

Linked List Class - 1

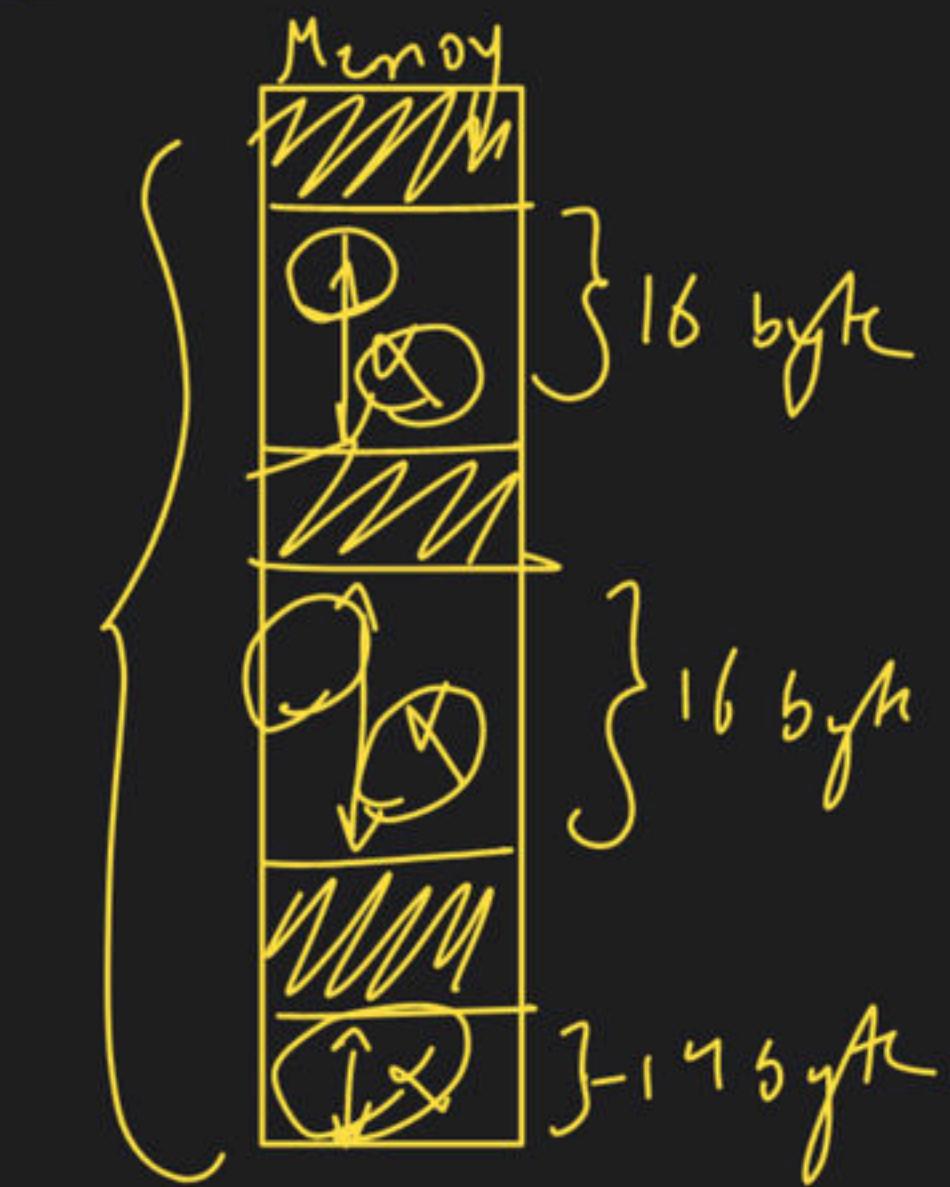
Special class

→ Linked List
collection of nodes

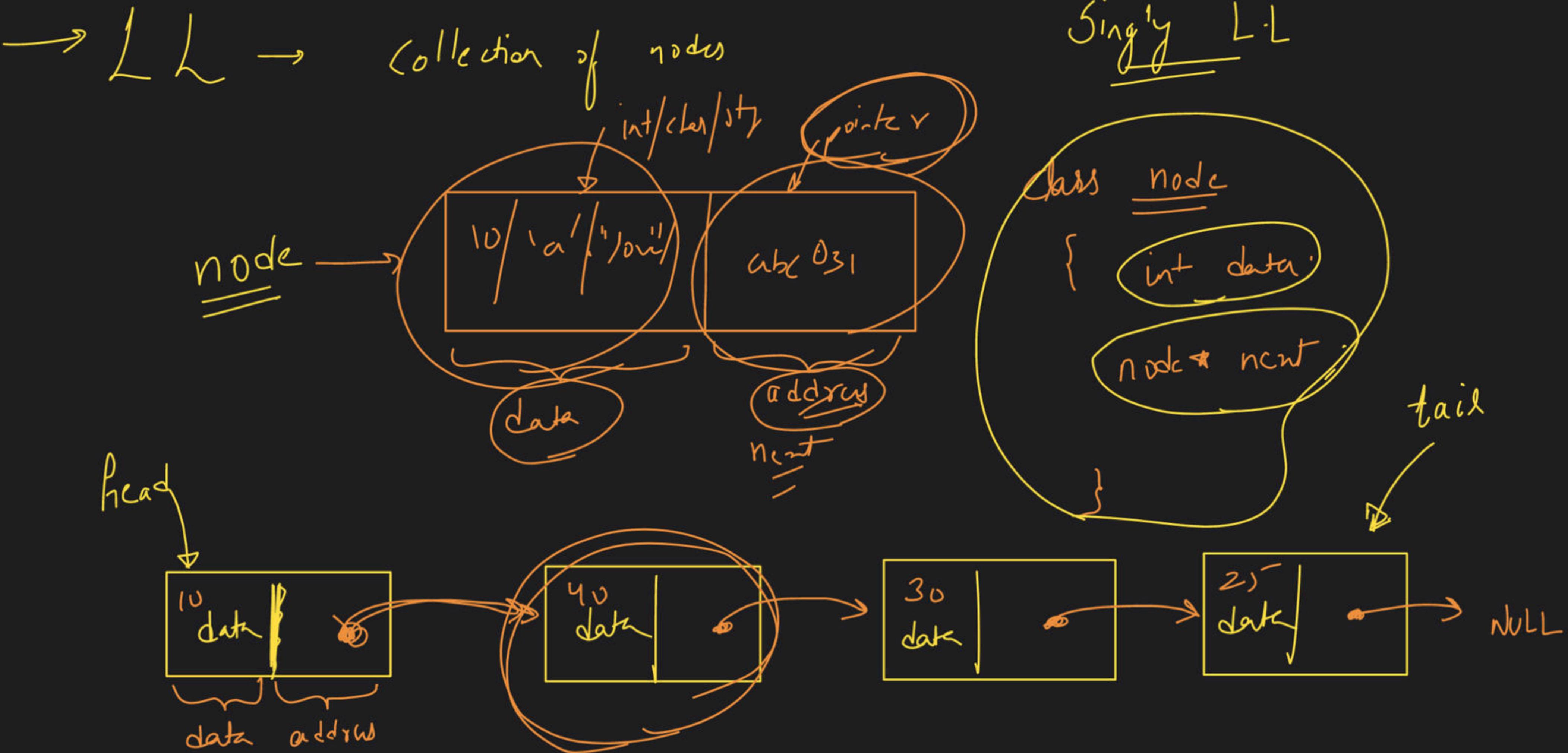


20 byte
LL

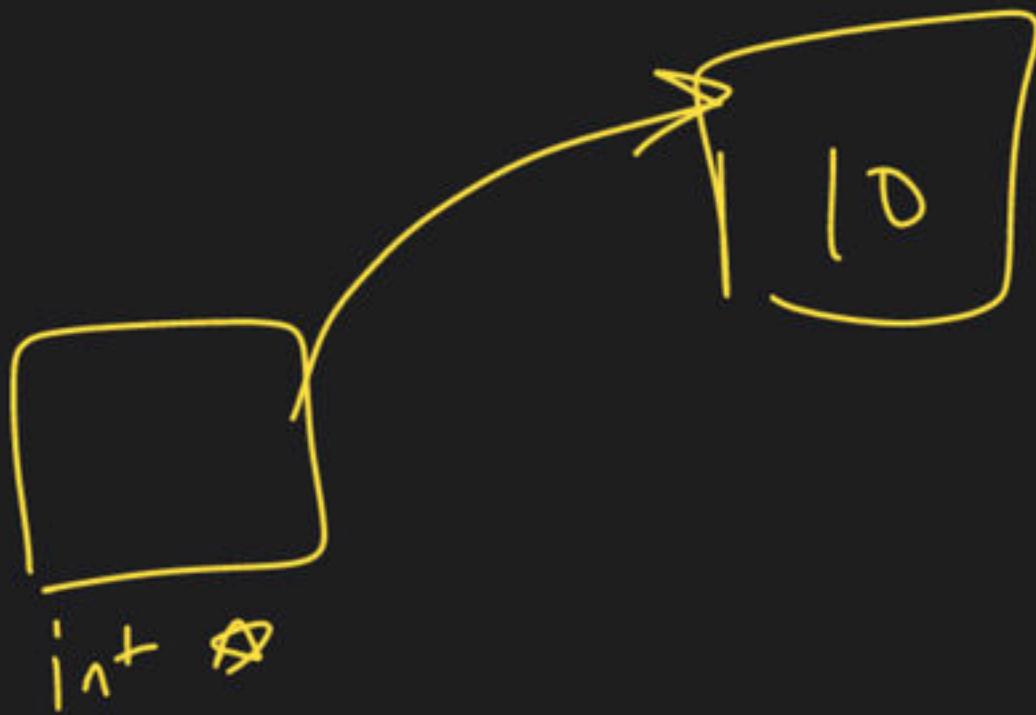
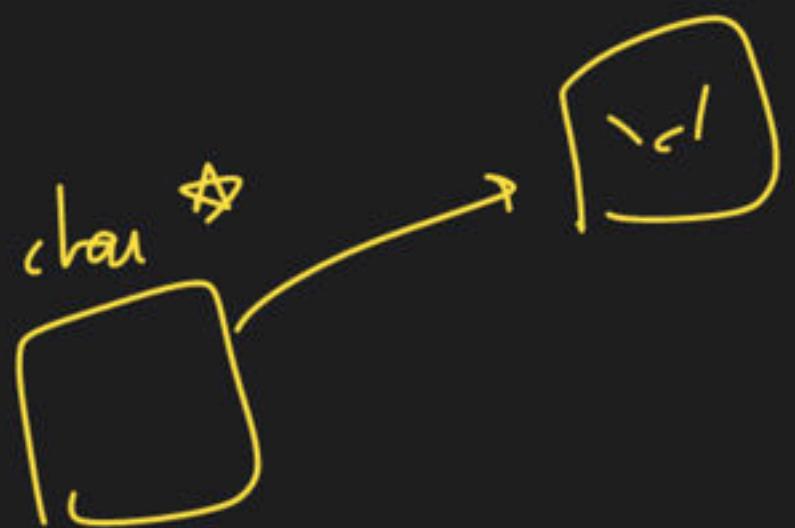
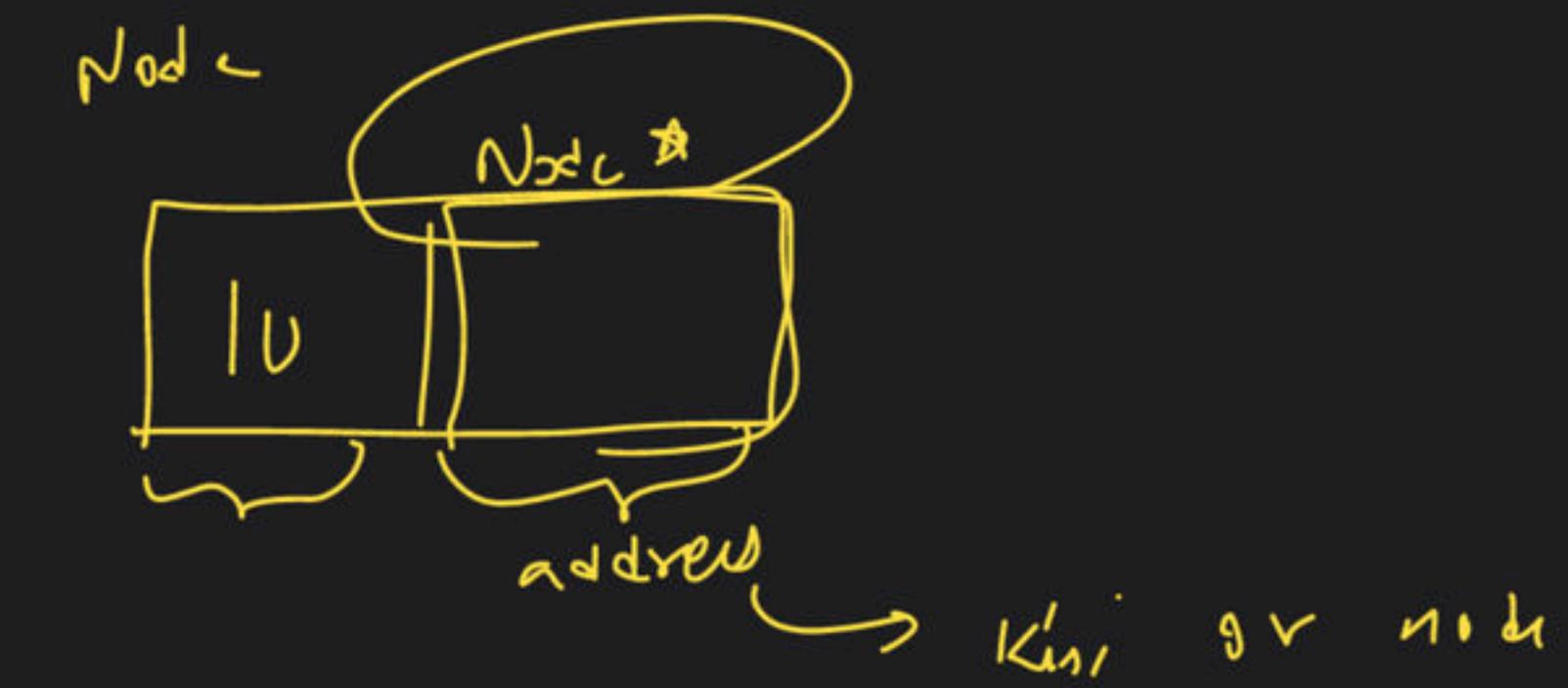
20 byte
array



