

LinkedList

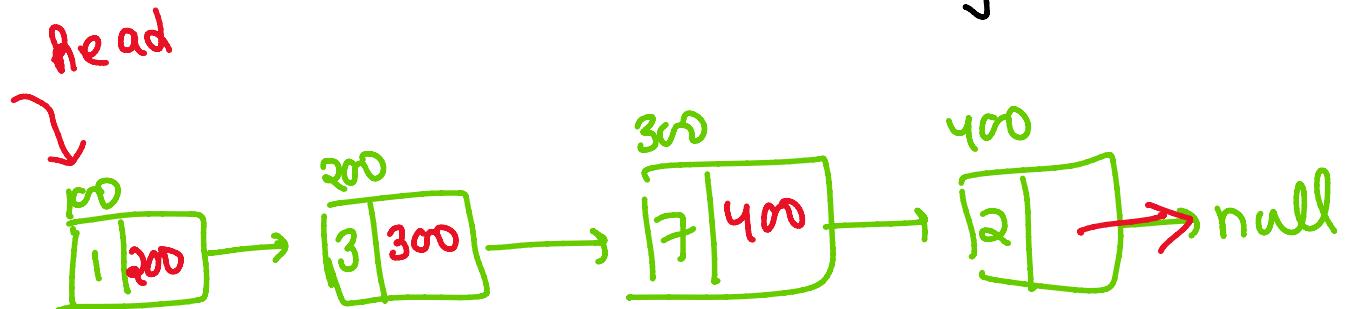
02 September 2023 13:07

Arrays
Set
Map
Matrix
LinkedList
Stack
Queue
Tree
Heap
Graphs

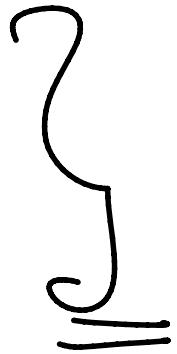
→ collection of nodes.

Node
[data] next

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



- insert
- delete
- search



Insertion in Linkedlist

02 September 2023 13:13

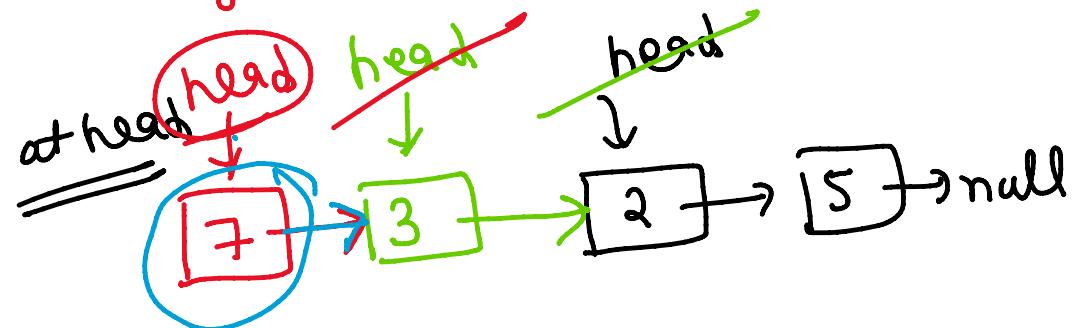
✓ at head

✓ at tail

→ at any given position

node data = 3

7



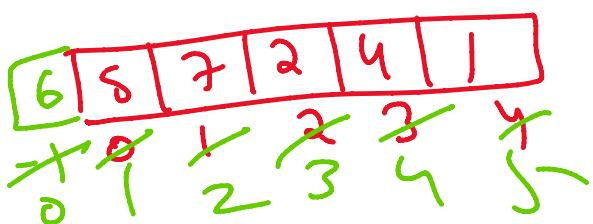
O(1)

1. Create a new node

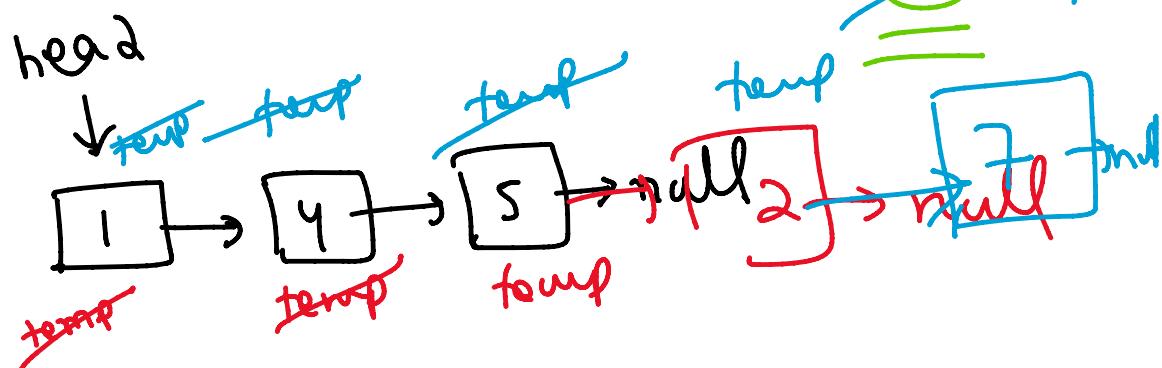
2. new node's next will point to head

3. Update head to new node

O(N)



