

CS & IT ENGINEERING



Data structures &
Programming
Linked List
Lec- 05



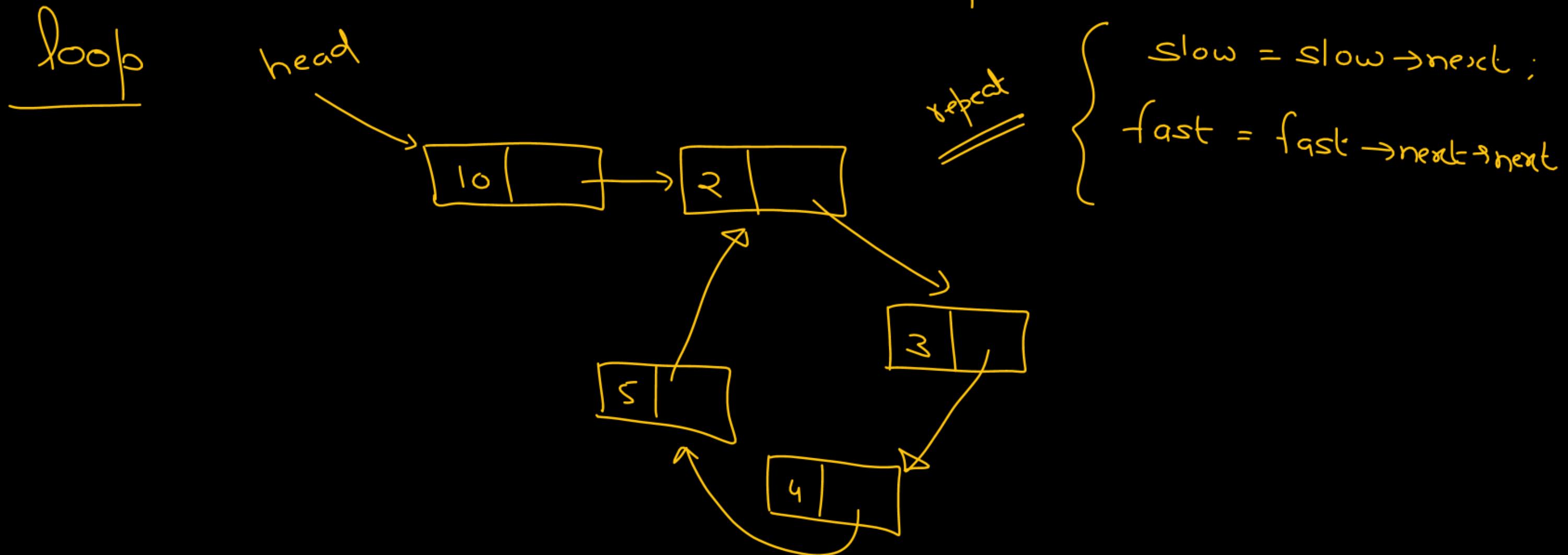
By- Pankaj Sharma sir

TOPICS TO BE COVERED



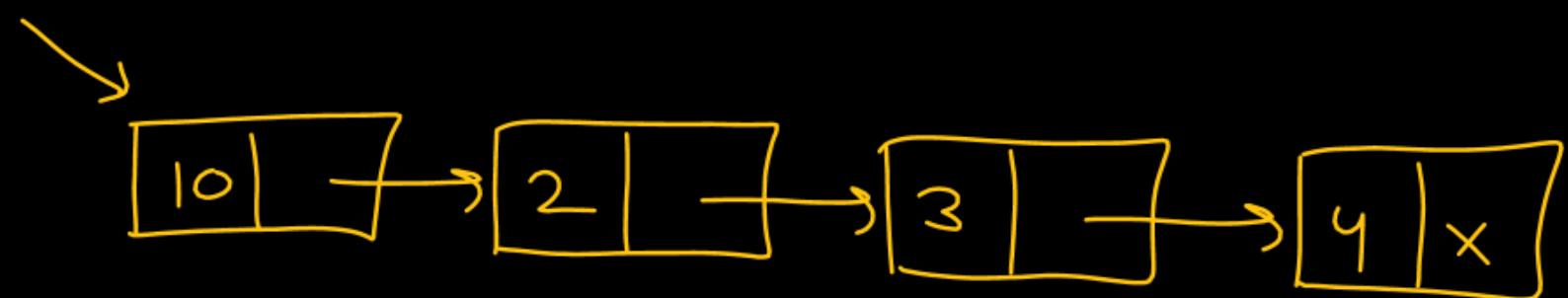
Detect loop in a l.l.

Given a l.l, find whether there is a loop or not.