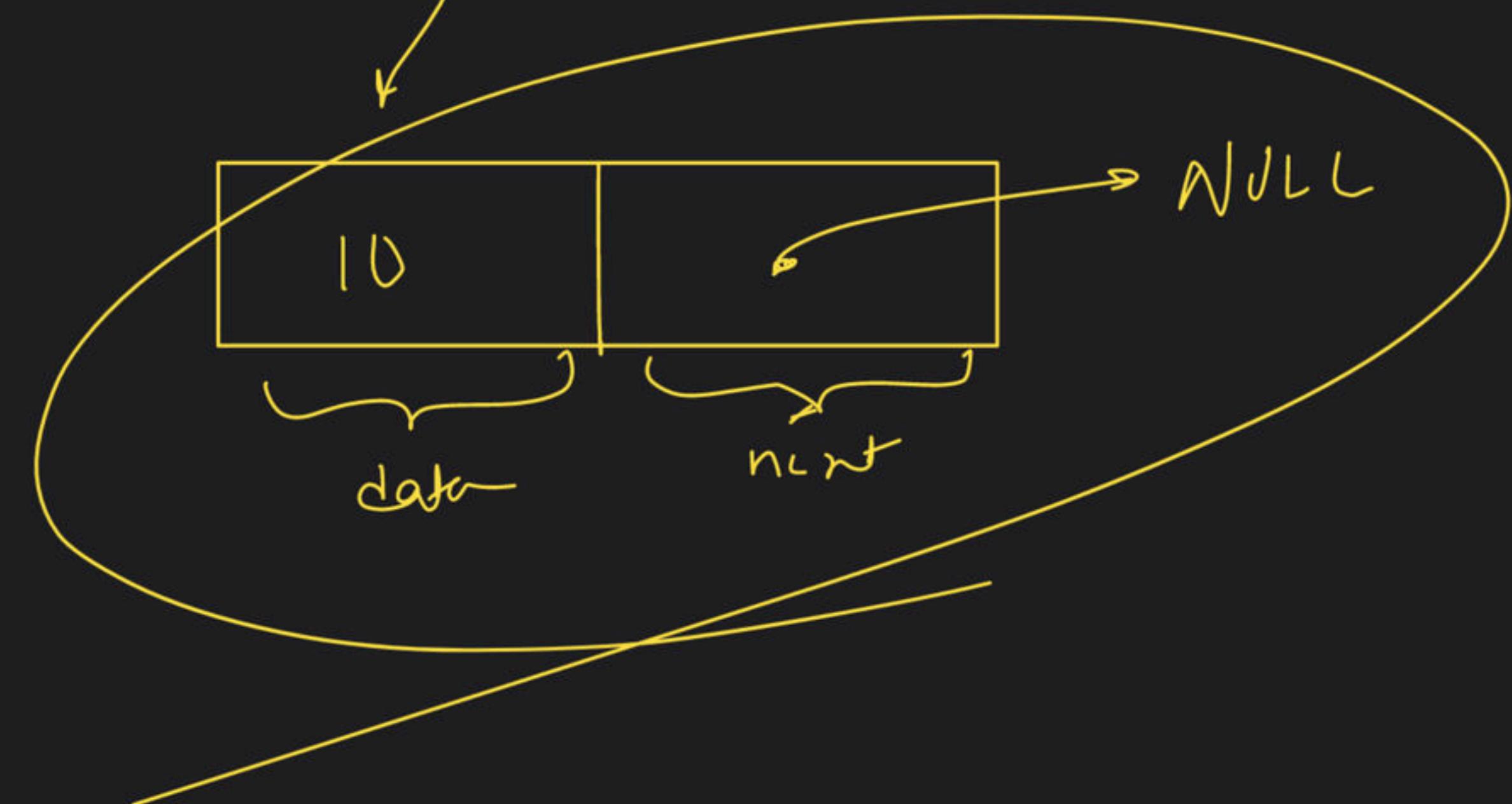
Total free
16 + 16 + 16
153 bytes
X



```
class Node  
{  
    int data;  
    Node* next;
```



Node * first = new Node(10)



L L

insertHead (101)

insertAtTail (200)

head

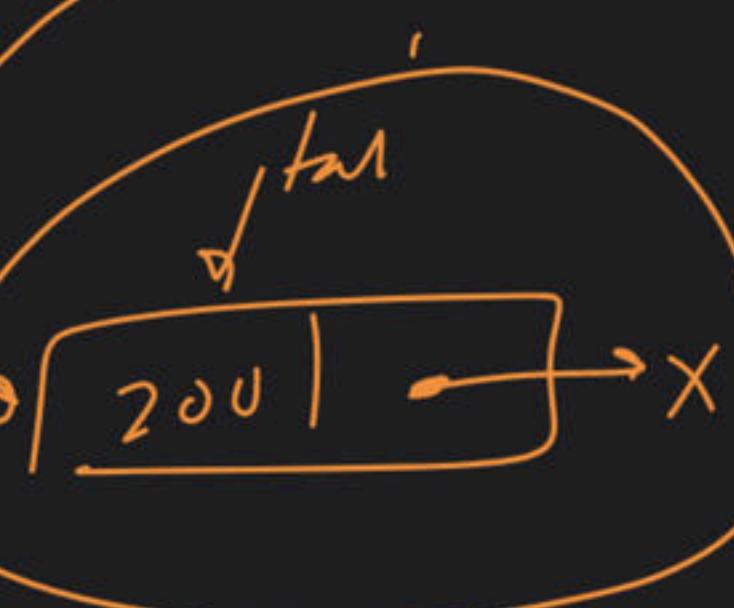
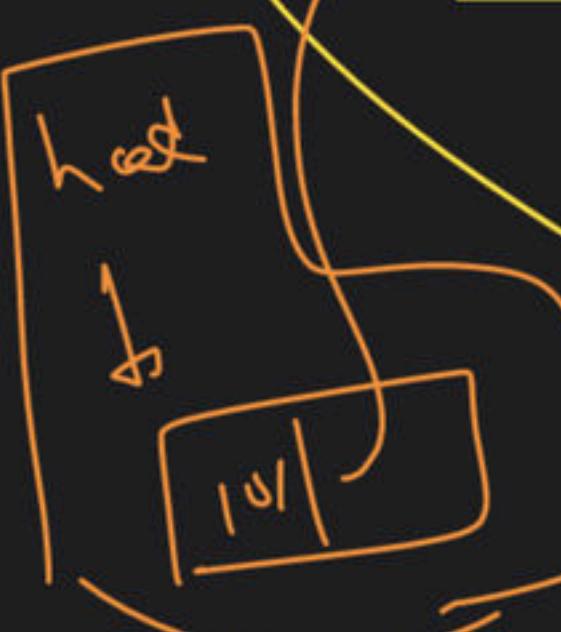
10	
----	--

20	
----	--

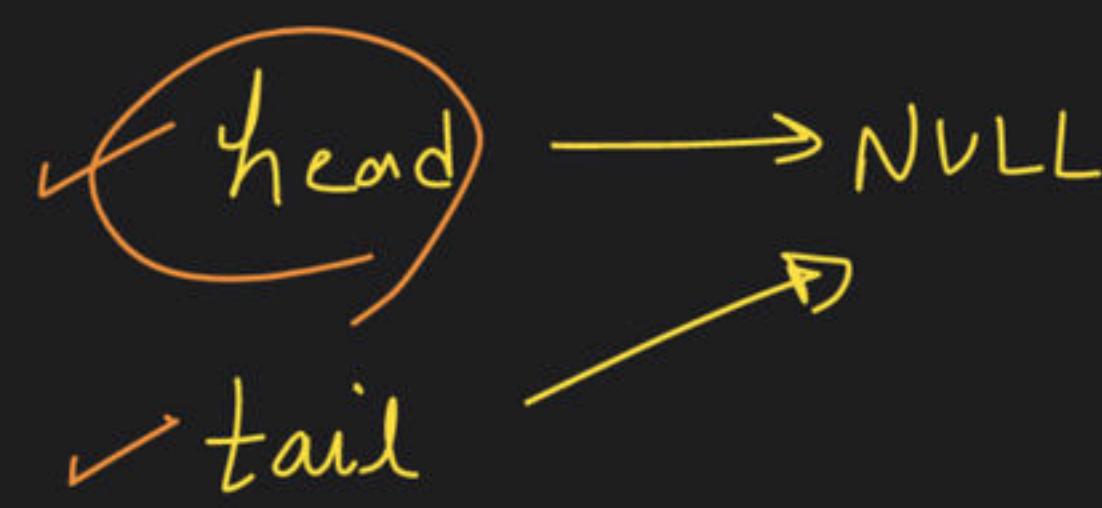
30	
----	--

40	
----	--

50	
----	--

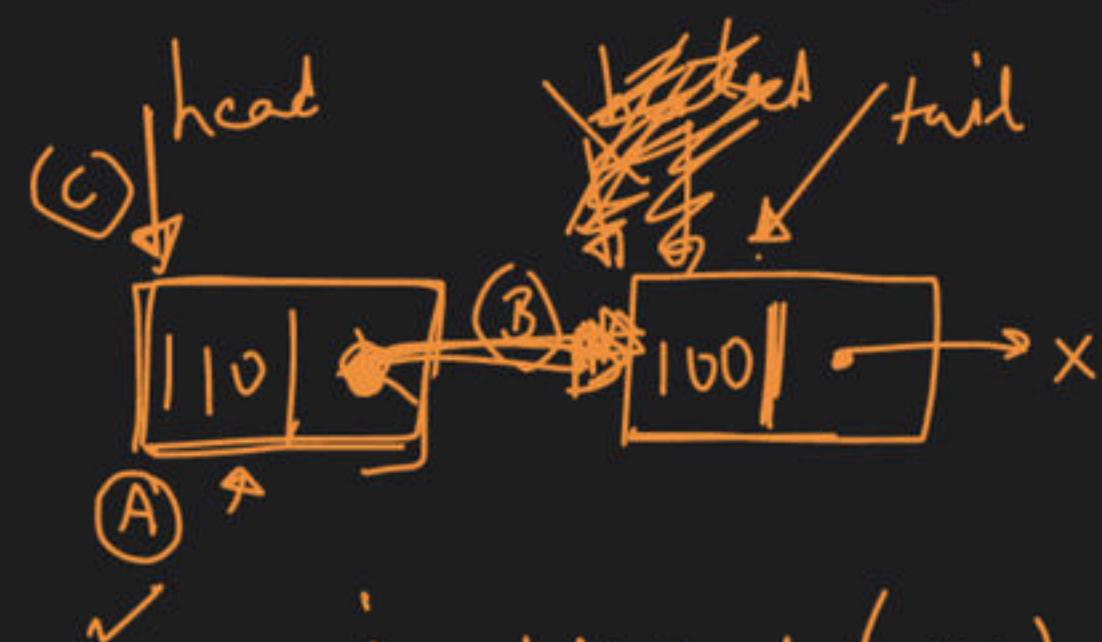


Doh is Mind.



