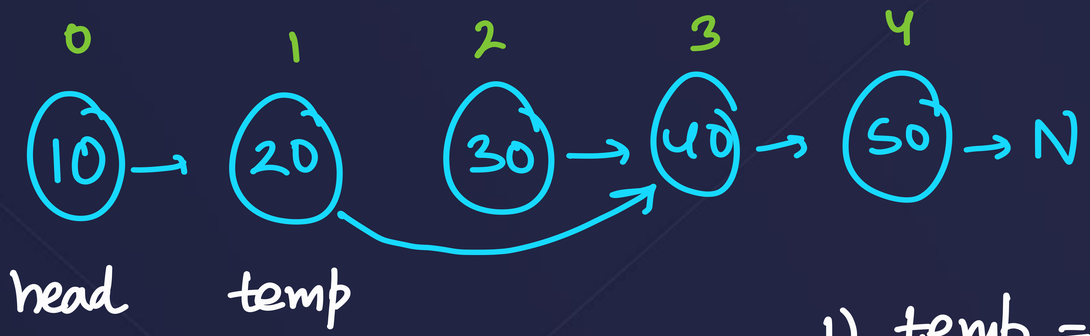
Linked List

Part – 2

Raghav Garg

Ques: Delete Node in a Linked List [Leetcode - 237]

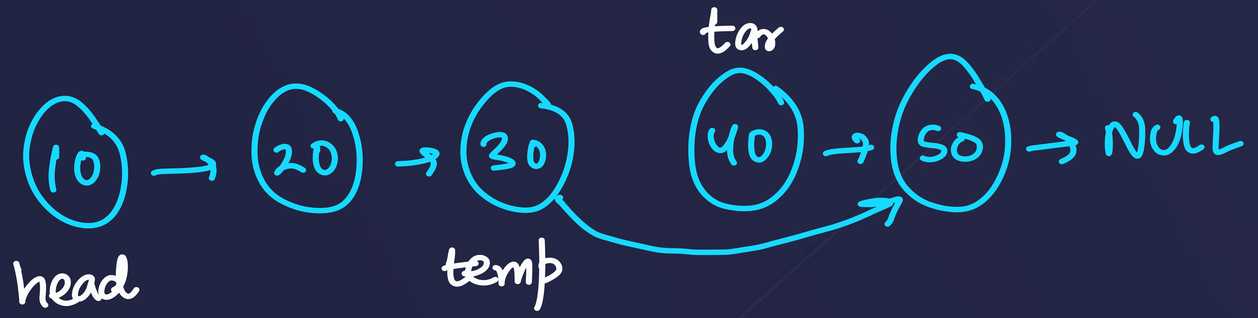
delete → head, tail, delete at idx



delete at (2)

1) temp = head
2) traverse temp till (idx-1)
3) temp → next = temp → next → next

Ques: Delete Node in a Linked List [Leetcode - 237]



delete (head, target)

↓

delete (head, targetVal)

