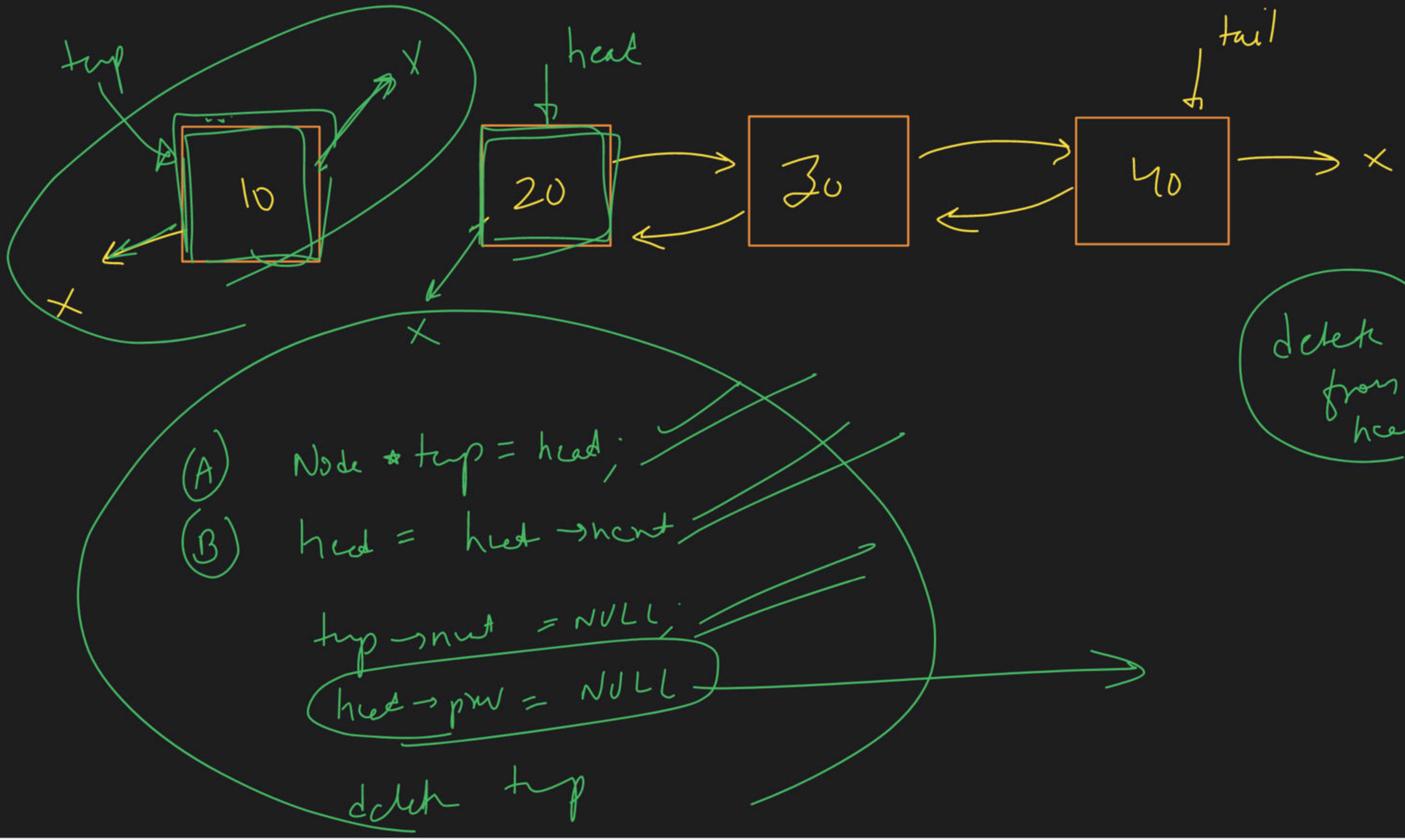
newNode → prev = prevNode

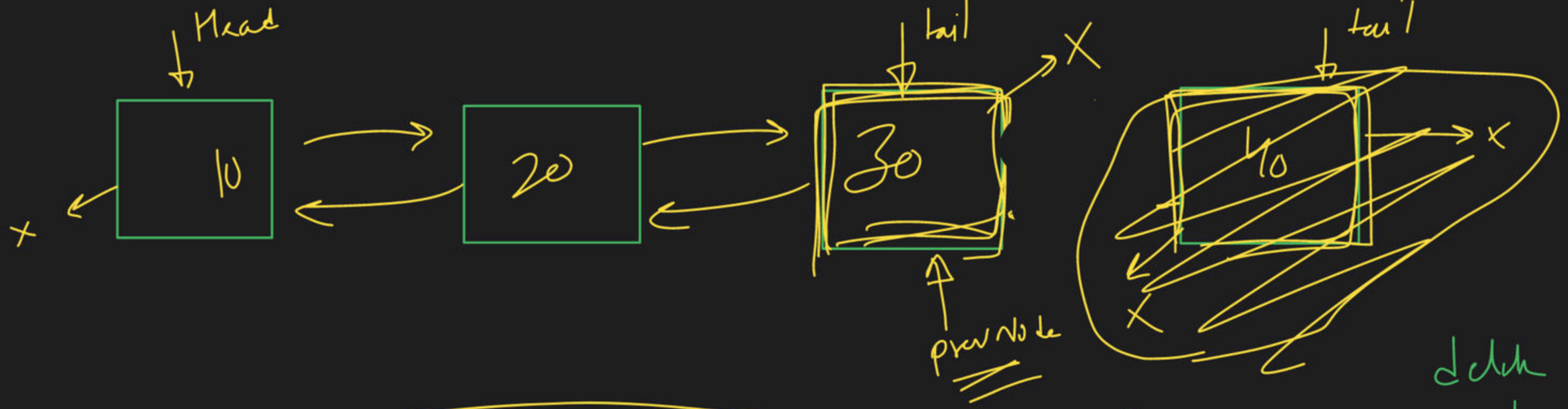
newNode → next = currNode

currNode → prev = newNode


head
prevNode
currNode
tail
10
20
30
40
I
II
III
IV
x
55
newNode
position = 3
55
(A) Create Node
(B) Set prevNode / currNode
(C) prevNode → next = newNode
(D) newNode → prev = prevNode
(E) newNode → next = currNode
(F) currNode → prev = newNode