→ 0 nodes

insertAtHead(100)



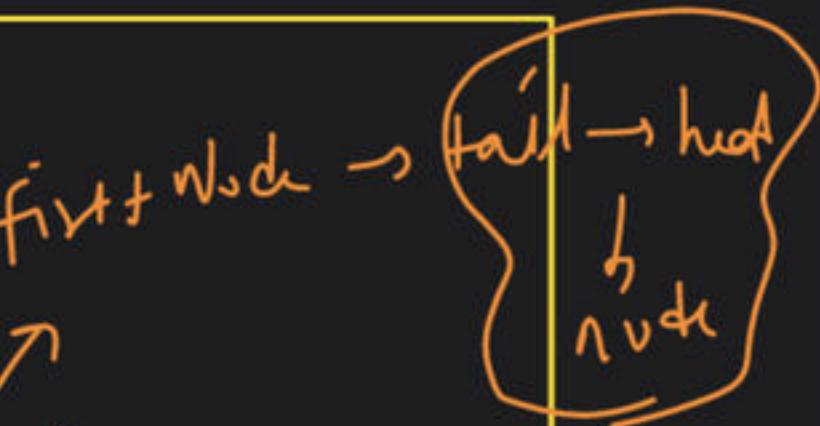
insertAtHead(110)

Inser~~tion~~

↳ insert At Head

↳ insert At Tail

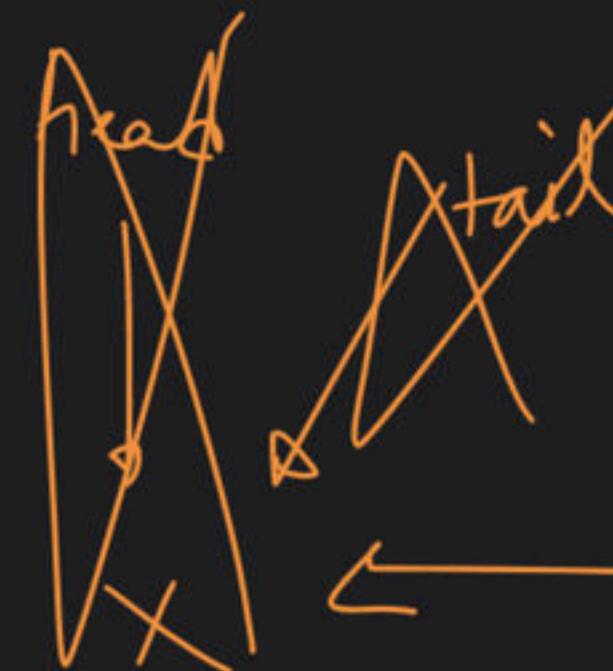
↳ insert At Position



(A) $\text{Node} \Rightarrow \text{newNode} = \text{new Node}(110)$

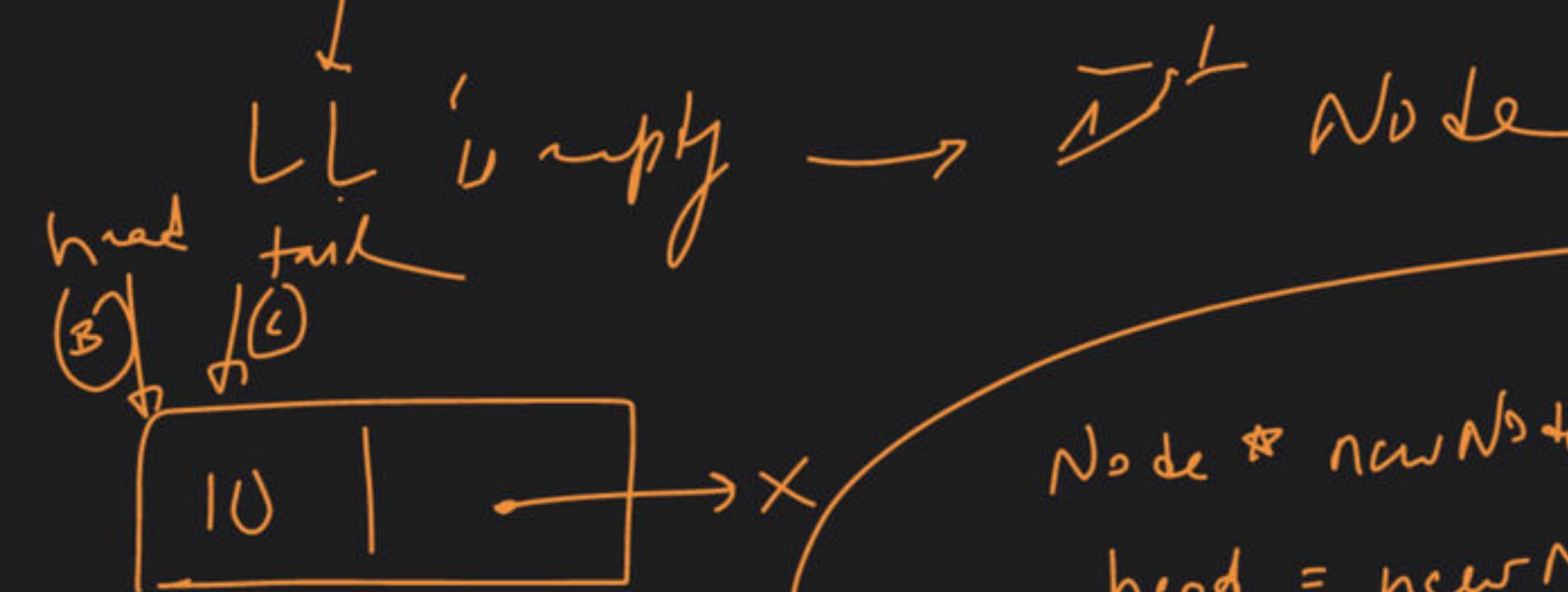
(B) $\text{nextNode} \rightarrow \text{next} = \text{head}$

(C) $\text{head} = \text{newNode}$

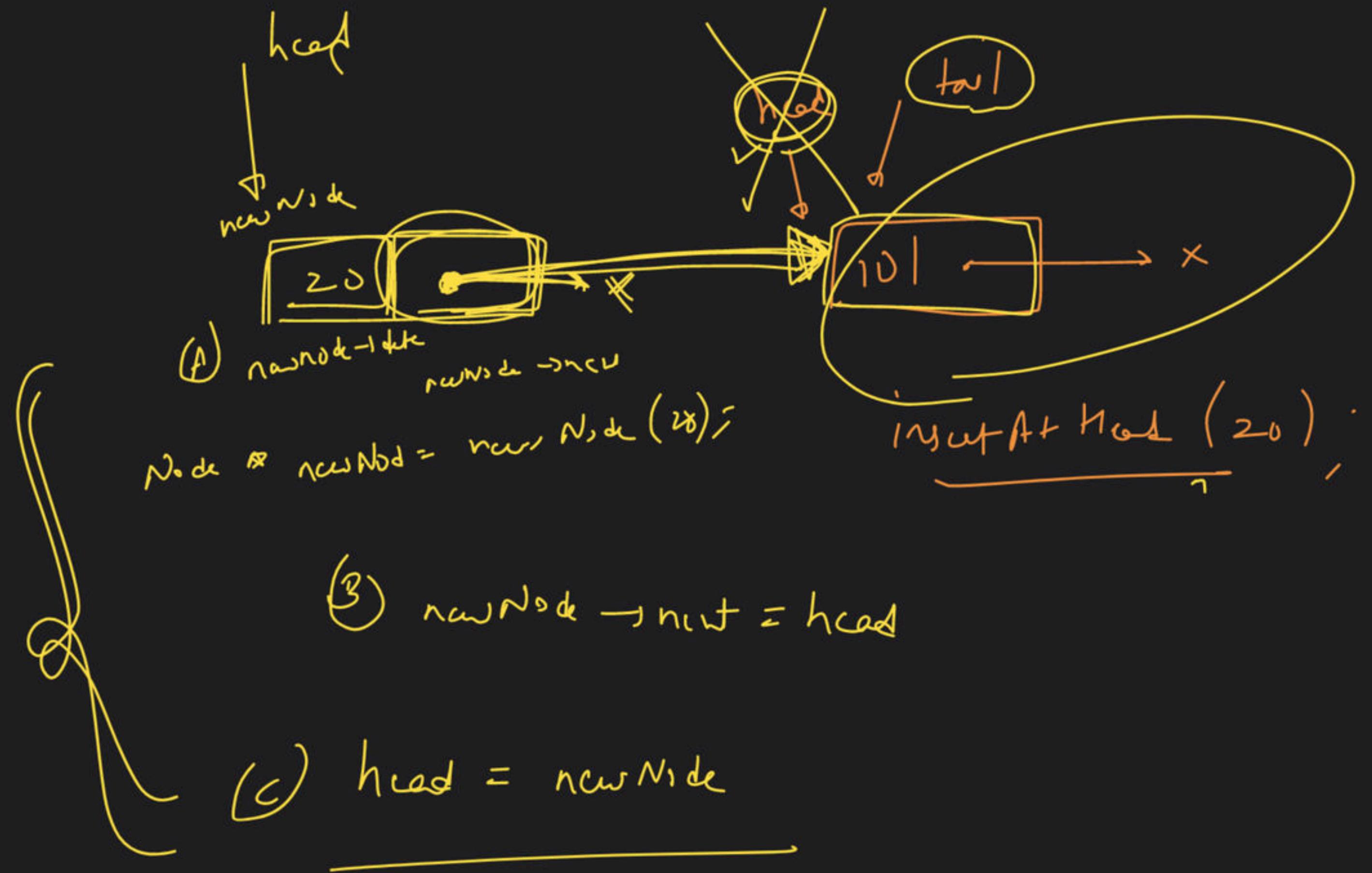


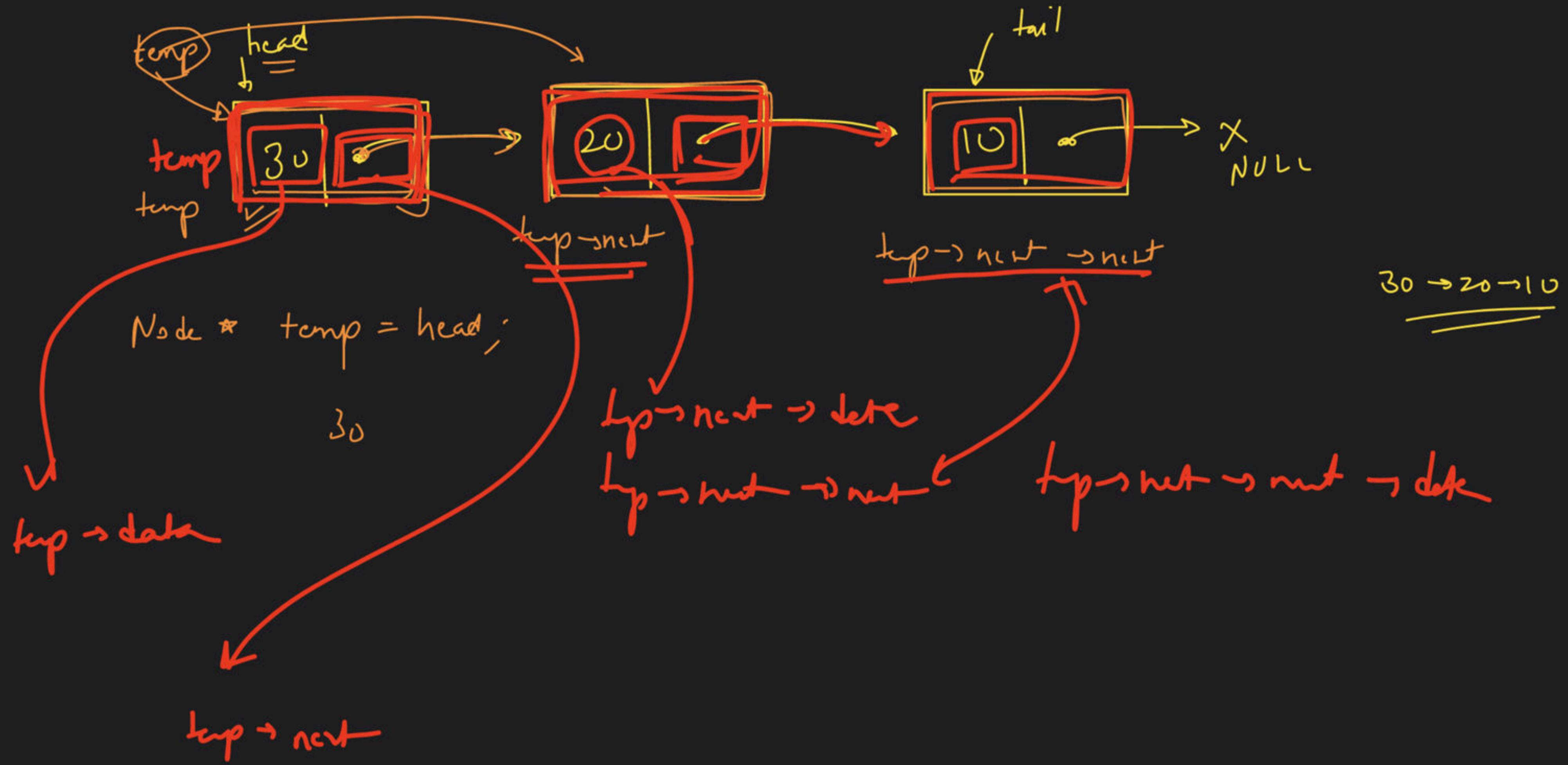
insert AtHead (10)