1) temp = head

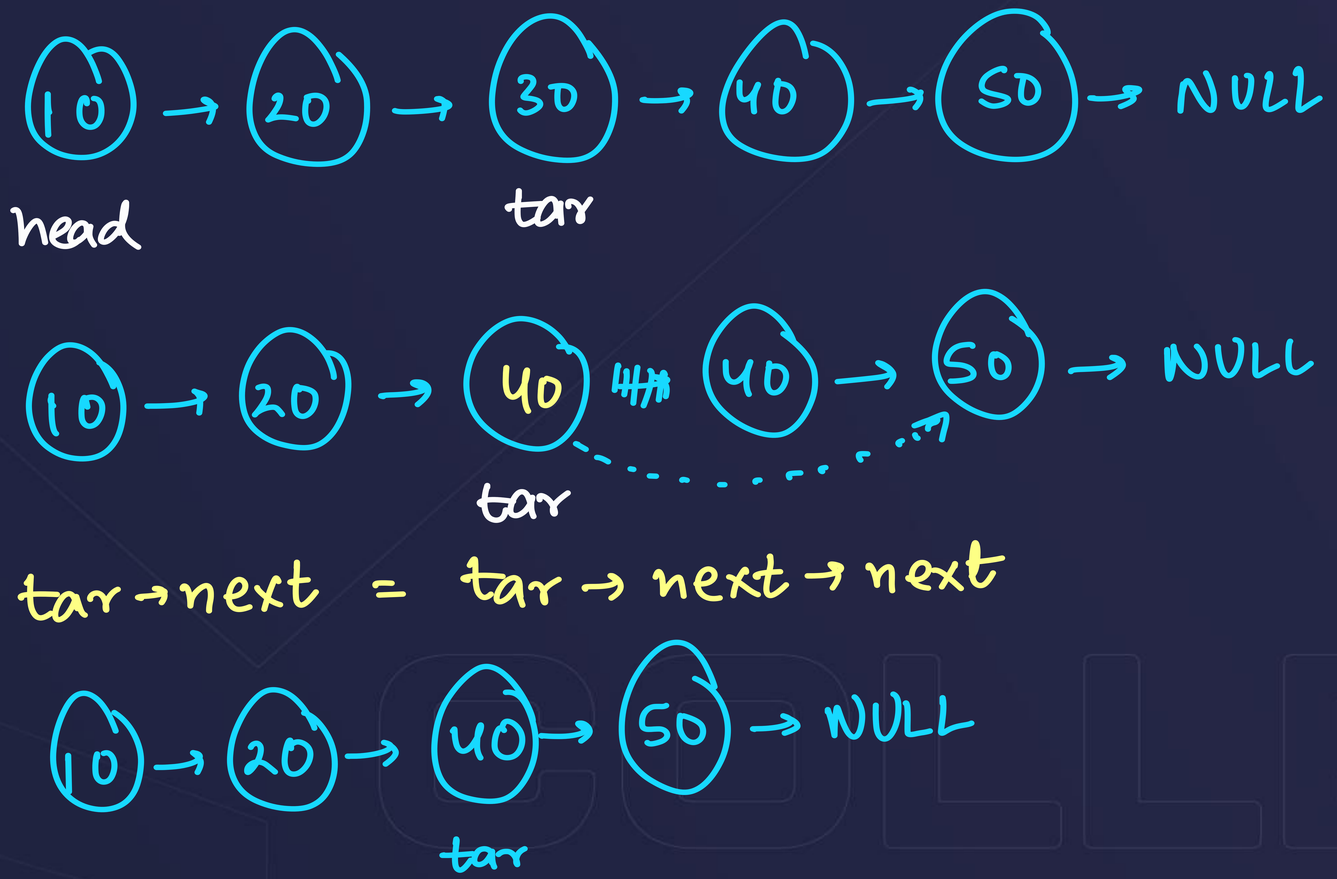
2) traverse temp till

temp → next = tar → break;

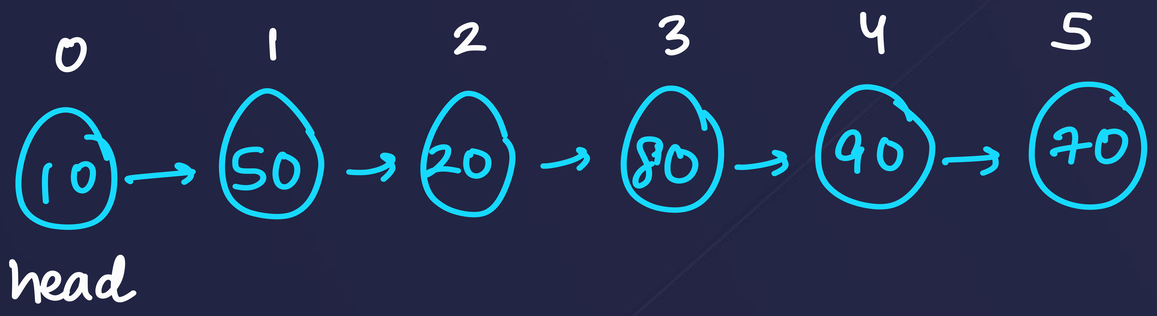
3) temp → next = temp → next → next

Ques: Delete Node in a Linked List [Leetcode - 237]

tar → val = tar → next → val



Ques: Middle of the Linked List [Leetcode - 876]



Length → 'n' → '0 to n-1'