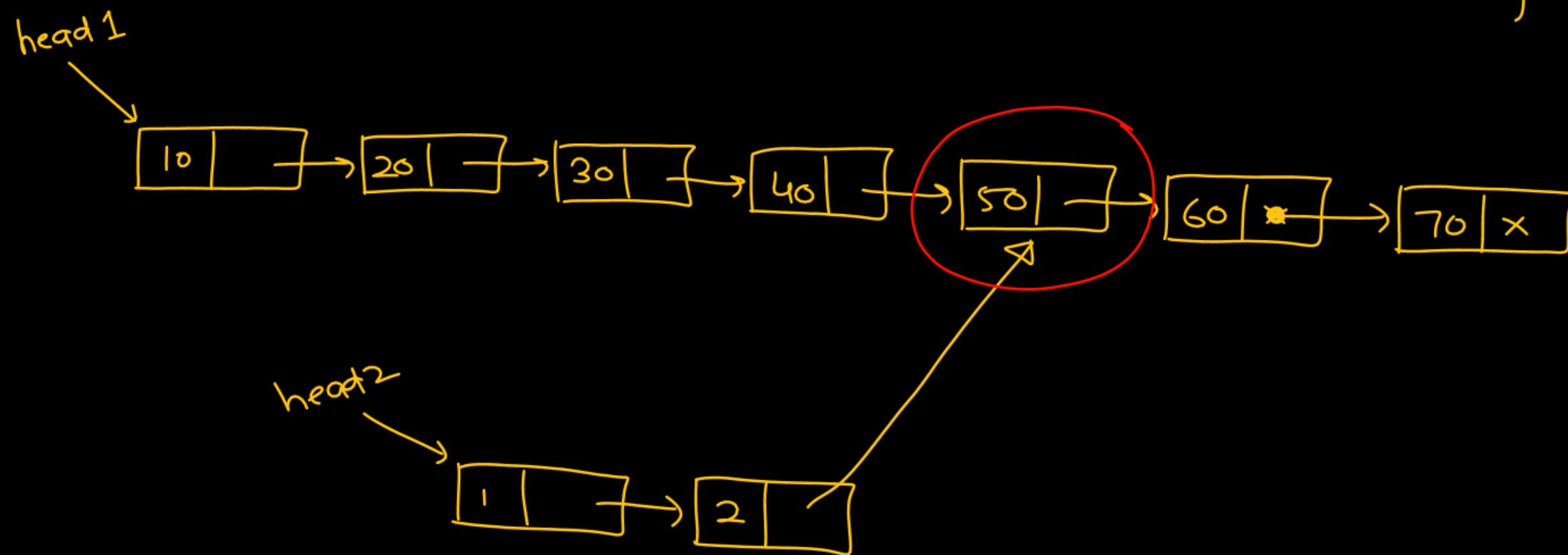
what if no loop

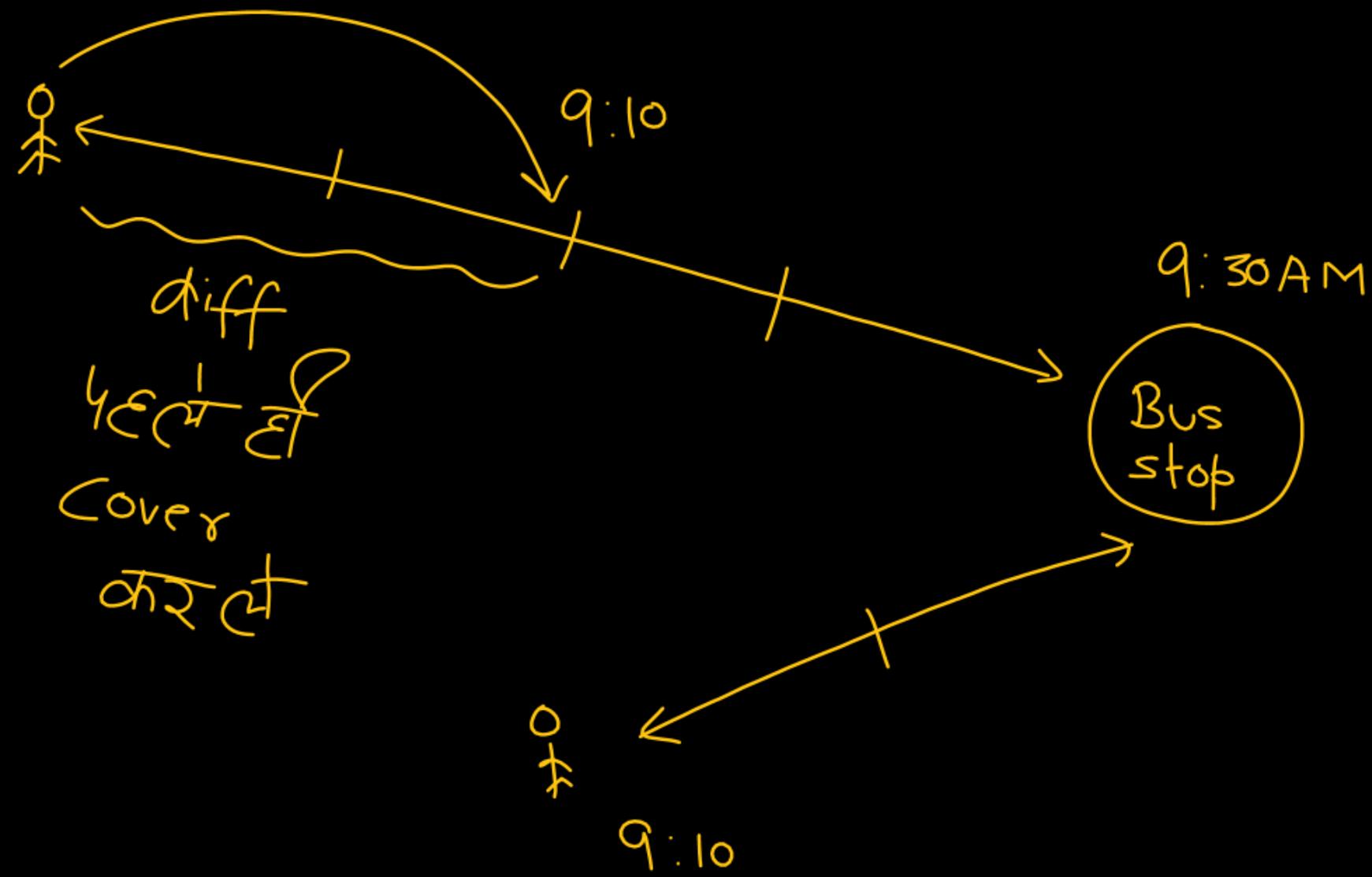
```
slow = fast = head;  
if (head == NULL)  
    return 0;
```