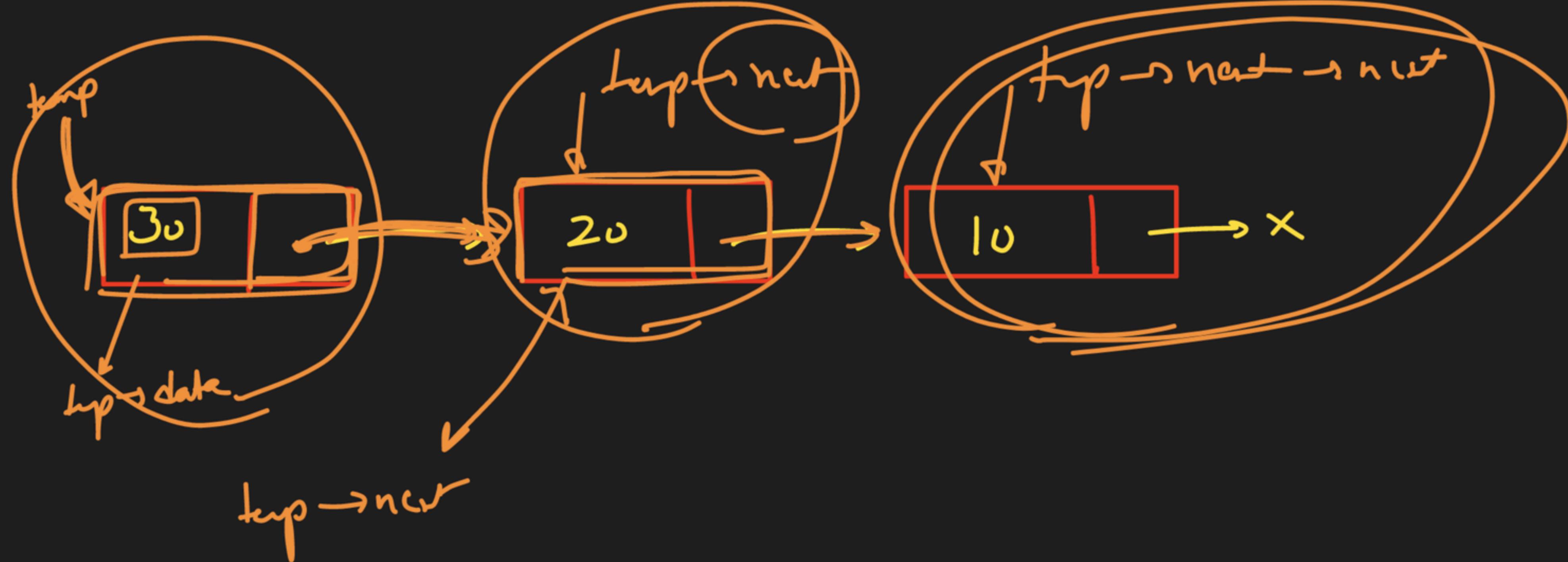
if (head == NULL && tail == NULL)

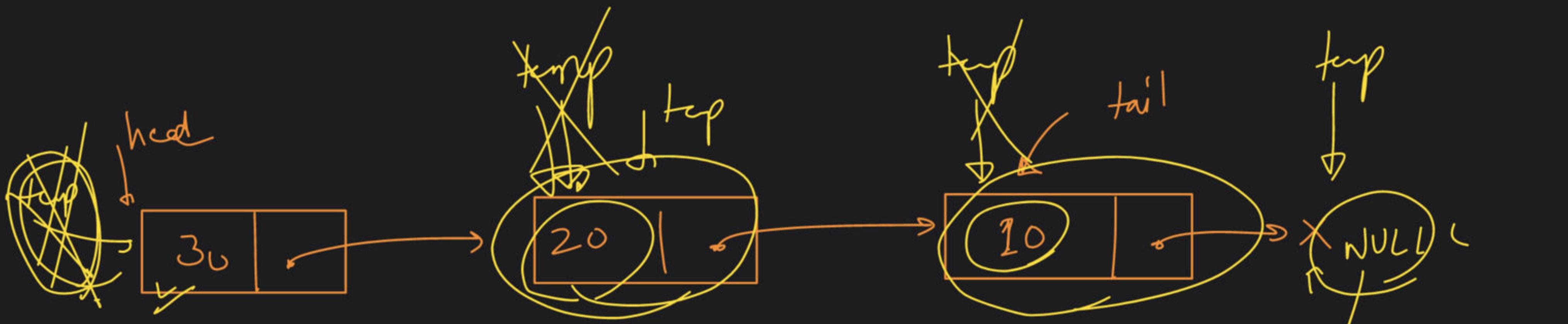


Node * newNode = new Node (value);
 head = newNode;
 tail = newNode;









$\text{Node} * \text{temp} = \text{head}$;

(cout << temp->data;