For Odd Length → $\frac{n}{2}^{th}$ index → middle

For Even Length → $\frac{n}{2}-1^{th}$ & $\frac{n}{2}^{th}$

↓ left middle

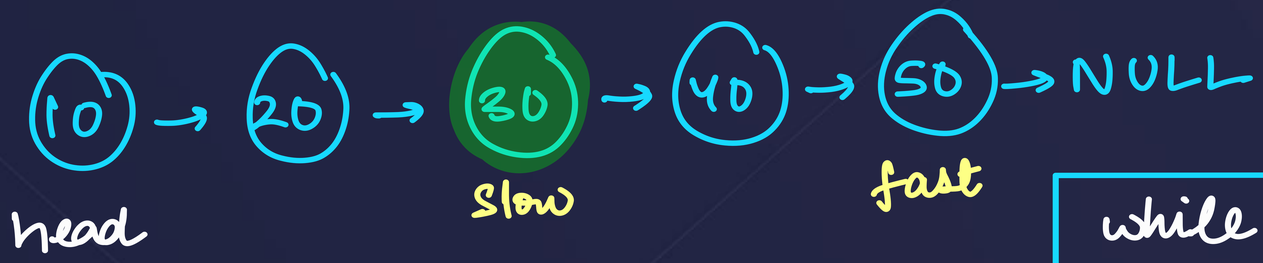
right middle

Ques: Middle of the Linked List [Leetcode - 876]

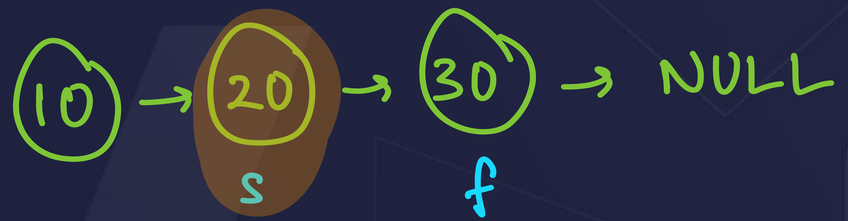
What interviewer wants ??

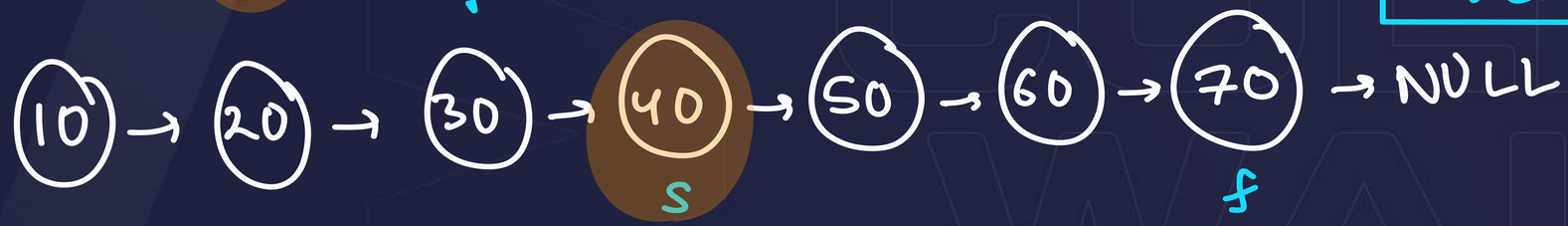
↳ to solve this ques in '1 pass'