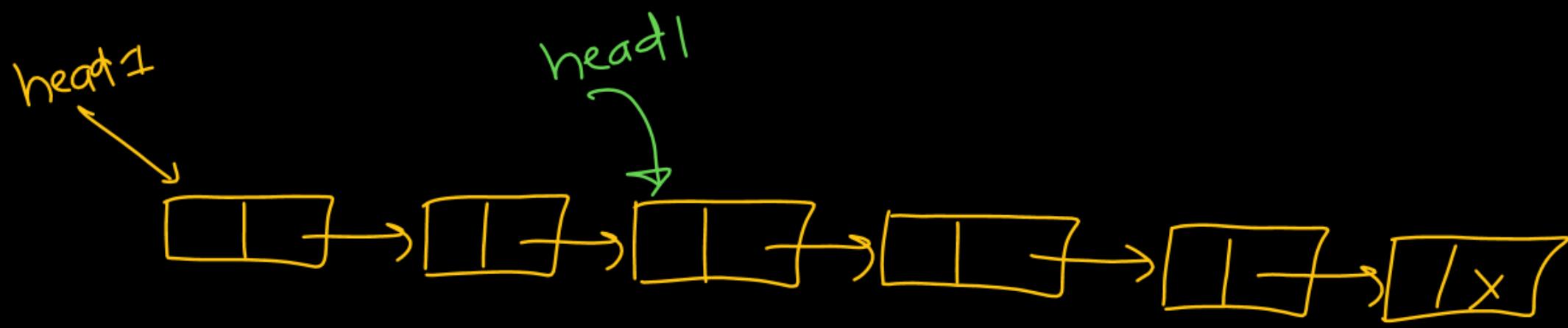


```
while (fast != NULL &&  
      fast->next != NULL)  
{  
    slow = slow->next;  
    fast = fast->next->next;  
    if (slow == fast)  
        return 1;  
}  
return 0;
```

Given 2 linked list , find there intersection point (if exists)