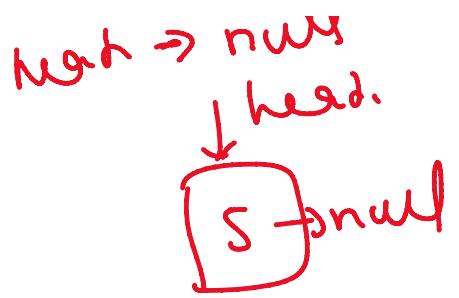
at tail



1. Iterate till next is null
2. Initialise temp's next to new node.

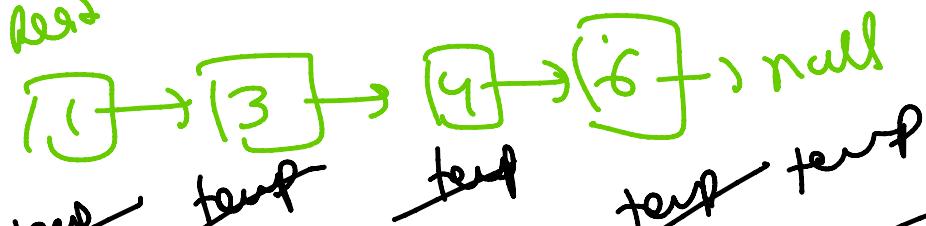
next → null
, head.

(5)