Deletion
Head
Tail
middle

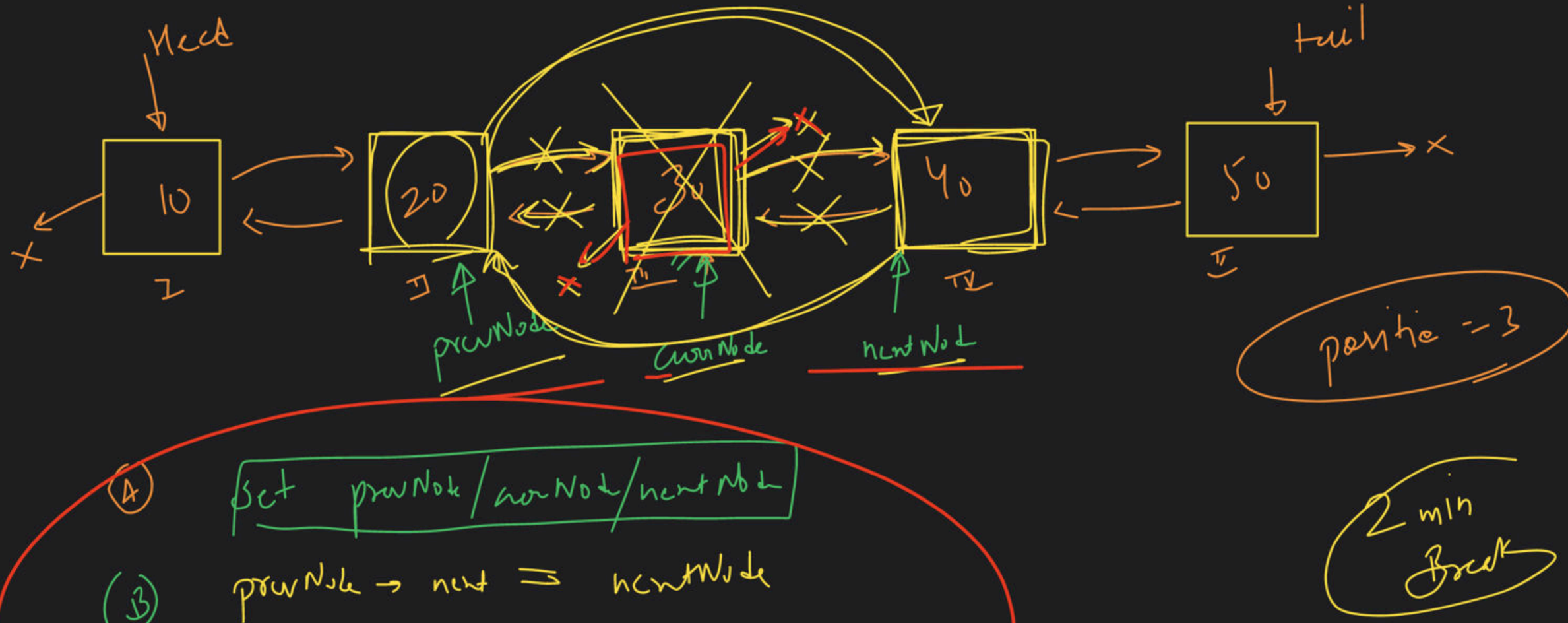

temp
head
tail
10
20
30
40
delete from head
(A) Node * temp = head;