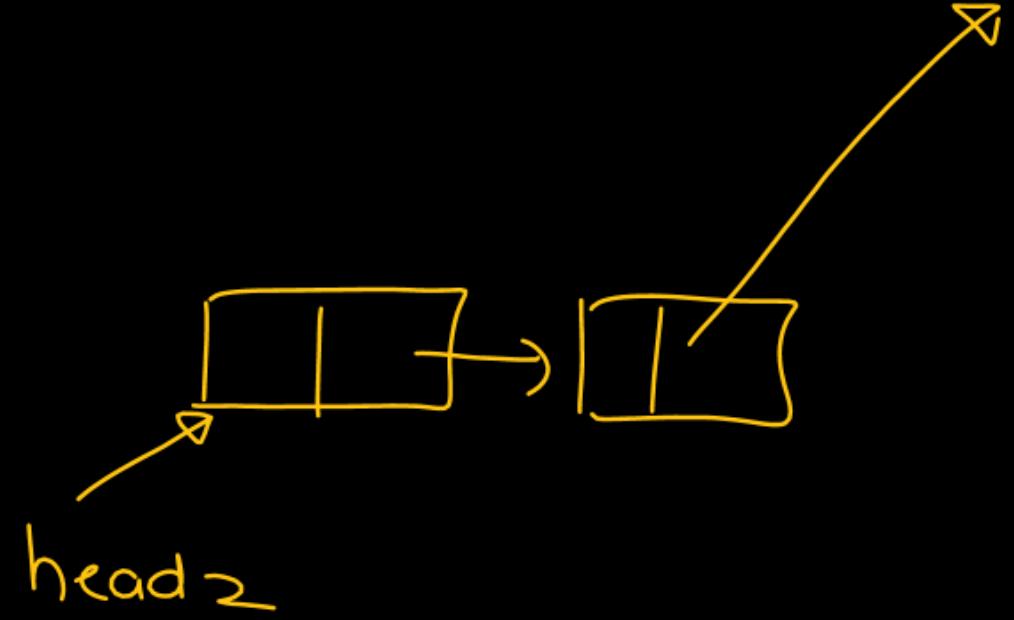




$$l_1 = 6$$

$$l_2 = 4$$

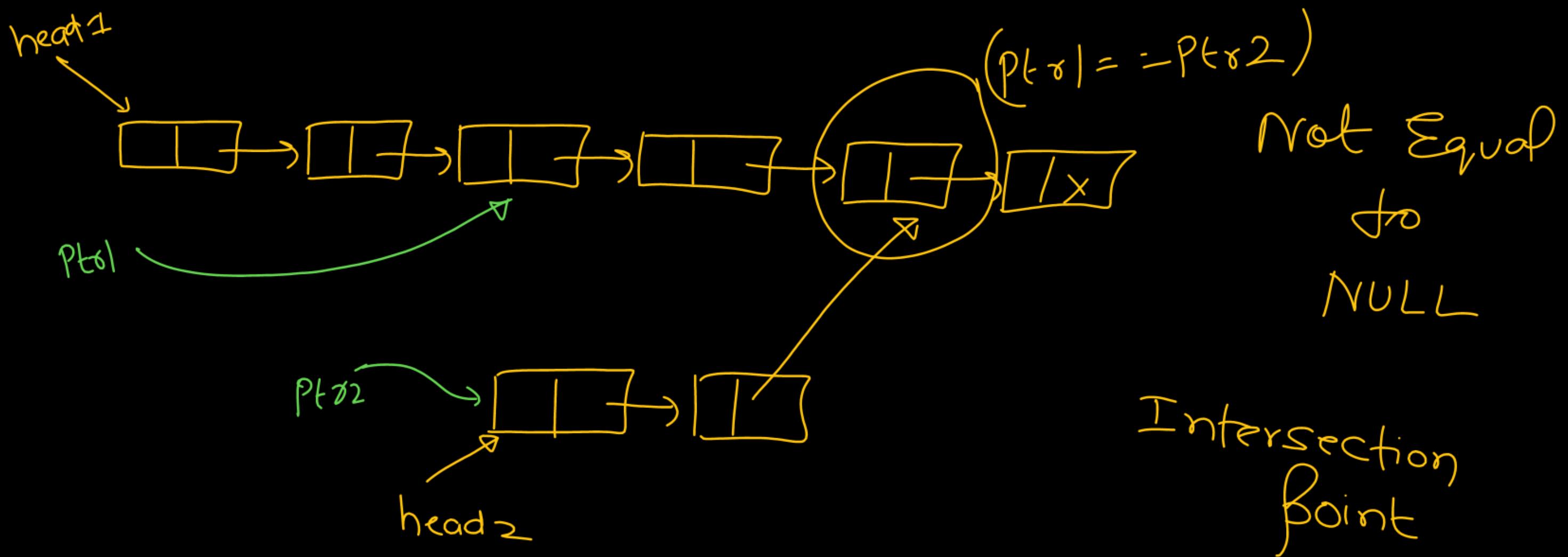
$$\begin{aligned} \text{diff} &= \text{abs}(l_1 - l_2) \\ &= 2 \end{aligned}$$

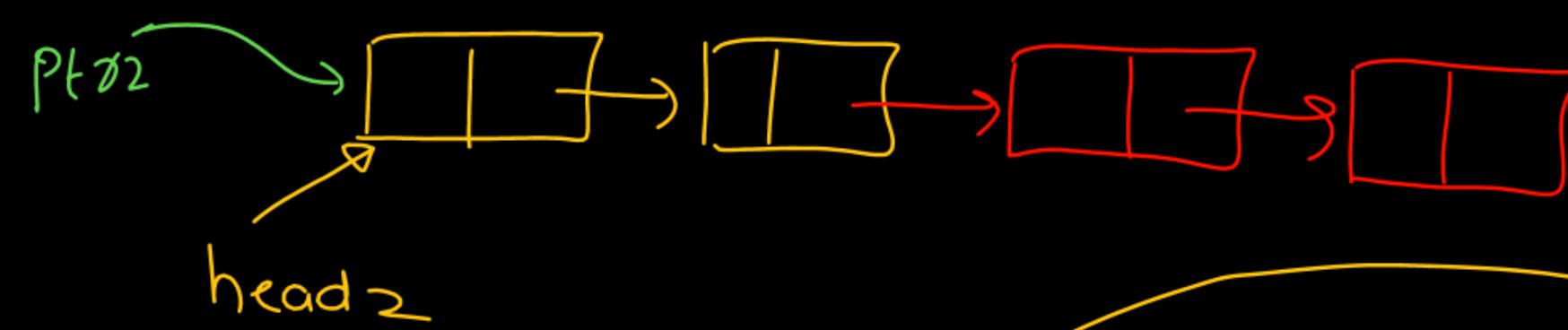
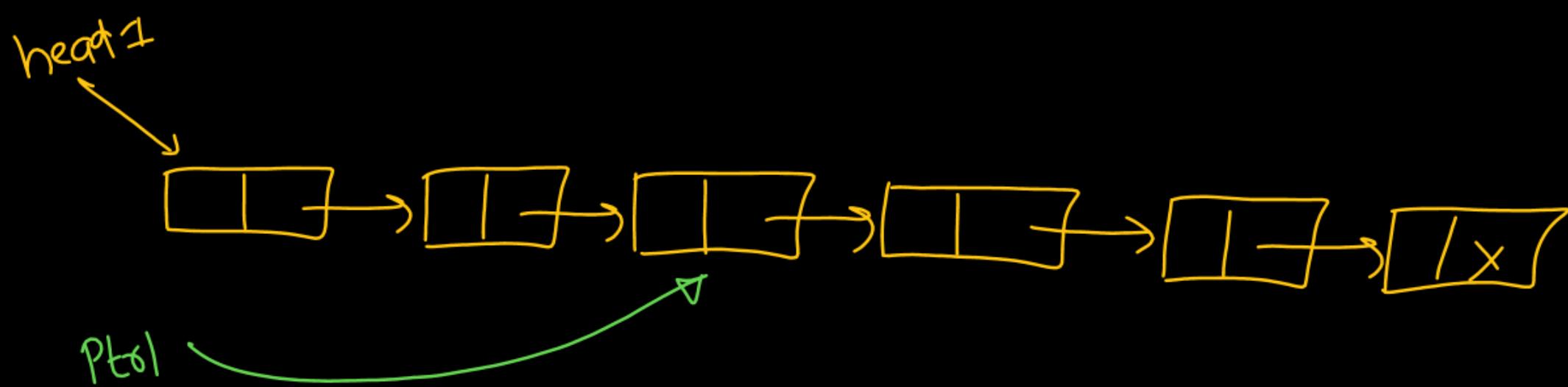