print

Recd



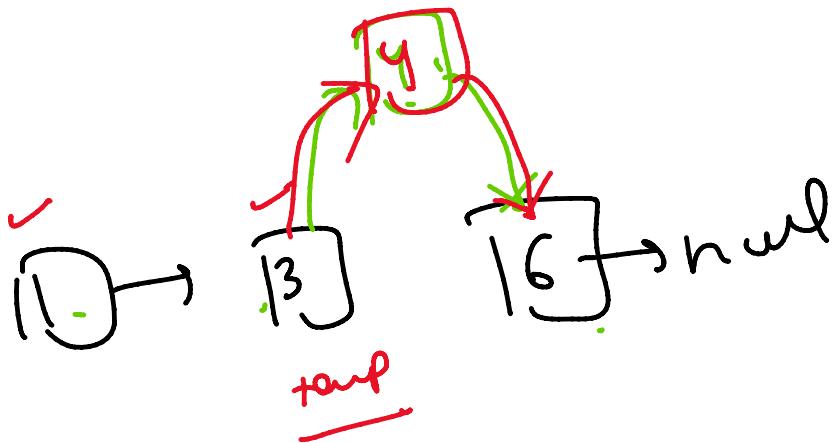
temp

temp temp

temp temp

$1 \rightarrow 3 \rightarrow 4 \rightarrow 6 \Rightarrow$

\Rightarrow
position 3



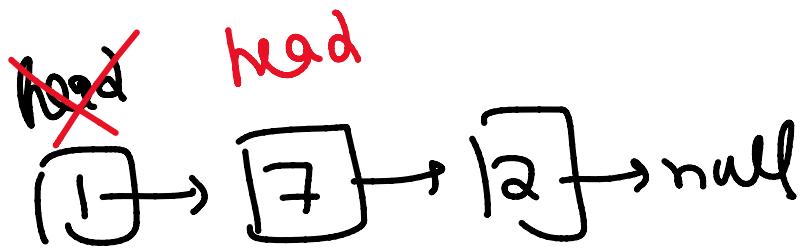
temp

Deletion in linkedlist

02 September 2023 13:34

- at head
- at tail
- at any given position

at head

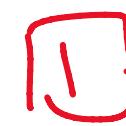


$\text{head} = \text{head} \rightarrow \text{next};$

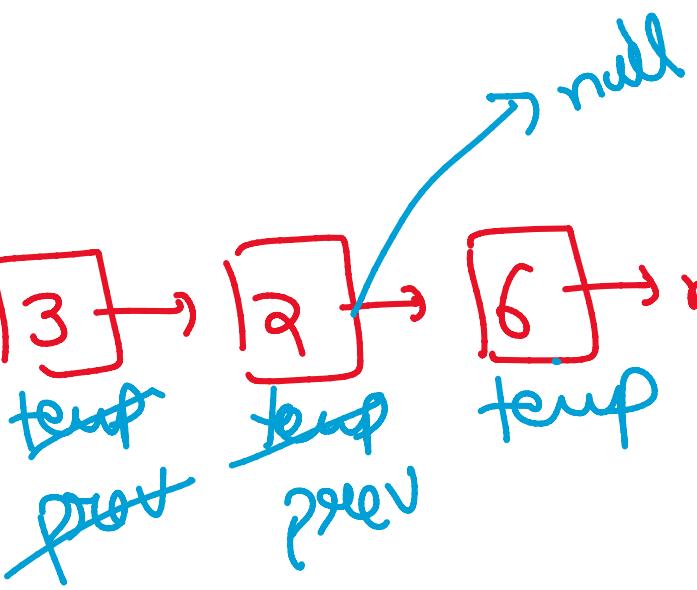
at tail

pos

head



prev

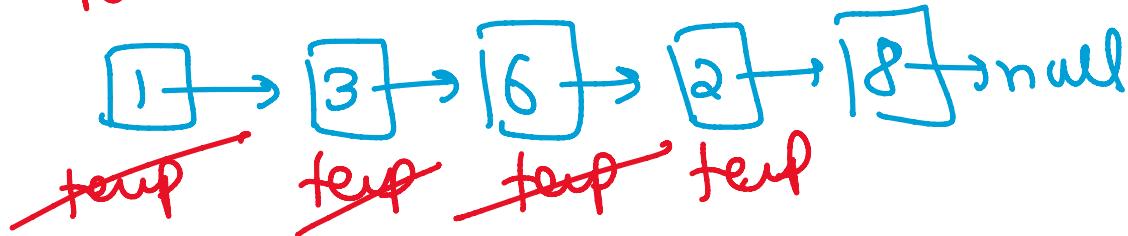


Searching in linkedlist

02 September 2023 13:42

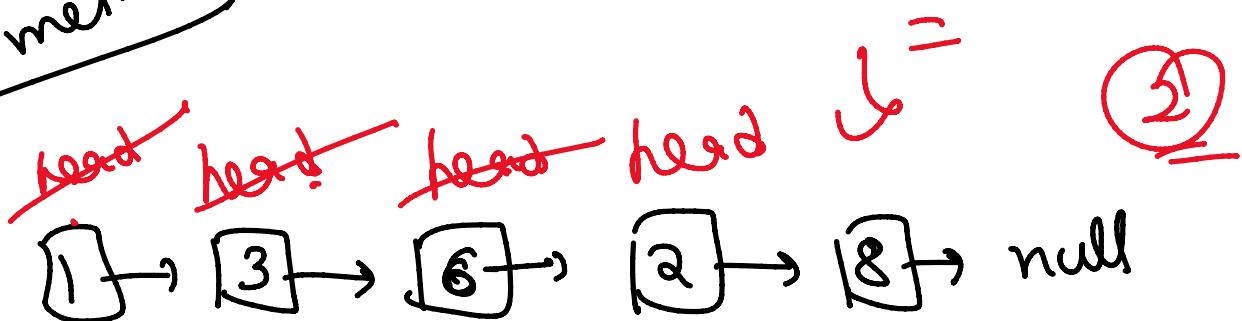


iterative head



while ($\text{temp} \neq \text{null}$)
{

recursive method



if ($\text{head} == \text{NULL}$)
return false;