temp = temp->next

loop

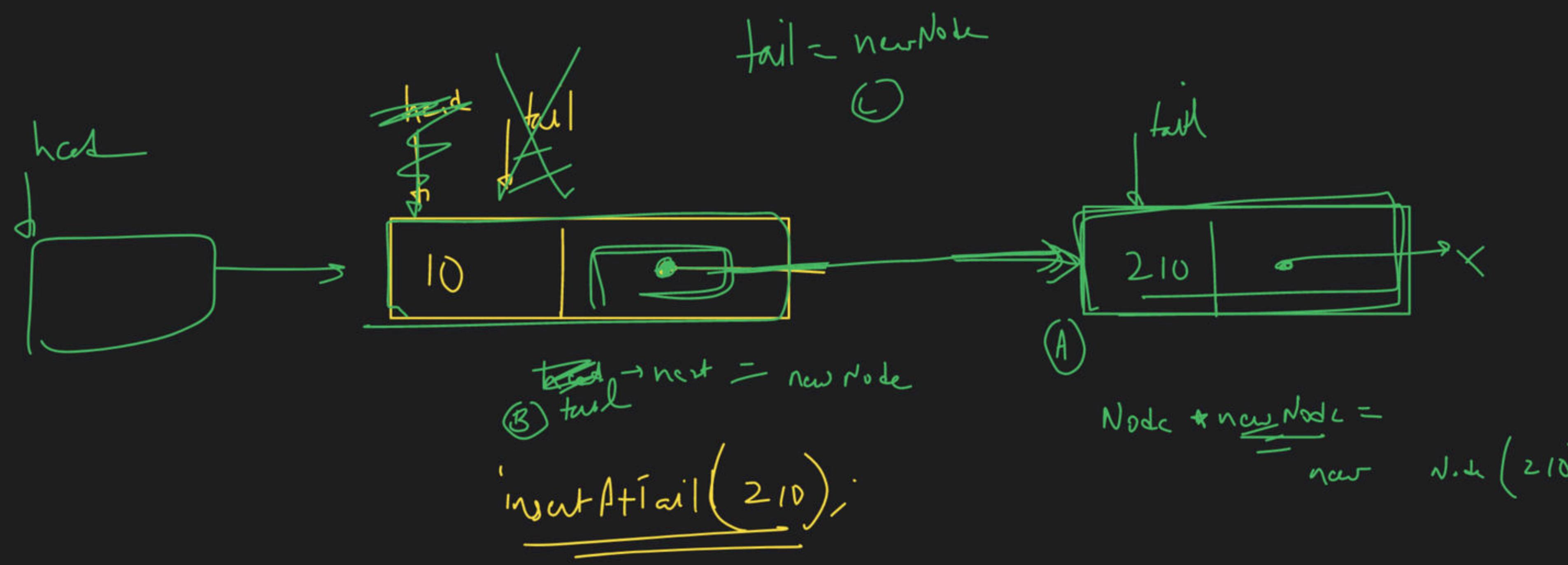
(cout << temp->data

temp = temp->next

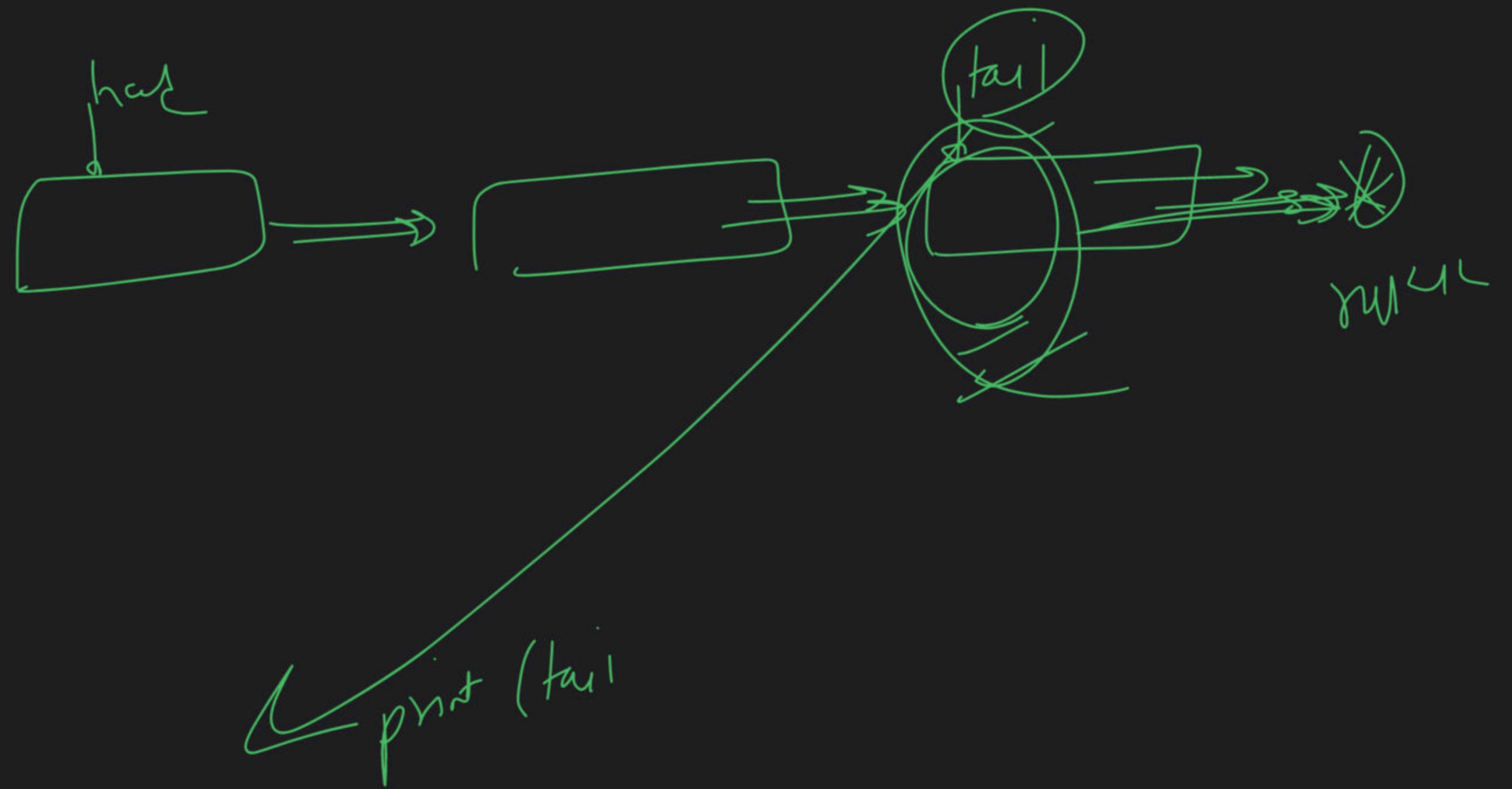
(cout << temp->data
temp = temp->next

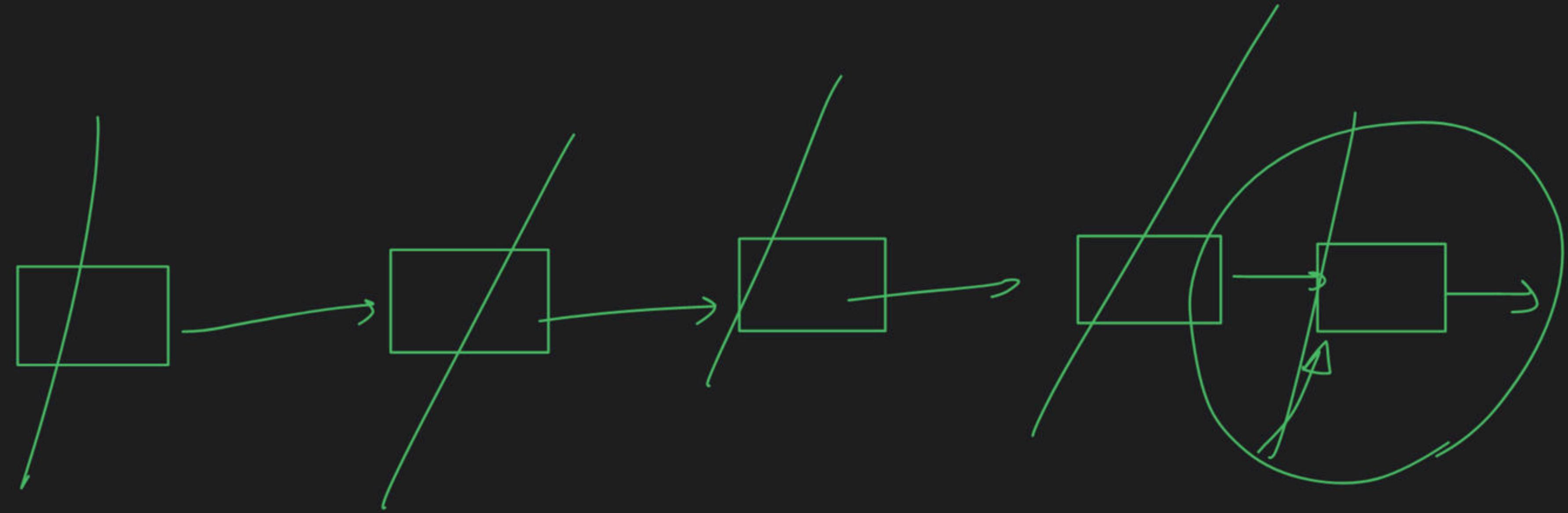
temp ko 1, temp
aage badhata)

i++
junk
garbage



2 min
 Break





array $\rightarrow O(1)$

LL $\rightarrow O(n)$

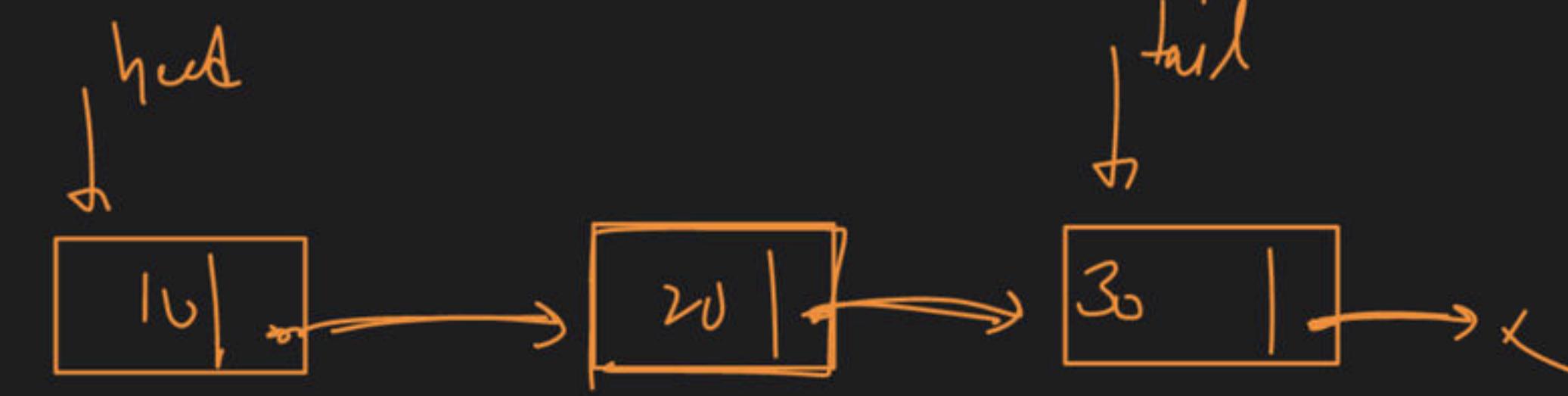
LL

Type

Singly LL

doubly LL

Circular LL



Node

class Node

```
{ int data;  
Node* next }
```

insertion

insert At Head

insert At Tail

insert At Position

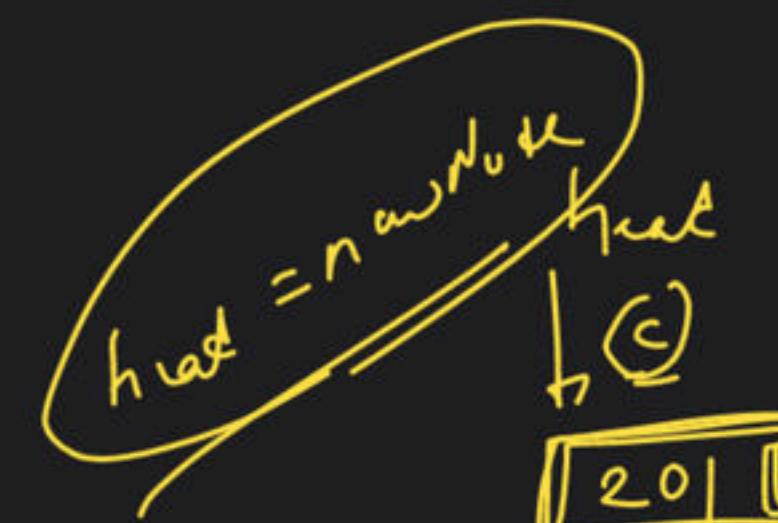
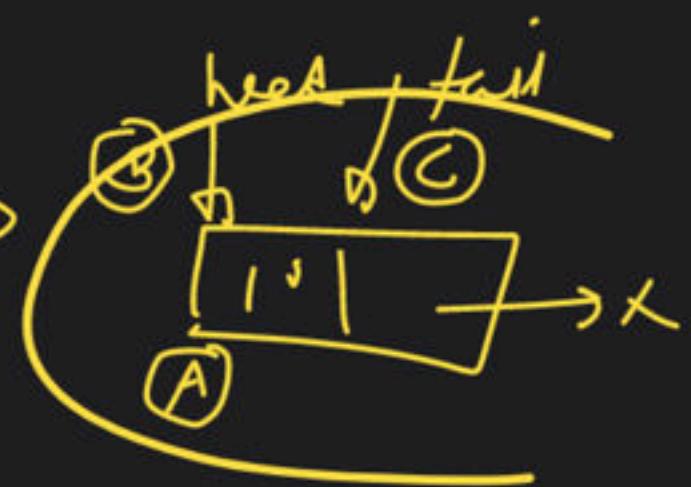
2 case

LL is empty

LL is not empty

if

(head = NULL
tail = NULL)



$$\underline{\text{newNode} \rightarrow \text{next} = \text{head}}$$

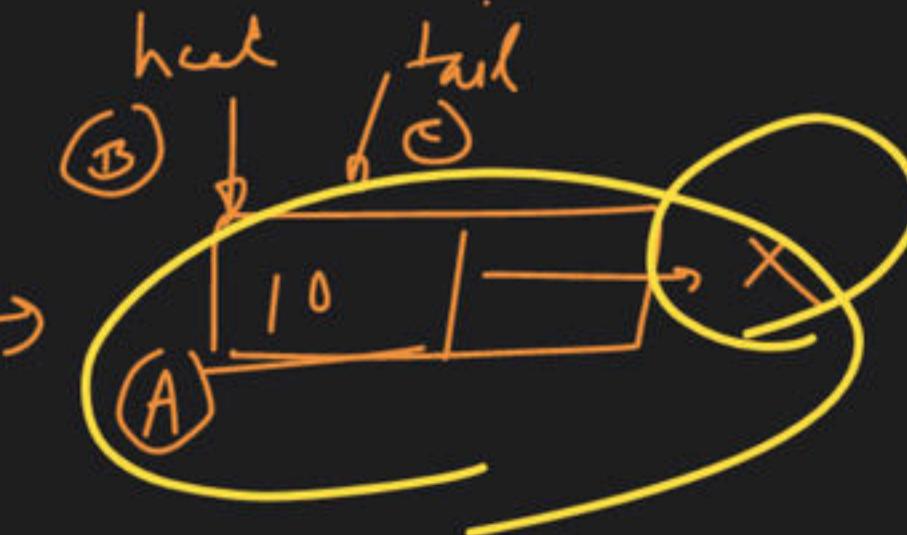
$$\underline{\text{Node } \& \text{ newNode } \rightarrow \text{next} = \text{new Node}(20)}$$