Odd Length



→ Slow & fast pointer technique

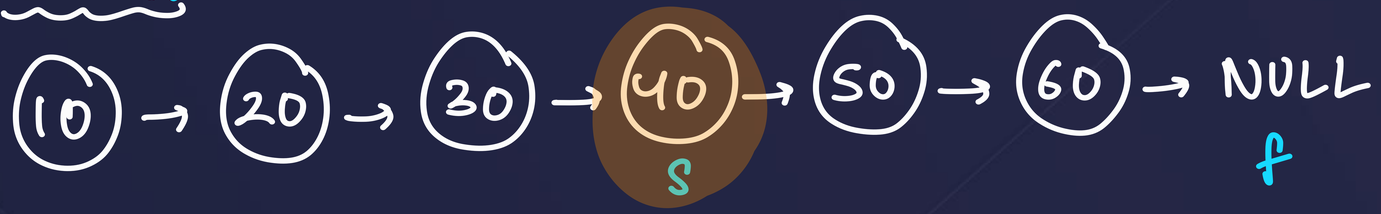


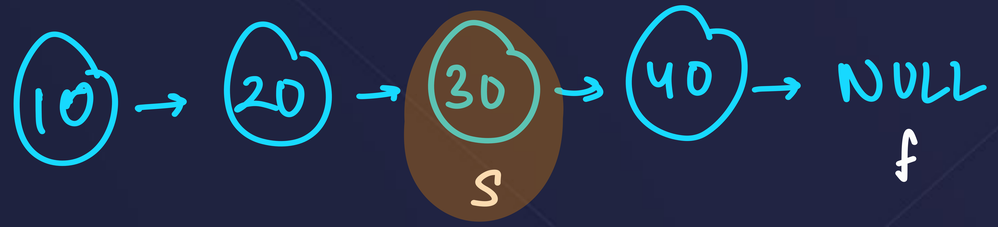



```
while (fast → next != NULL){
    slow = slow → next;
    fast = fast → next → next;
}
return slow;
```

Ques: Middle of the Linked List [Leetcode - 876]

Even Length List






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

Ques: Middle of the Linked List [Leetcode - 876]

```
ListNode* slow = head;
ListNode* fast = head;
while(fast->next!=NULL && fast!=NULL){
    slow = slow->next;
    fast = fast->next->next;
}
return slow;
```

If for any node 'temp',
we want to check/get the value
of temp→next then PLS FIRSTLY
CHECK IF TEMP ITSELF !=NULL

Ques: Middle of the Linked List [Leetcode - 876]

```
ListNode* slow = head;
ListNode* fast = head;
while(fast!=NULL && fast->next!=NULL){
    slow = slow->next;
    fast = fast->next->next;
}
return slow;
```

if (con1 && con2)

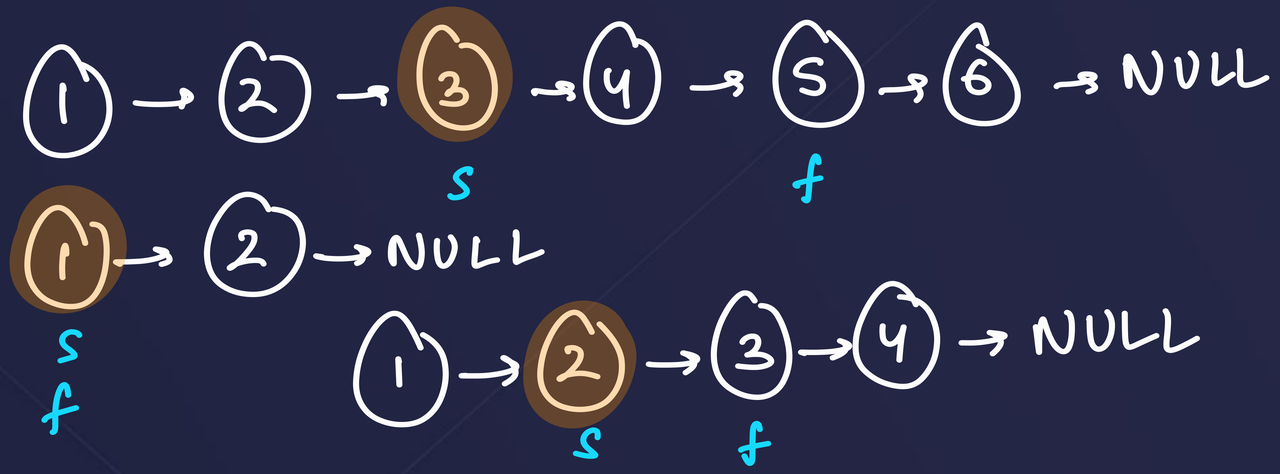
↓

F

Short ckt

Ques: Middle of the Linked List [Leetcode - 876]