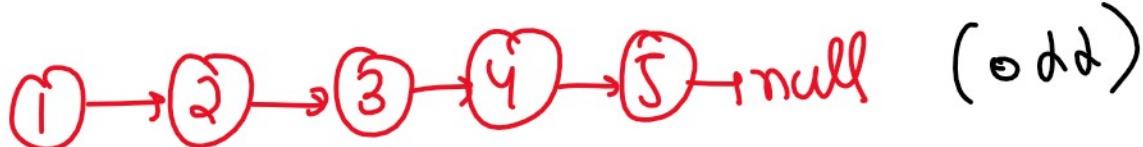
Mid point in linkedlist

02 September 2023 13:52



$$\text{ans} = 3$$

(even)
2 mid point's



$$\text{ans} = 3$$

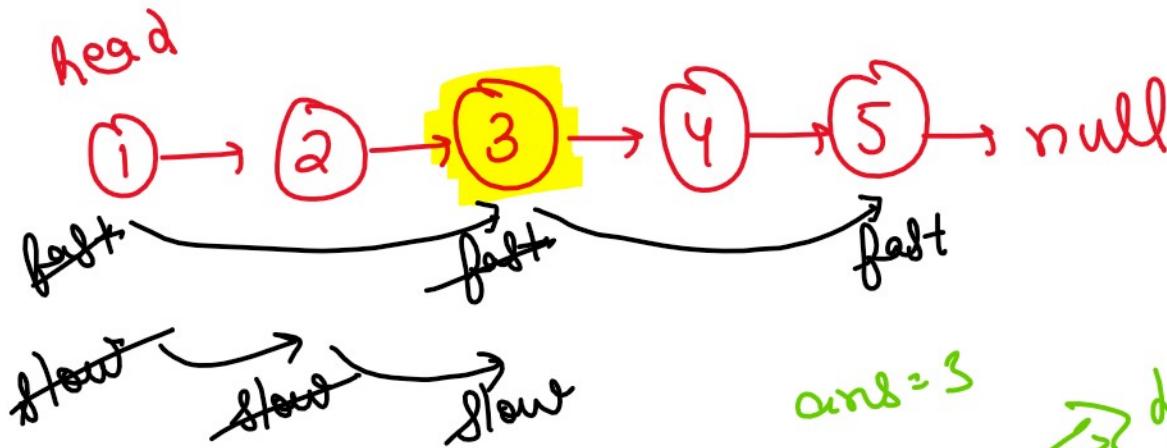
1 mid point

- ✓ find length of linkedlist + .
- ✓ move to $\text{length}/2$ nodes.

Time: $O(N)$
Space: $O(1)$

To solve this in single pass.

$N \rightarrow N/2$



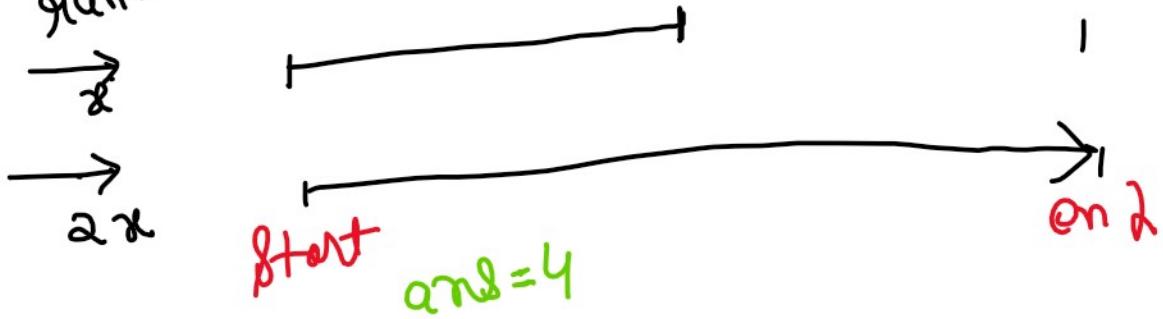
.. techniques.

mid

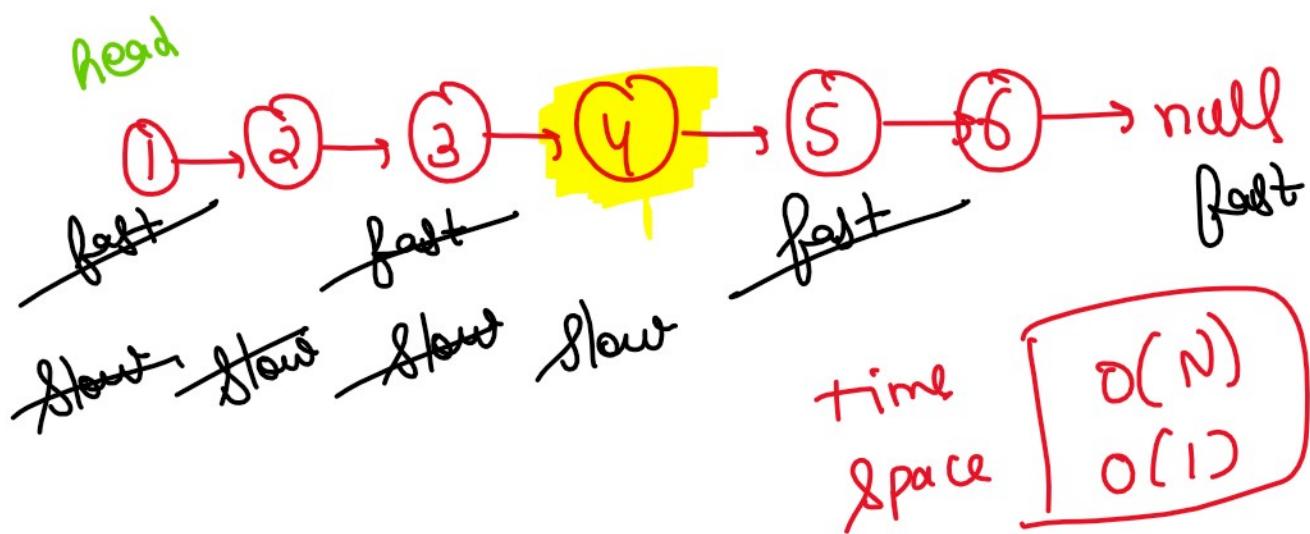
ans = 3
while ($\text{fast} \neq \text{NULL}$)
 $\text{fast} \rightarrow \text{next} \neq \text{NULL}$

duplicate

runner technique.