2 Case

insert A + tail

LL is empty →

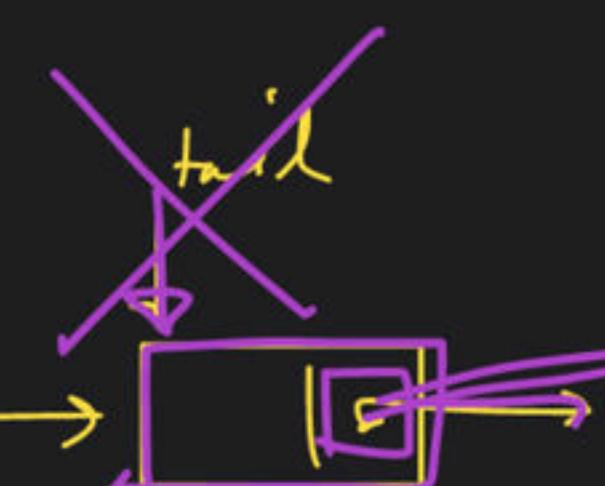
if
head = NULL
tail = NULL



LL is not empty

tail = newNode

head

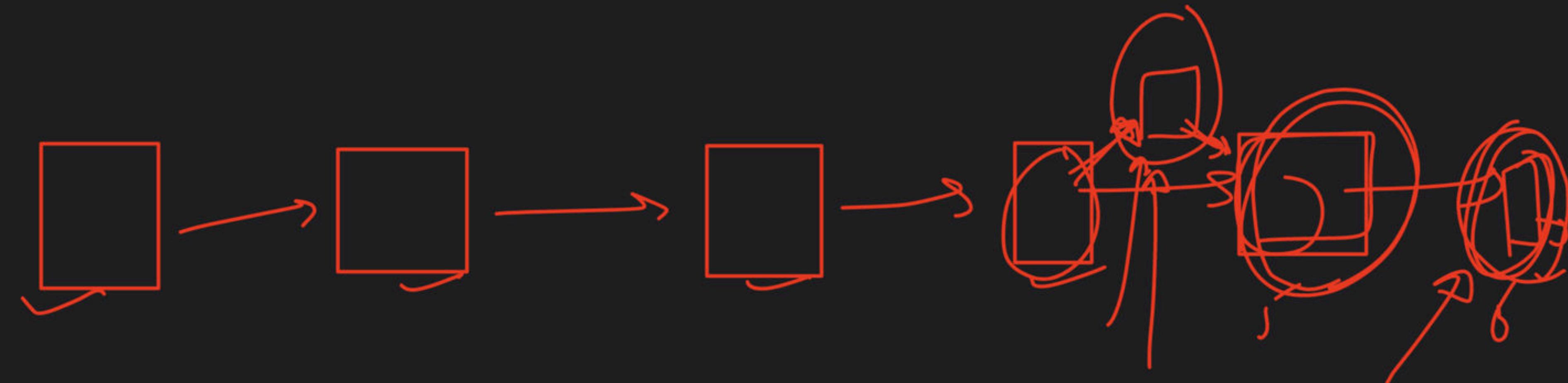
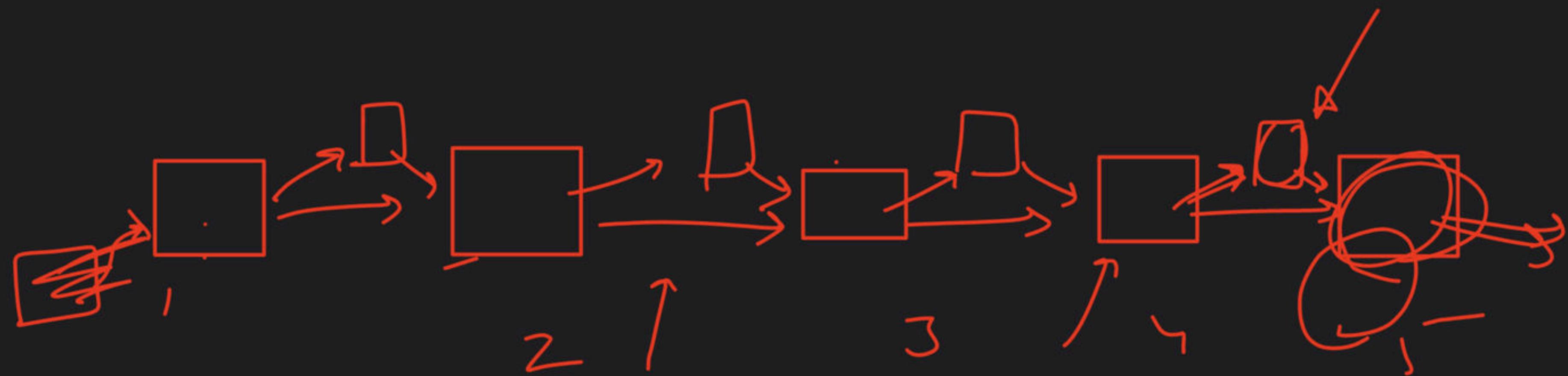


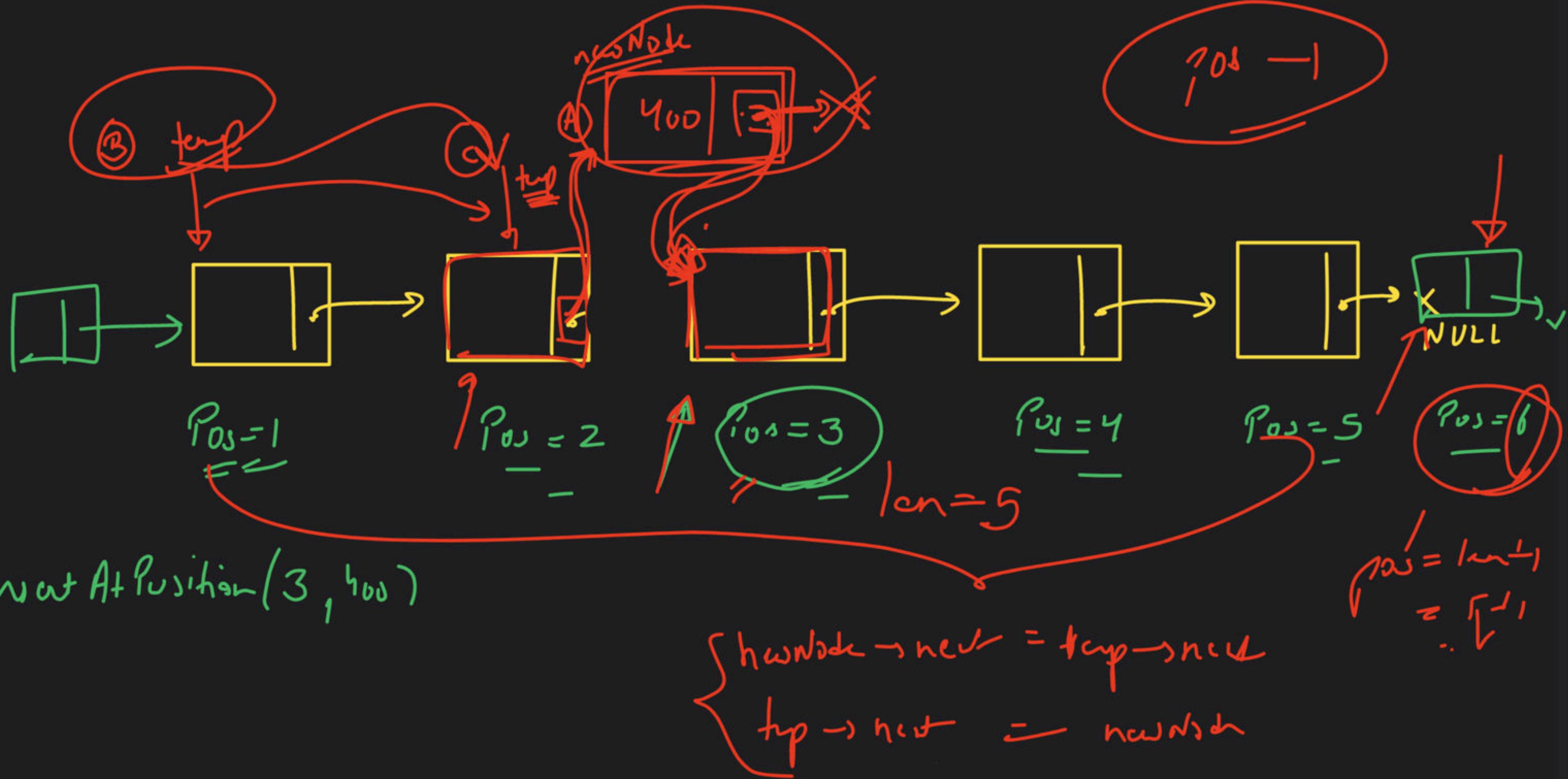
insert A + tail (y₀₀)

(3)

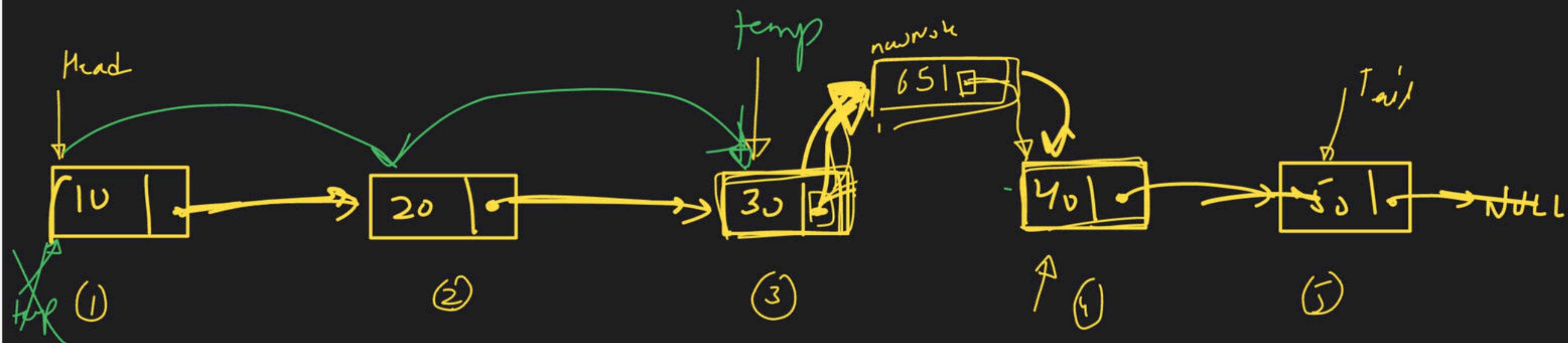
tail → next = newNode

Node * newNode =
new Node(y₀₀);





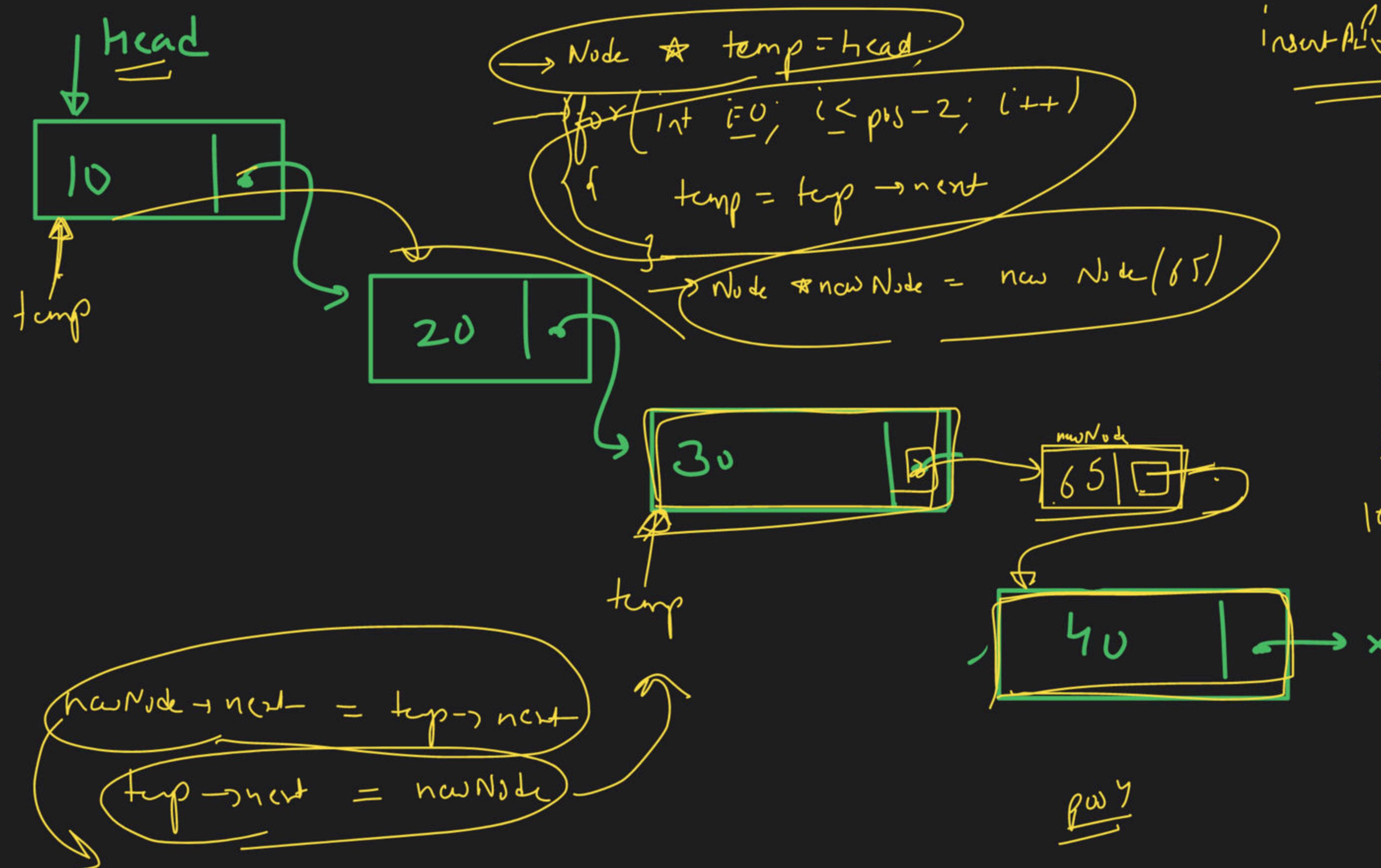
position - 1 → will be node



insert At Position (4, 65, head, tail)

position - 2

Node * temp = head;
for (i=0; i < pos-2; i++)
 temp = temp -> next;
}
Node * newNode = new Node(65);
newNode -> next = temp -> next;
temp -> next = newNode;