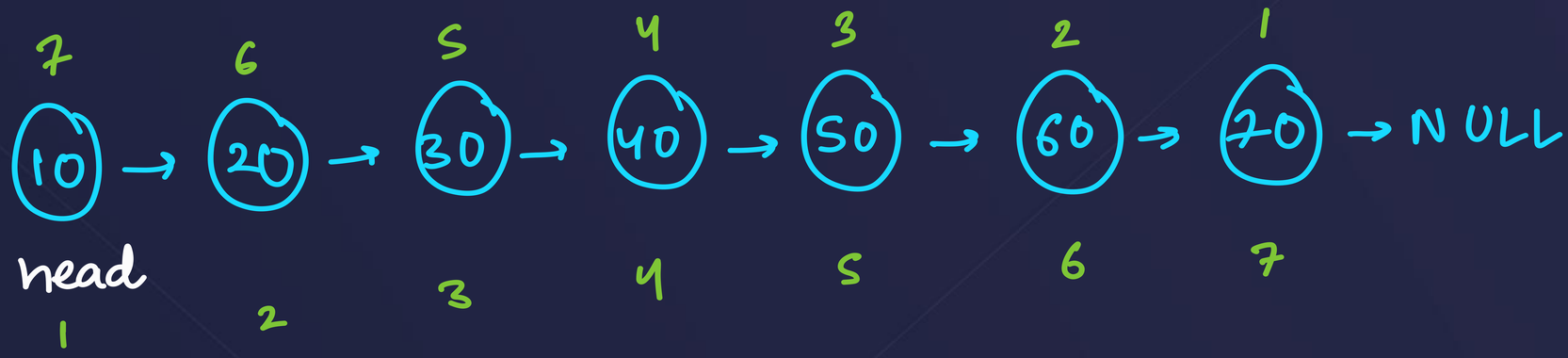
How to find left middle ??




```
while (fast->next->next != NULL ) {
    slow = slow->next;
    fast = fast->next->next;
}
```

3

Ques: Remove Nth Node from End of List [Leetcode - 19]



find out the 3rd node from last ?

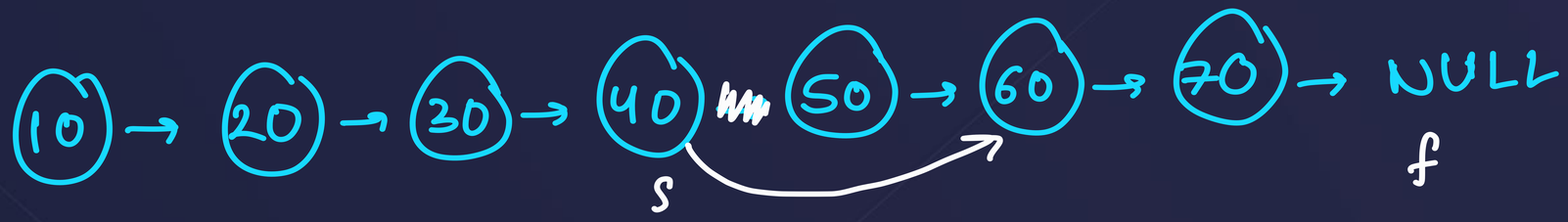
→ len = 7

n^{th} from end = $(len - n + 1)^{th}$ node from start

Find/

Ques: Remove Nth Node from End of List [Leetcode - 19]

Using Slow & Fast Pointers :



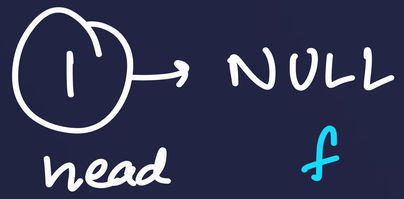
n = 3

↓

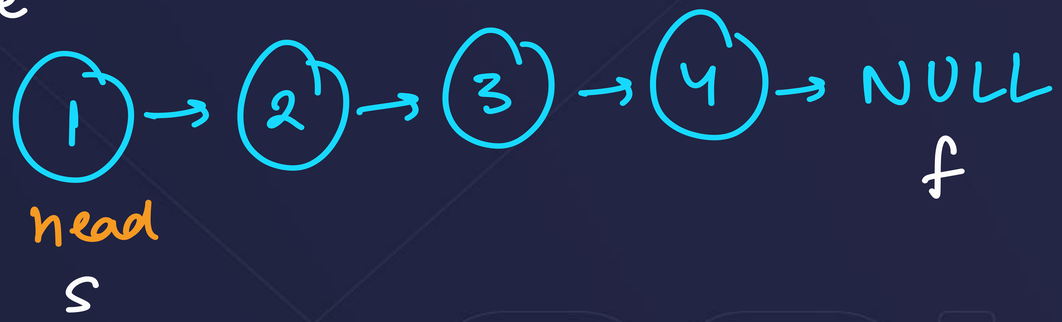
f → 'n+1' steps aage le jao

```
for(int i=1; i<=n+1 ; i++){
    fast = fast->next;
}
while (fast != NULL){
    slow = slow->next;
    fast = fast->next;
}
slow->next = slow->next->next;
```