2 times (diff time)

head1 = head1 → next;

pointer

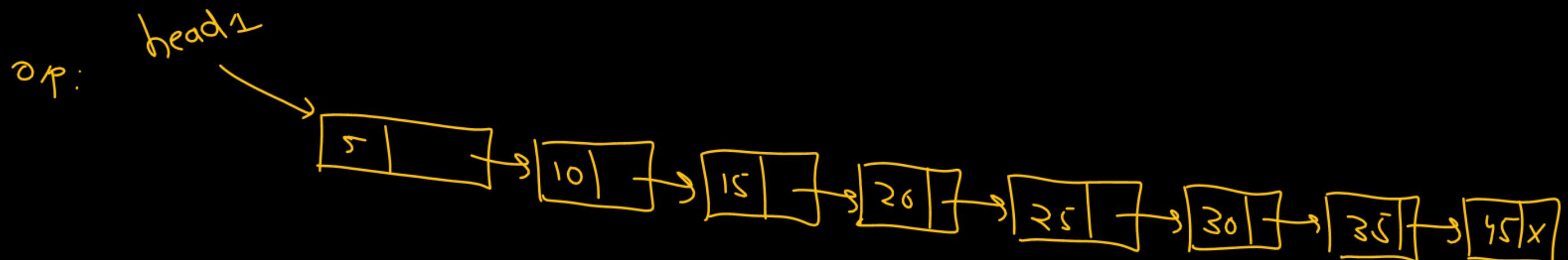
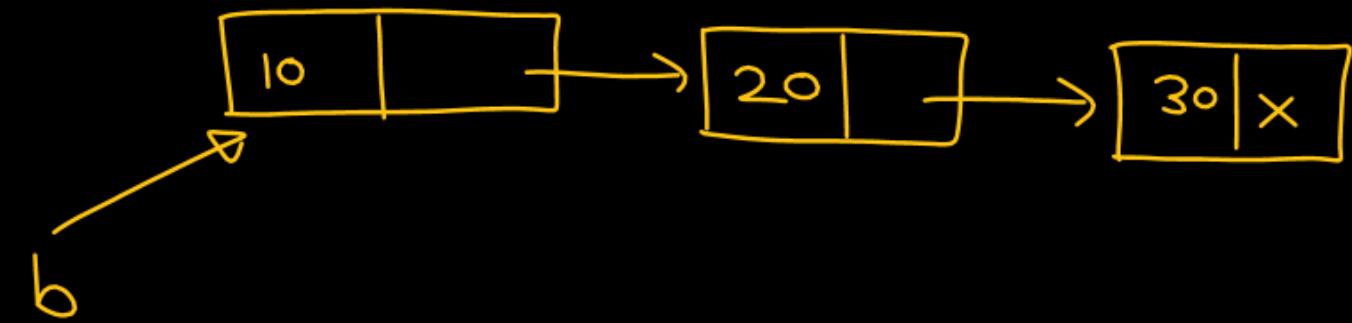
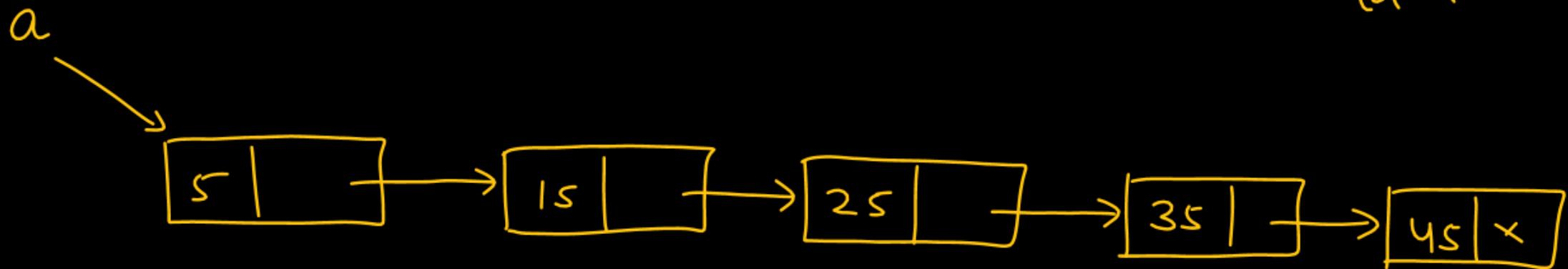