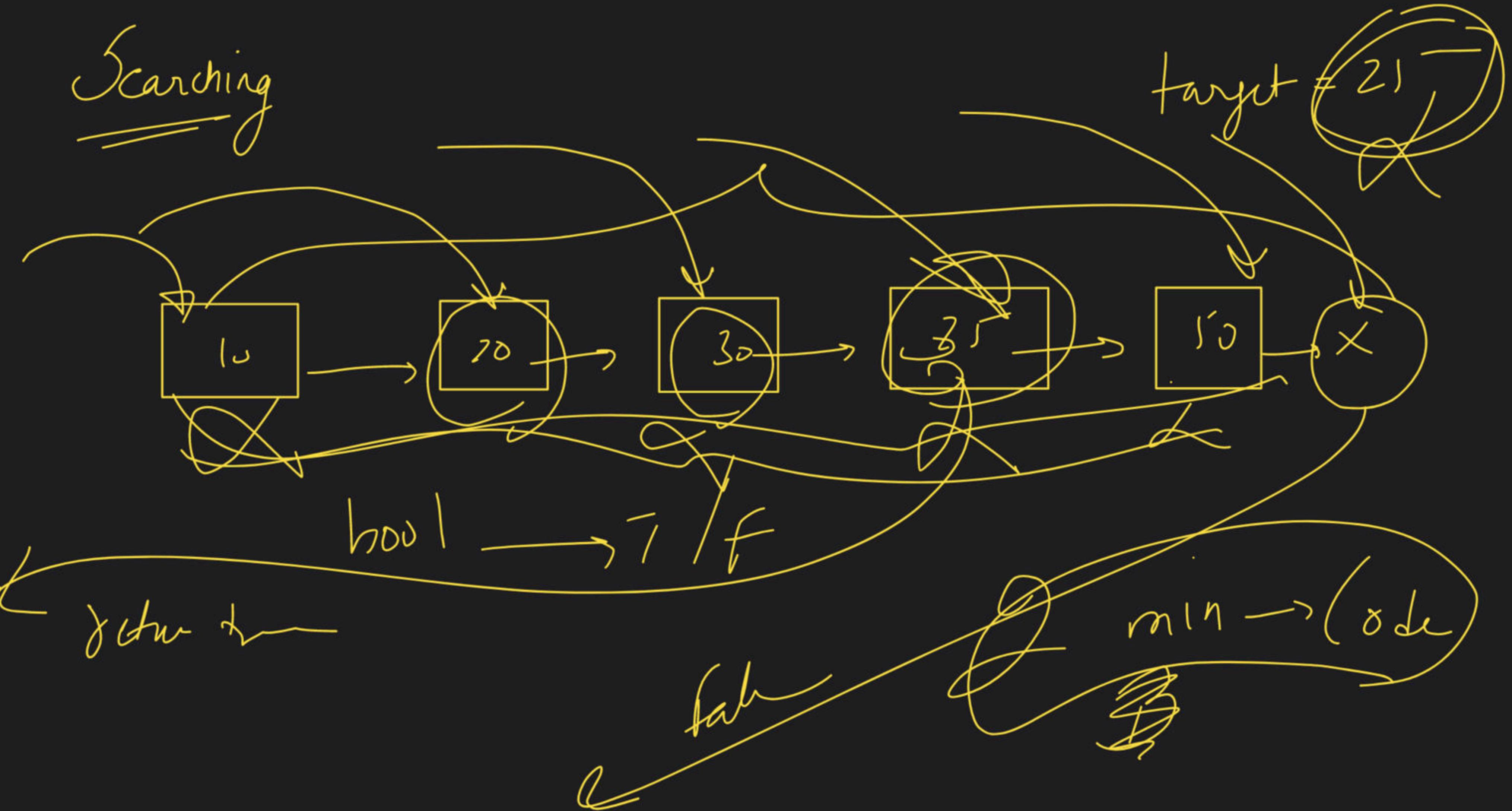


insert AVL σ (4)
65 value

4 → 2 shr
2 → 3 shr
10 → 8 LH
20 → 18 RL
100 → 98 LH
100 - 2
3 shr



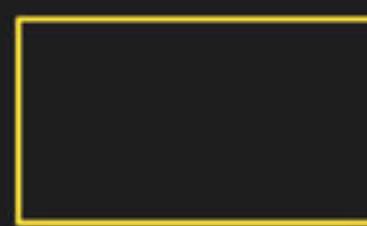
Scanning



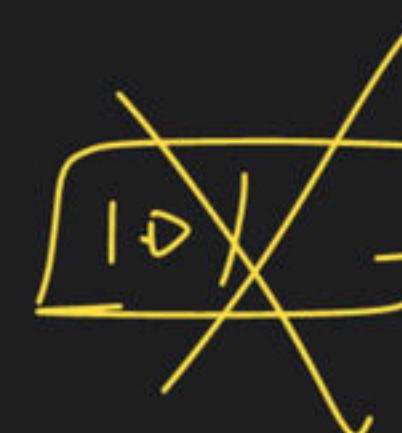
Deletion

~~0~~ LL is empty

cannot
delete

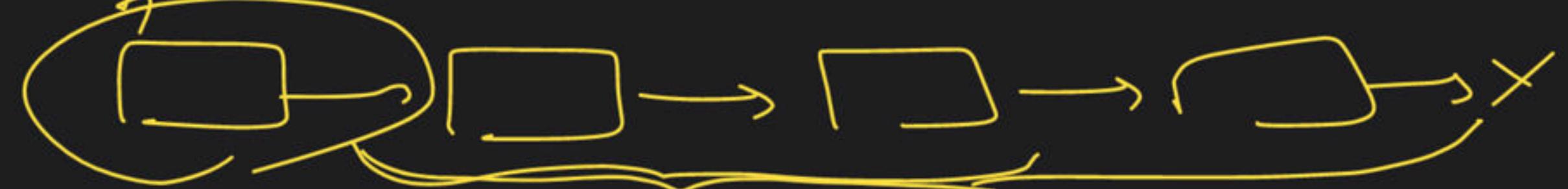


~~1~~ Single nod

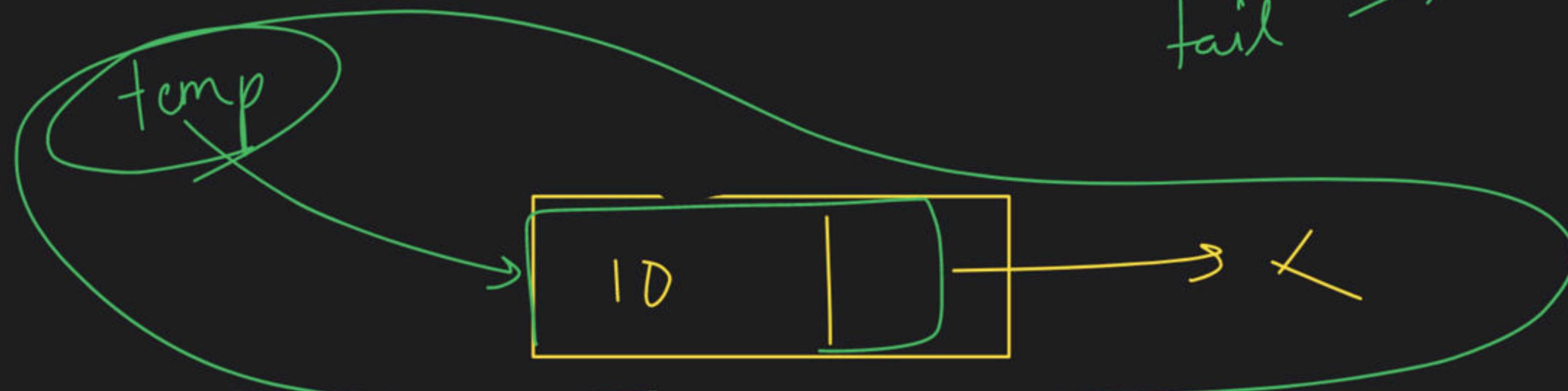


head = NULL
tail = NULL

~~2~~ multiple nodes



head → X
tail →

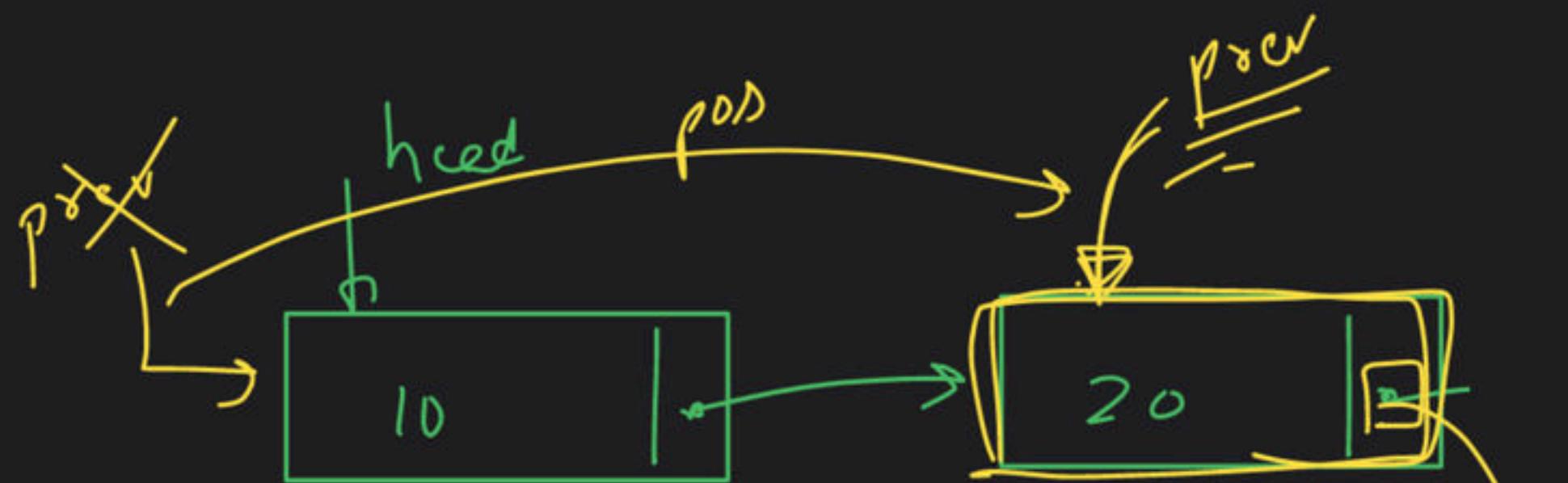


Node * temp = head;

head = NULL

tail = NULL ;