Ques: Remove Nth Node from End of List [Leetcode - 19]

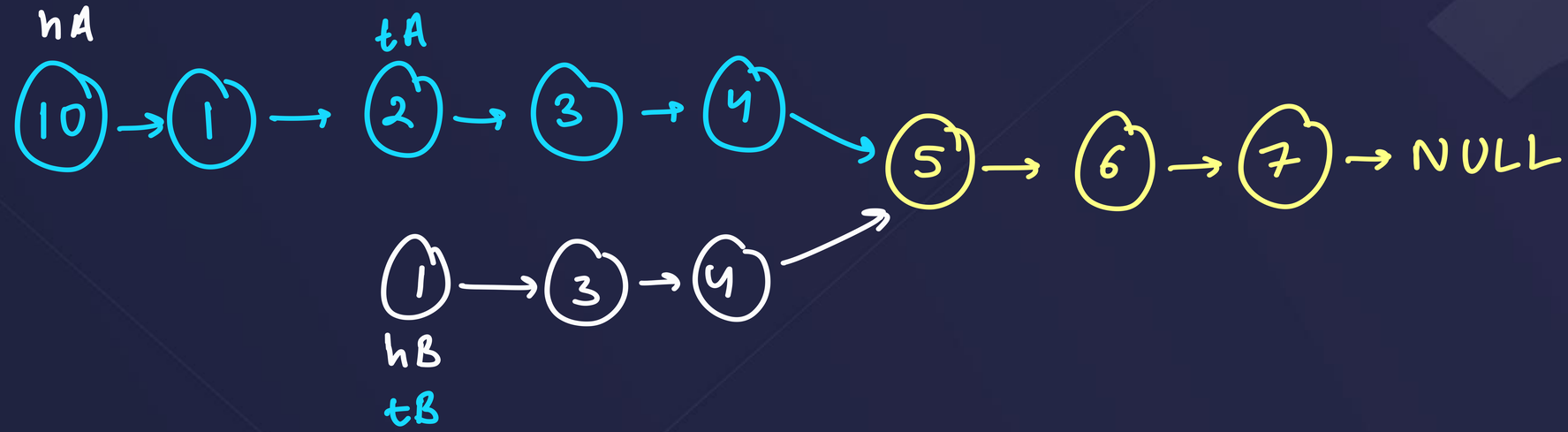


fast → (n+1) aage



n = 4th node from last delete

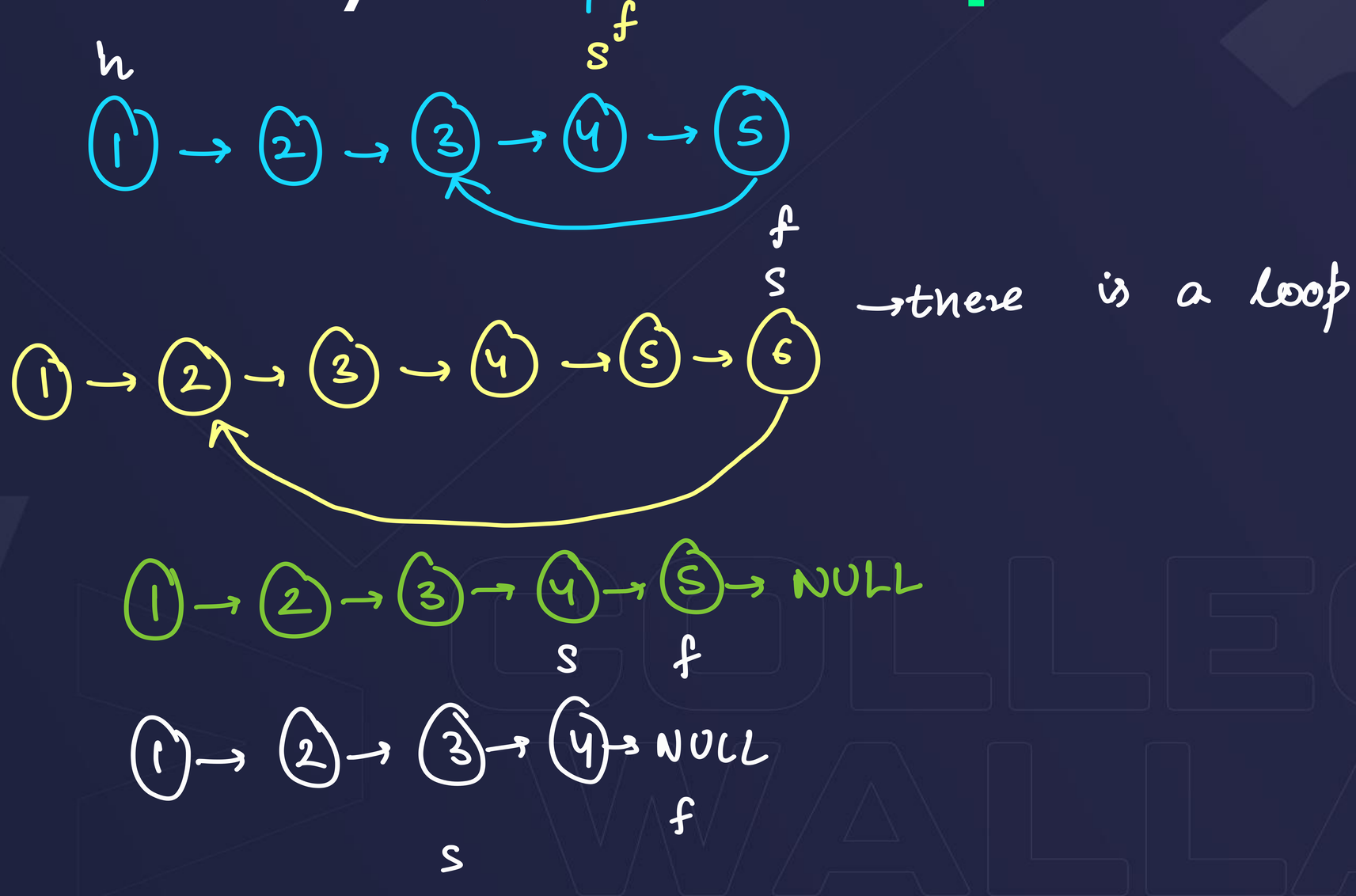
Ques: Intersection of two Linked Lists [Leetcode - 160]



Hint : Find the lengths of both lists

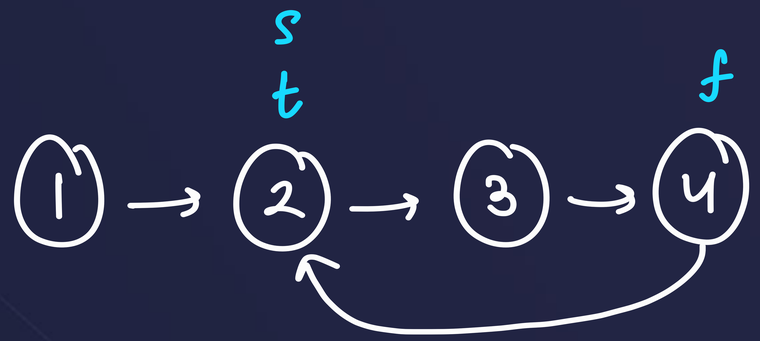
Steps → Larger list ke temp ko aage leaao such that |lA-lB|