(B) head = head->next
temp->next = NULL;
head->prev = NULL
delete temp



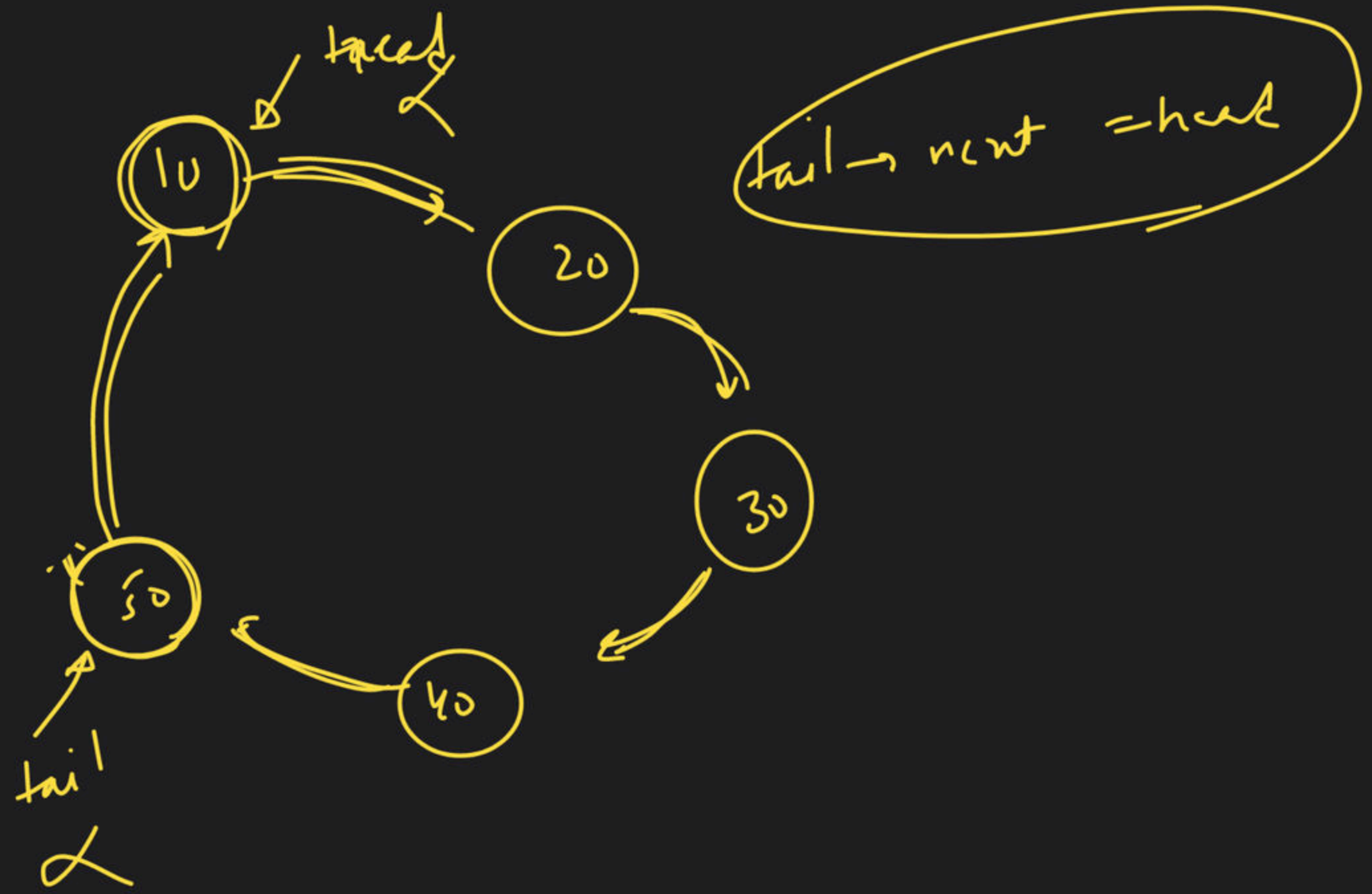
delete from tail (10X)