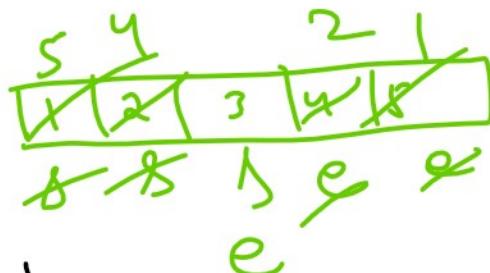
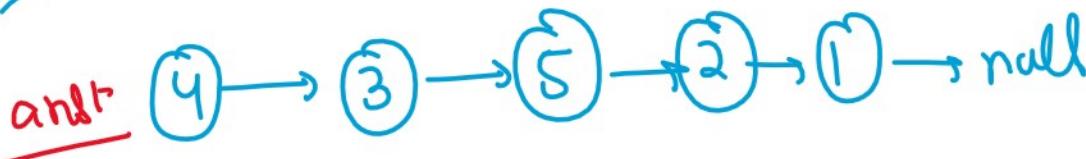
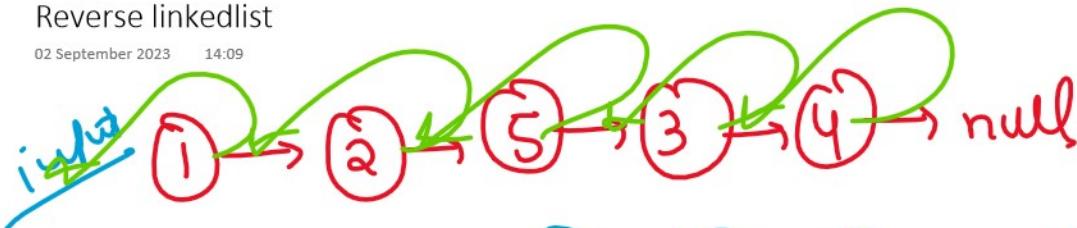


where \leftarrow fast \rightarrow next! - NULL



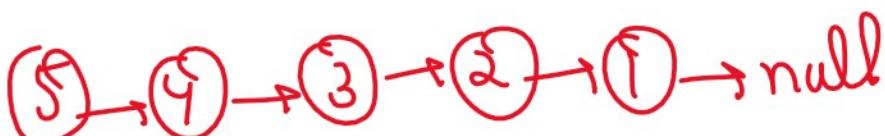
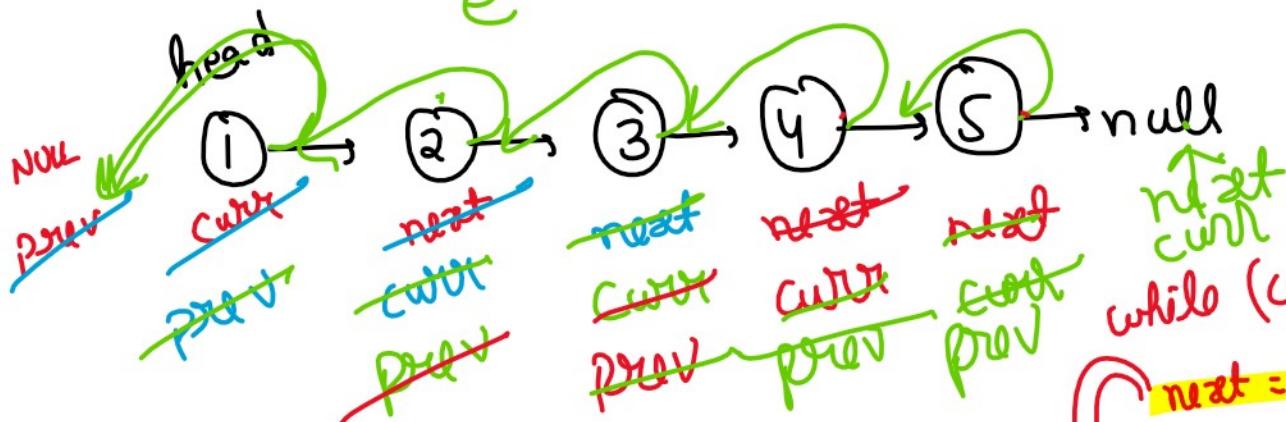
Reverse linkedlist