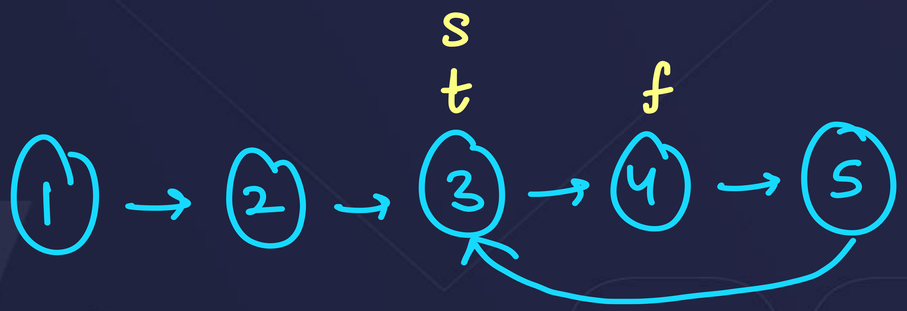
Ques: Linked List Cycle/Loop [Leetcode - 141]



Ques: Linked List Cycle II

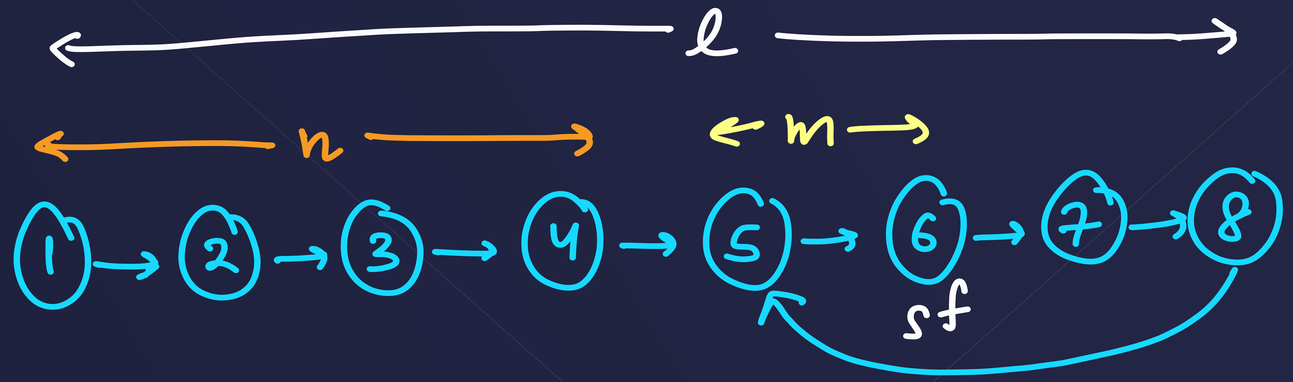
[Leetcode - 142]





Why is the slow & fast algo working ? (PROOF)

1) Slow is moving at 1 & Fast is moving at 2.



Distance travelled by slow
$= n + m$

Distance travelled by fast
$= l + m$

$\Rightarrow 2(n+m) = l+m$

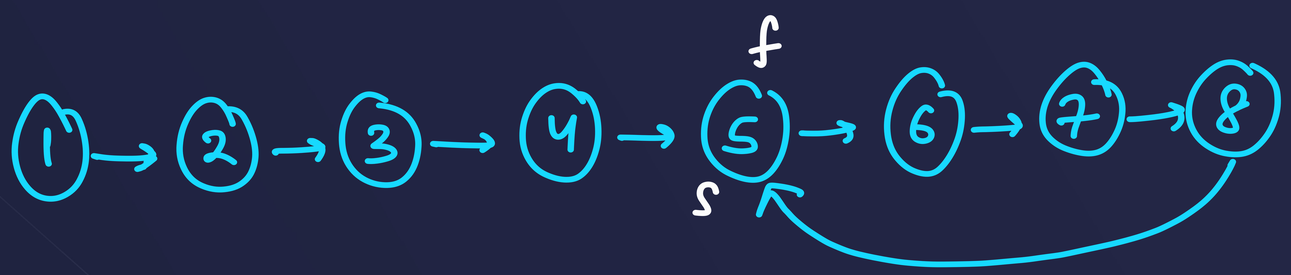
$\Rightarrow 2m + 2n = m + l$

$\Rightarrow \boxed{m = l - 2n}$ → this proofs

s → 1 2 3 4 5

f → 2 4 6 8 10

if 9 can find 'm' such sbf are together, yes
our method will work



slow → 1x

fast → 2x, 3x, 4x, ...

'Maza aa gaya'

Next Lecture

More problems on Linked Lists!

Learning about the types of Linked Lists!

THANK YOU