(A)

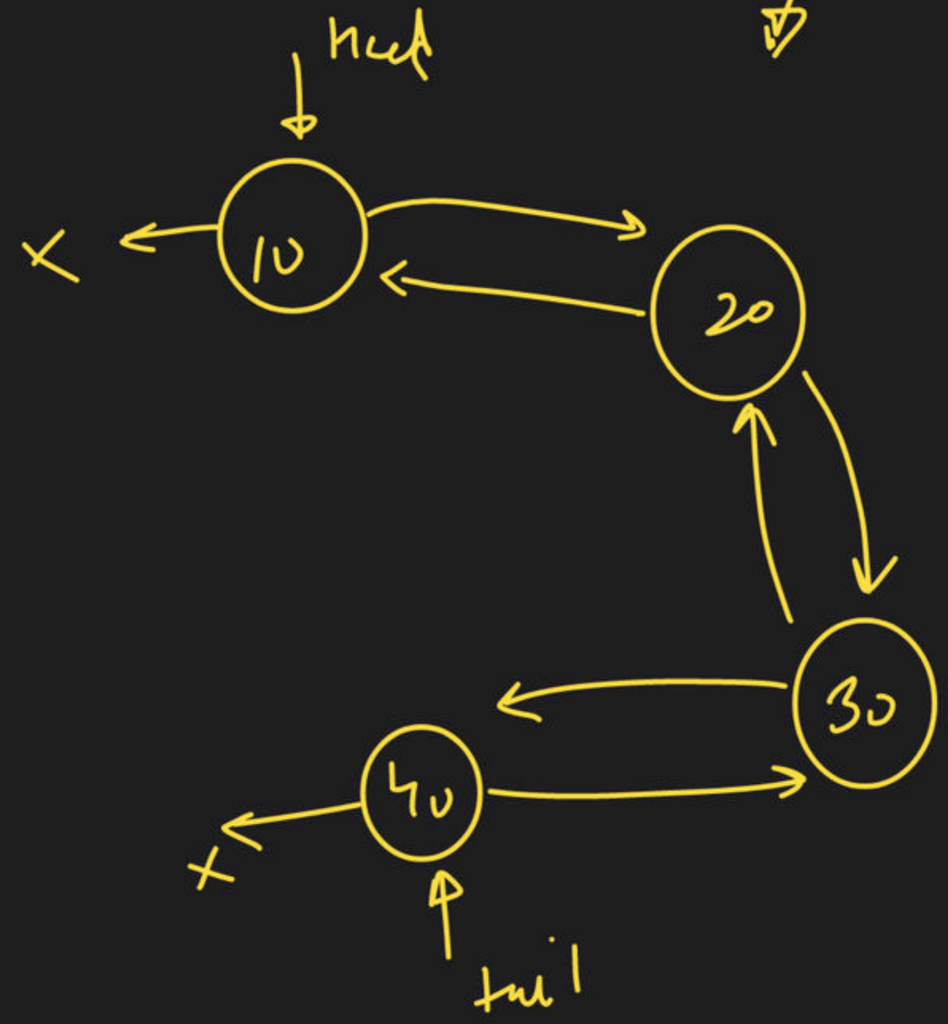
```
Node *prevNode = tail->prev
prevNode->next = NULL
tail->prev = NULL
delete tail
tail = prevNode
```


Head
tail
10
20
30
40
50
prevNode
currNode
nextNode
position = 3
4
set prevNode / currNode / nextNode
3
prevNode → next = nextNode
currNode → prev = NULL
currNode → next = NULL
nextNode → prev = prevNode
delete currNode
2 min Break