02 September 2023 14:09

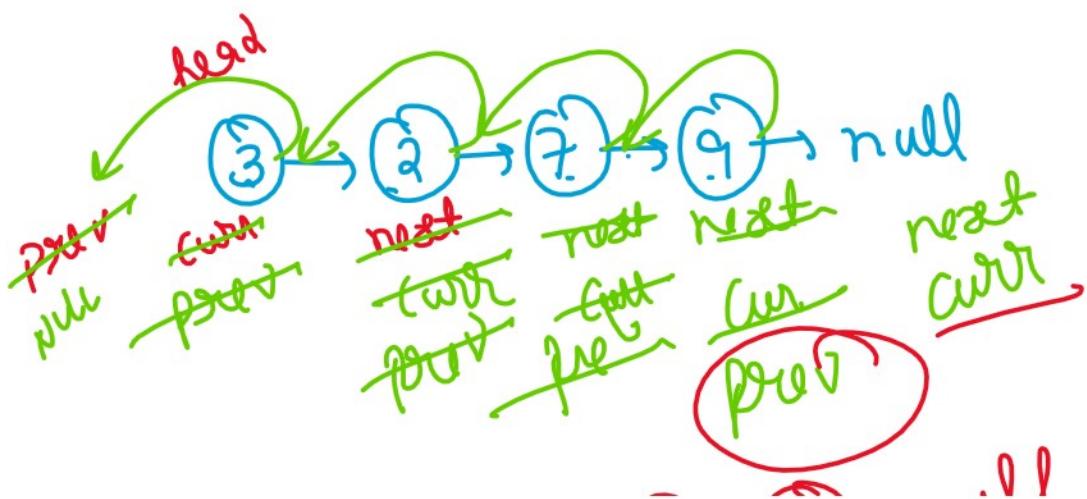


~~swapping~~

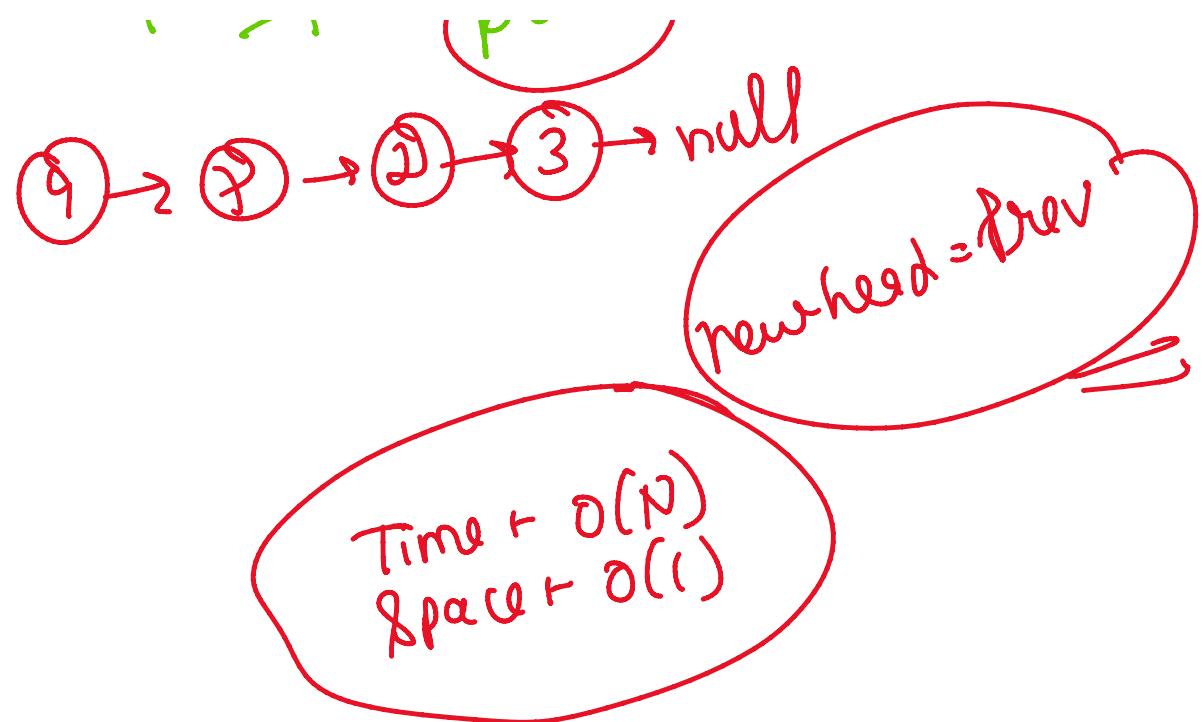
prev



while ($curr \neq null$)
 next = curr → next
 curr → next = prev
 prev = curr
 curr = next