pt1 = pt2 = \dots =
NULL

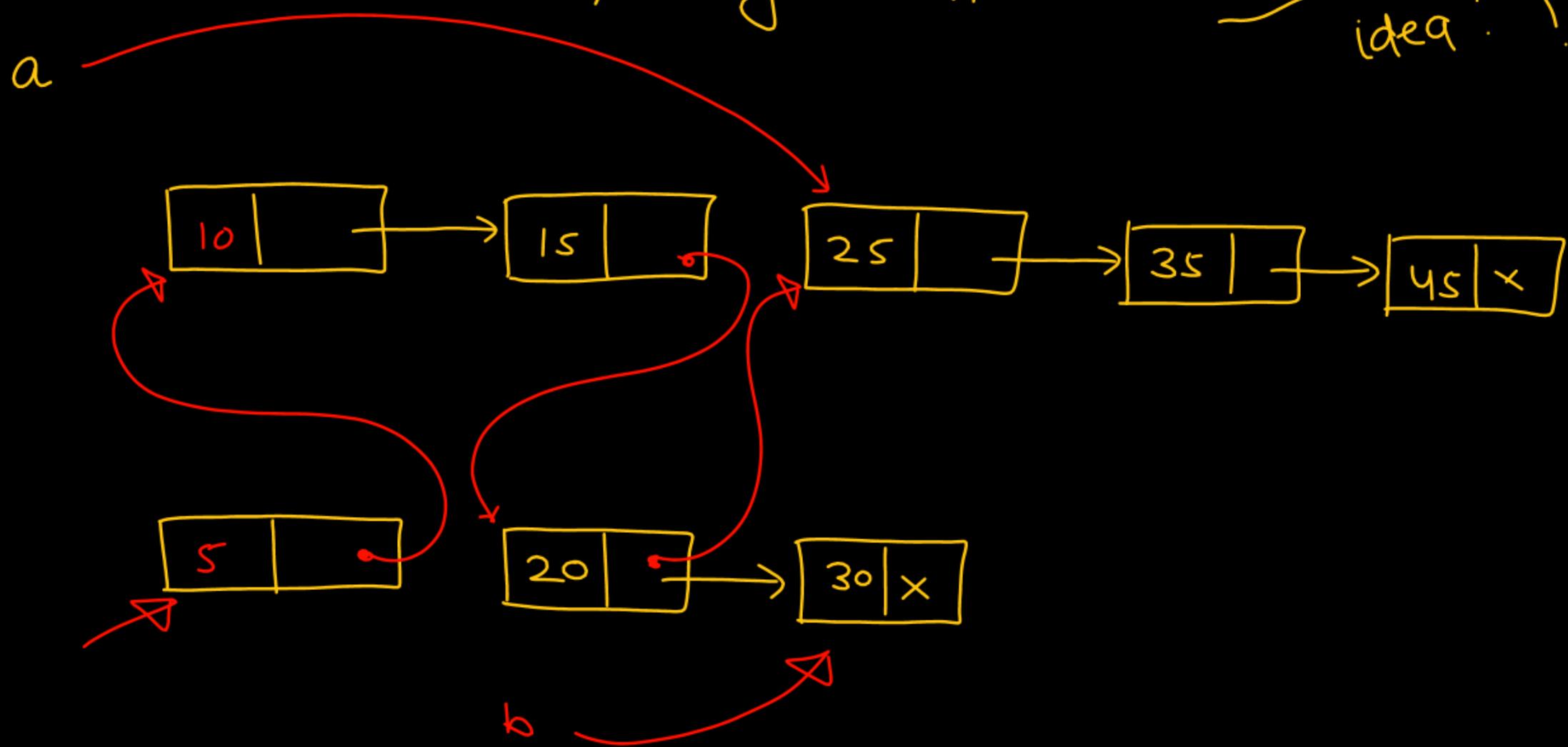
Given 2 sorted L.L., merge them.

2 min idea : ?

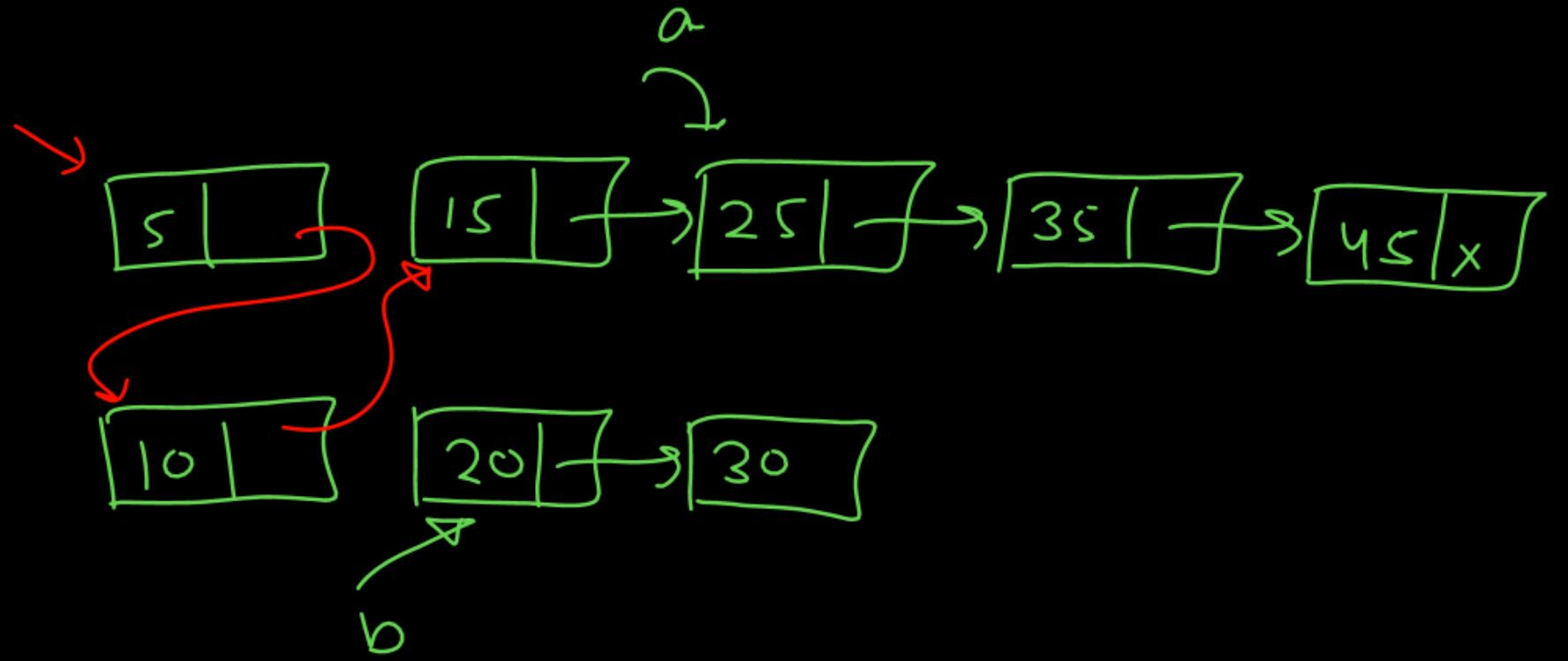


Given 2 sorted L.L., merge them.