Circular → Singly Linked List



Circular → Doubly L.L

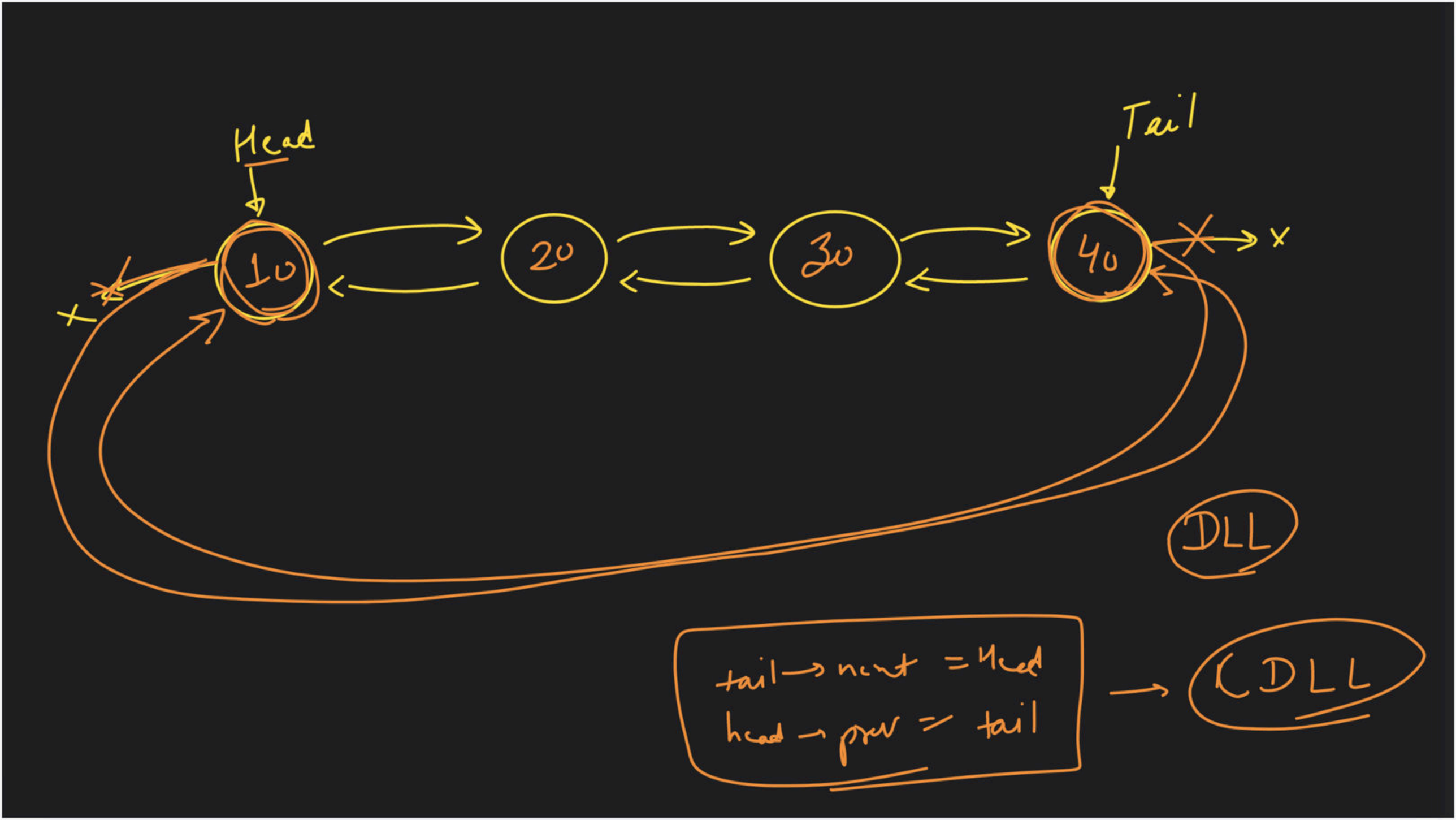


Marathon

→ L.B

↓

L.L → Query


Head
Tail
10
20
30
40
x
x
DLL
tail → next = Head
head → prev = tail
CDLL


detect & delete
loop


$\mathcal{G}_{pm}$
LL
III