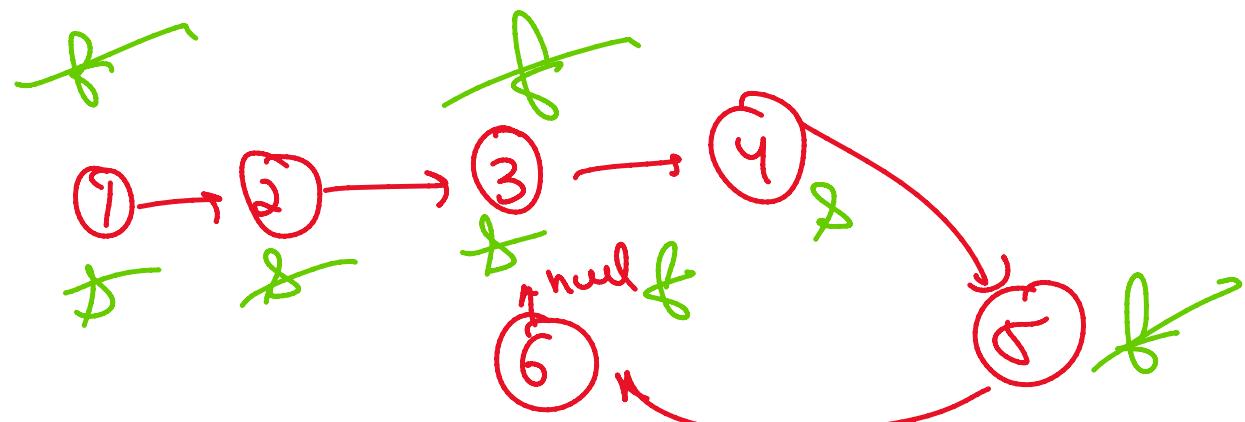
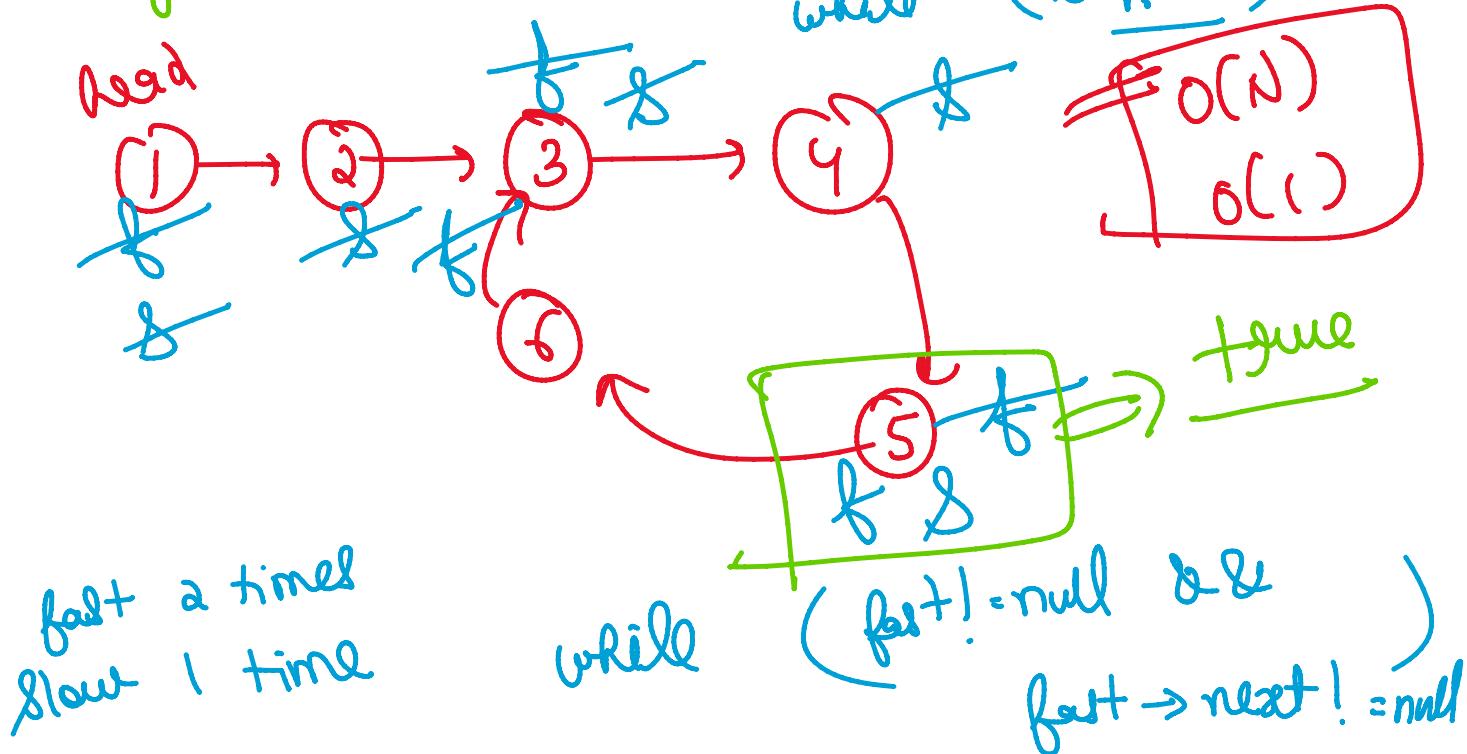