delete temp



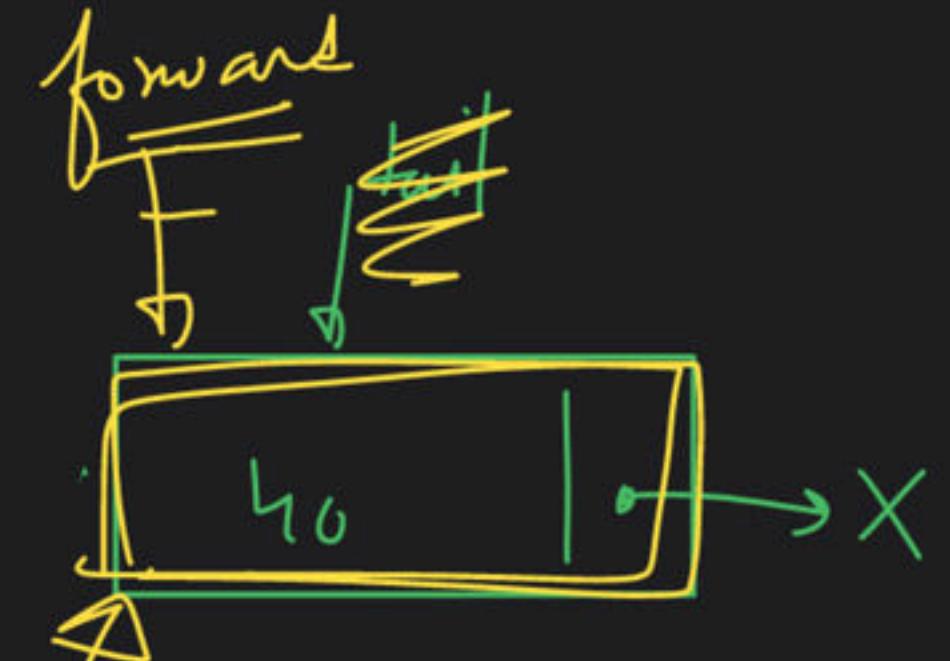
$pos = 1$

Node * prev = head
 for ($i=0; i < pos - 1; i++$)
 {
 prev = prev->next
 }

$deletePos(3)$

Node * curr = prev->next

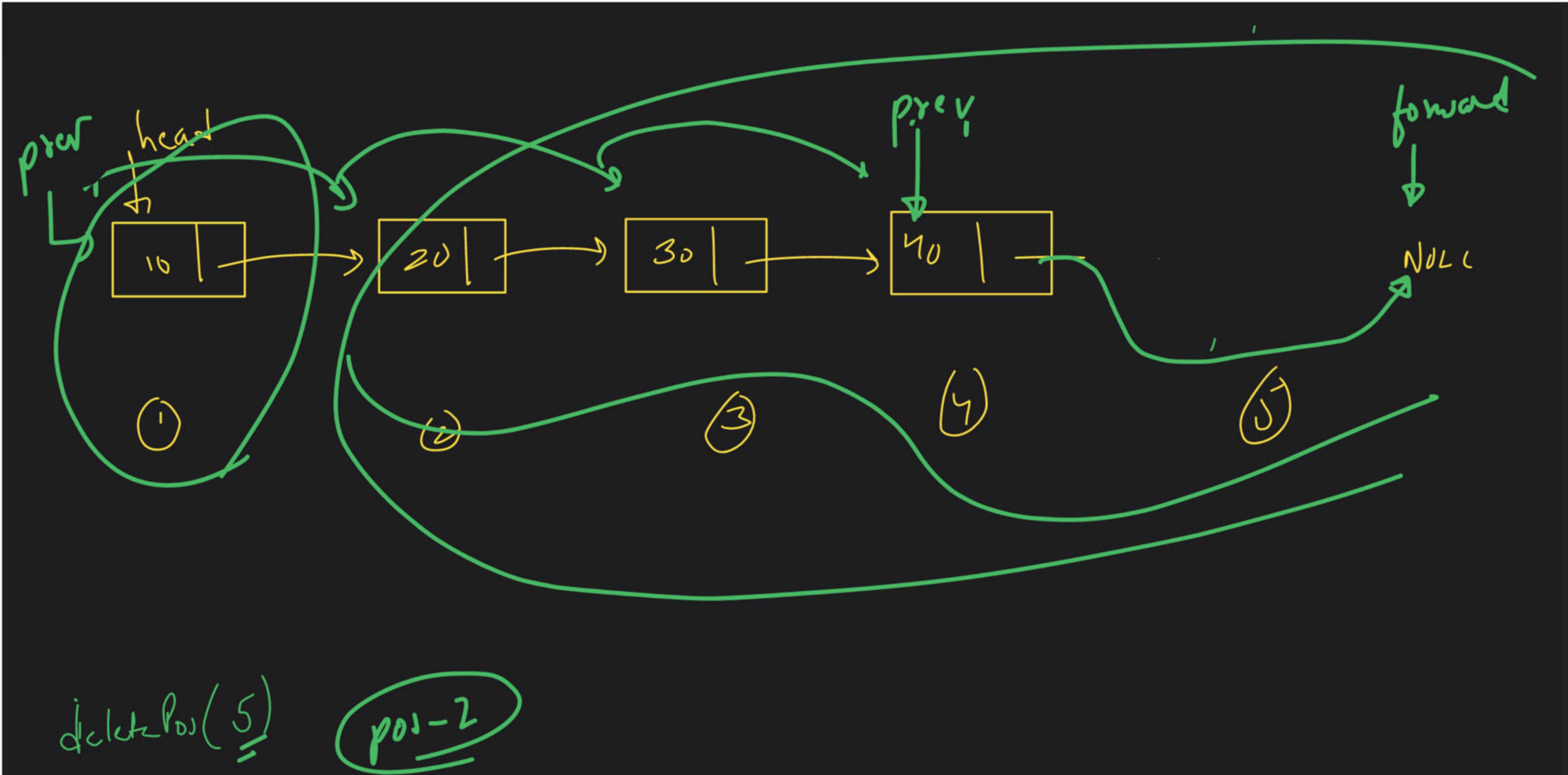
Node * forward = curr->next;

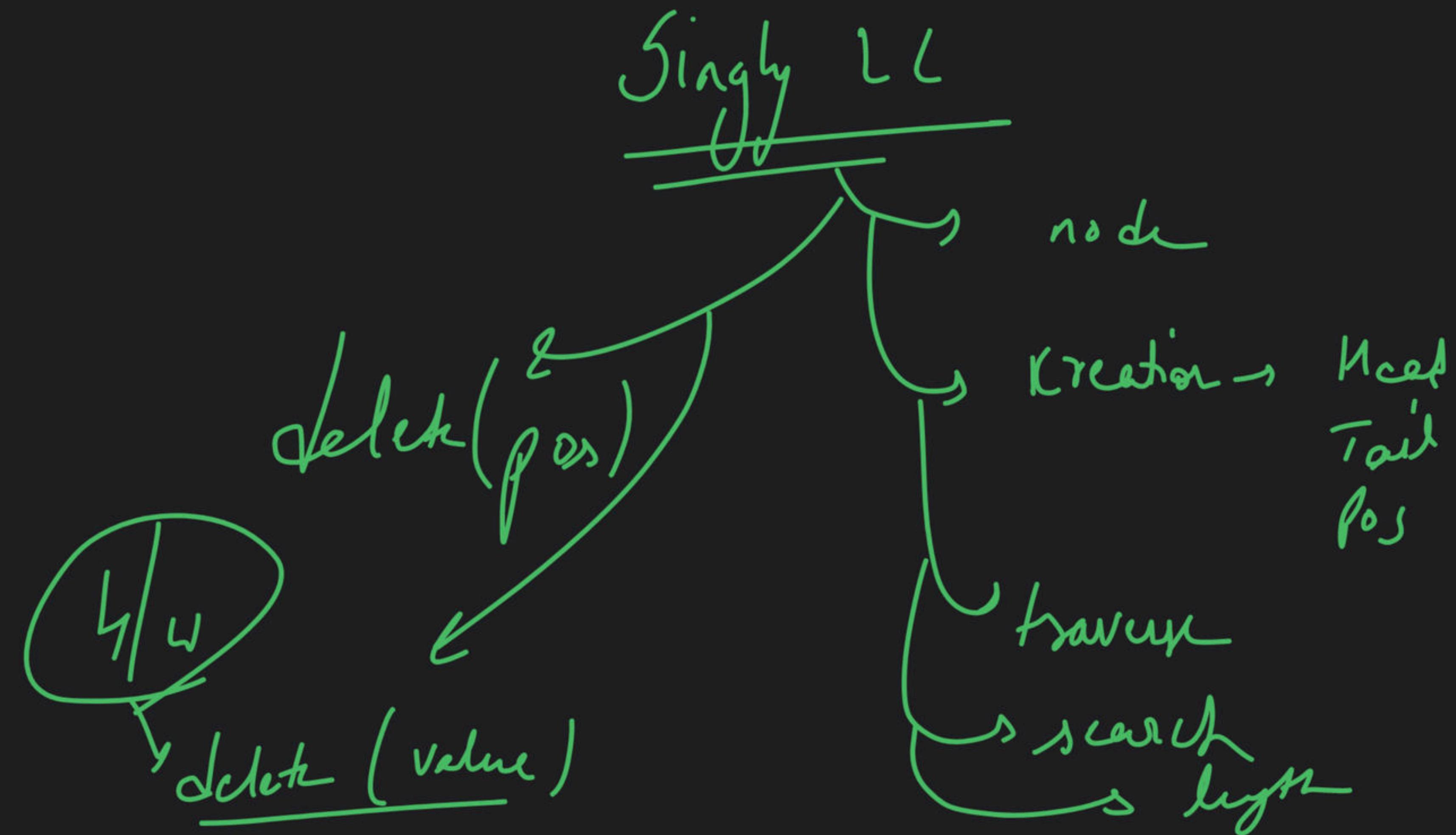


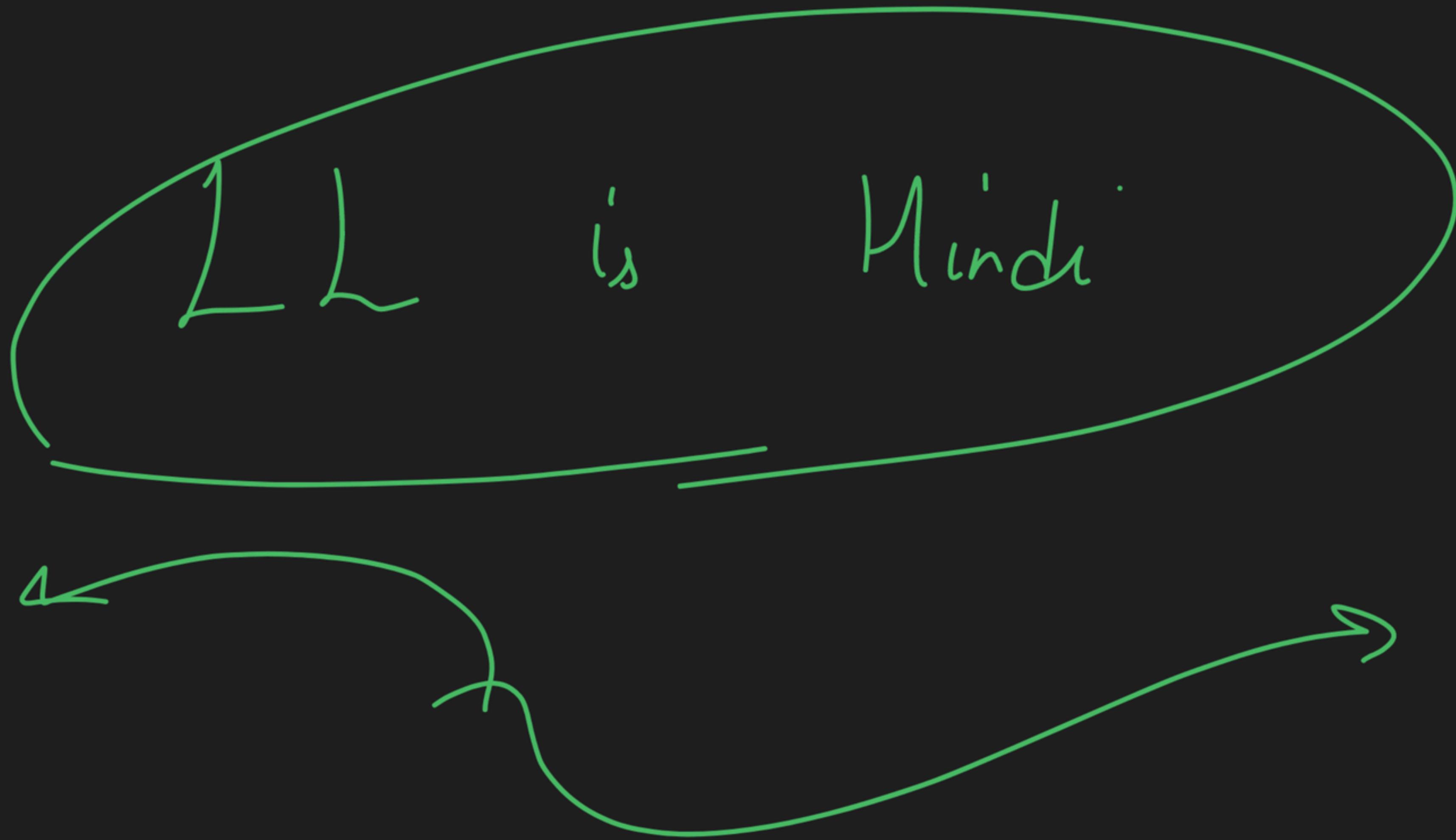
prev->next = forward
 curr->next = NULL
 delete curr;











Wed

8:30 fm

DLL

Love
Baffar

luv

.

.