2 min idea : ?



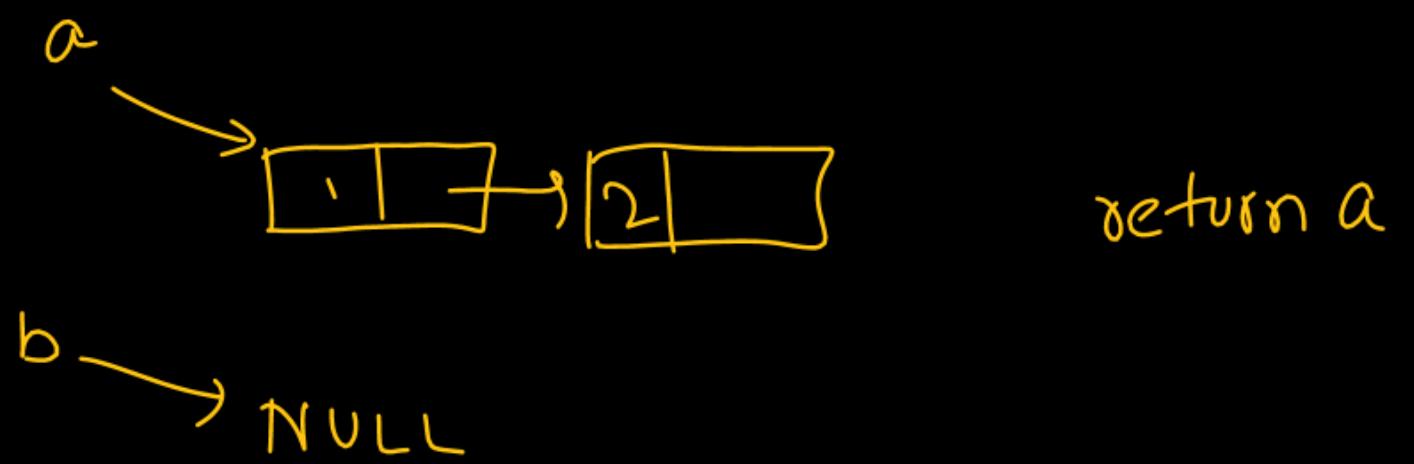
head
last
Connect on α



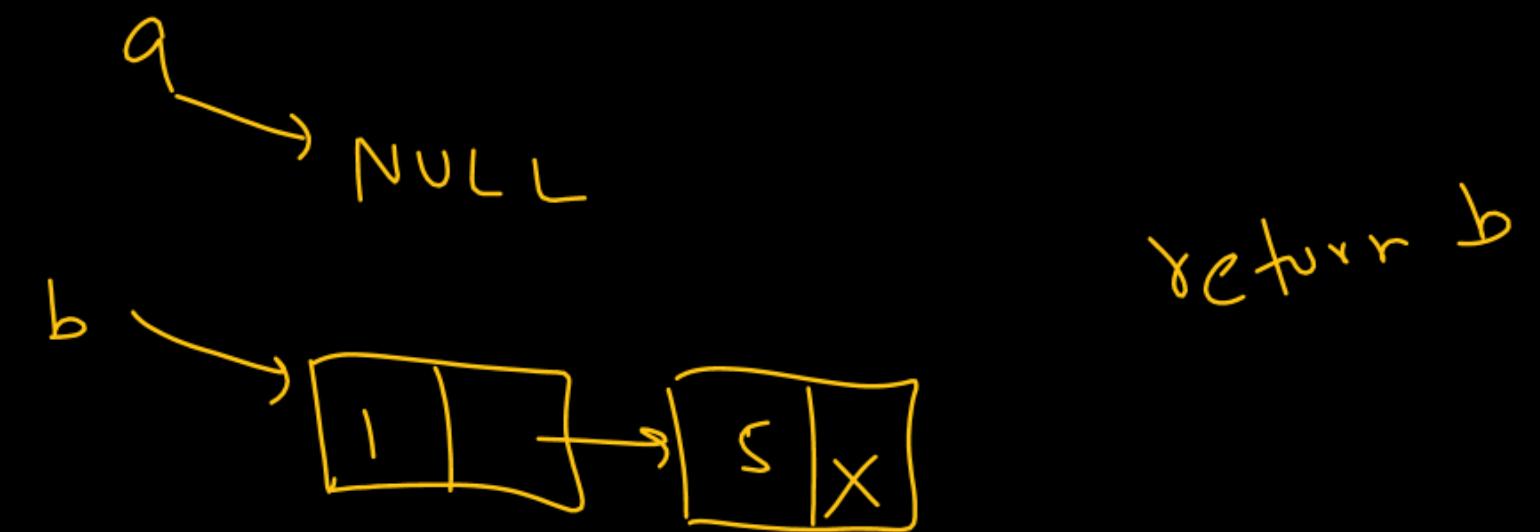
last
node
Connect ~~on β~~

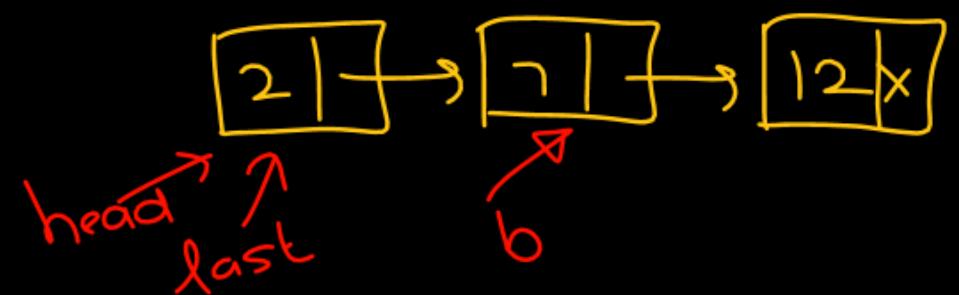


case 1



case 2:





first
node

```
if ( a->data < b->data )
```

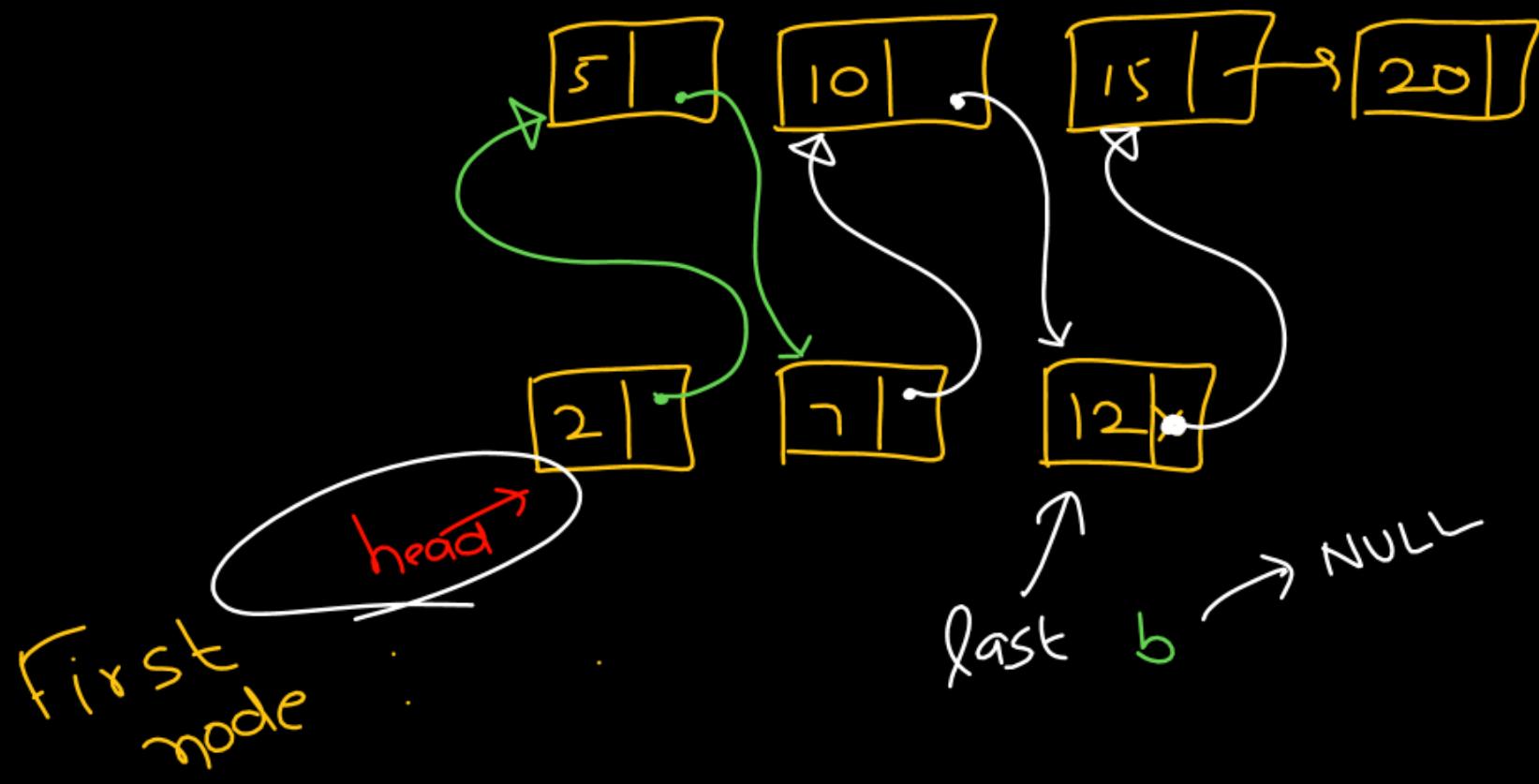
```
last = head = a ;  
a = a->next ;
```

```
}
```

```
{ else
```

```
last = head = b ;  
b = b->next ;
```

```
}
```



resultant LL को

if ($a \neq \text{NULL}$)

$\text{last} \rightarrow \text{next} = a;$

else

$\text{last} \rightarrow \text{next} = b;$

repeat while ($a \neq \text{NULL}$ & $b \neq \text{NULL}$)

{

 if ($a \rightarrow \text{data} \leq b \rightarrow \text{data}$)

 // resultant LL के एंड
 में append करना है

$\text{last} \rightarrow \text{next} = a;$

$\text{last} = a;$

$a = a \rightarrow \text{next};$

 }

 else { // resultant LL के एंड
 में append

$\text{last} \rightarrow \text{next} = b;$

$\text{last} = b;$

$b = b \rightarrow \text{next};$

 }

}

- 1) Traversal
- 2) Search
- 3) Insertion
- 4) Deletion
- 5) Last node
- 6) Second last node
- 7) Middle element
- 8) Loop
- 9) Intersection point
- 10) Print reverse

Review

- 11) L.L. reverse
- 12) Merge 2 sorted LL
- 13) length of LL.

- 6. Types of LL
- 7. PYQs