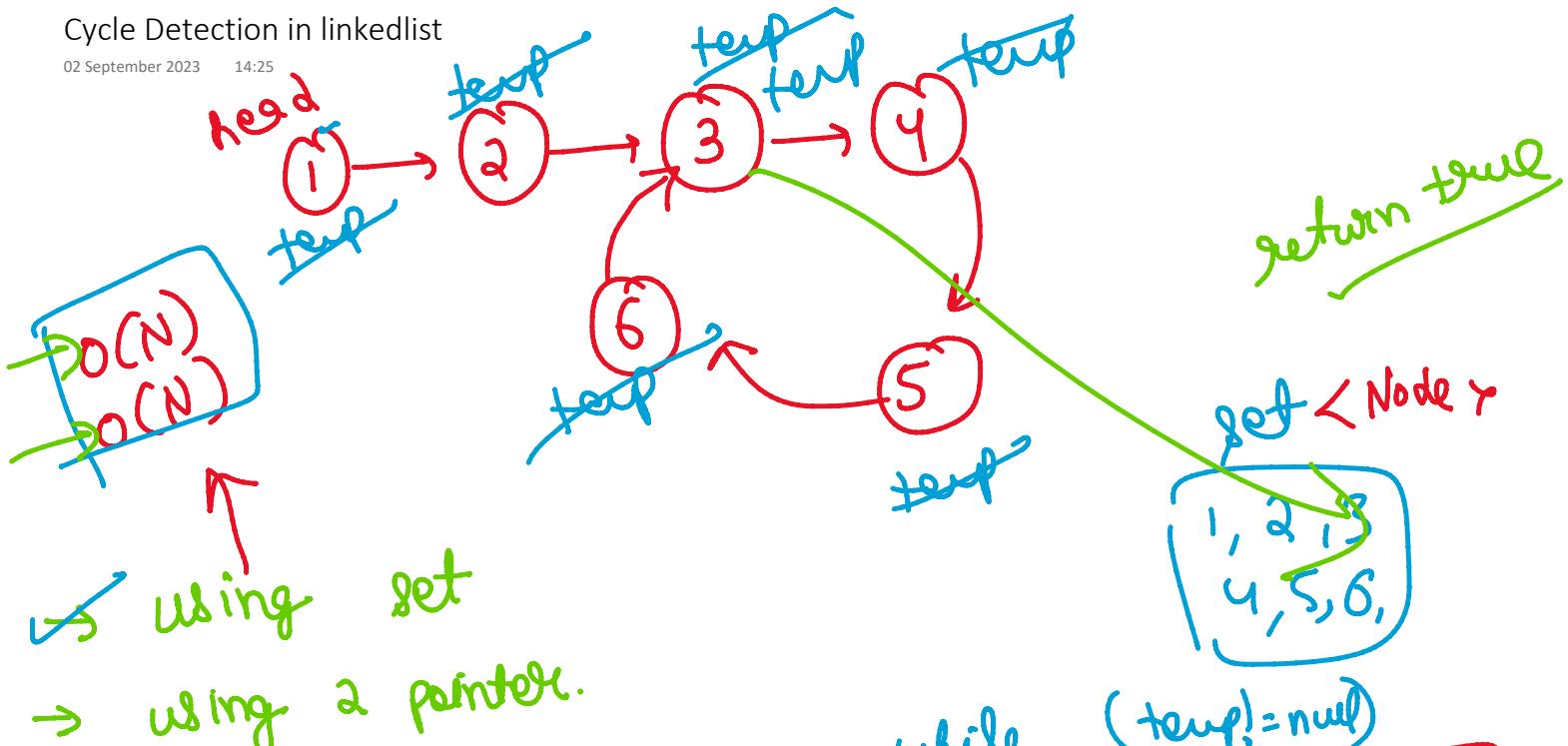


.01



Cycle Detection in linkedlist

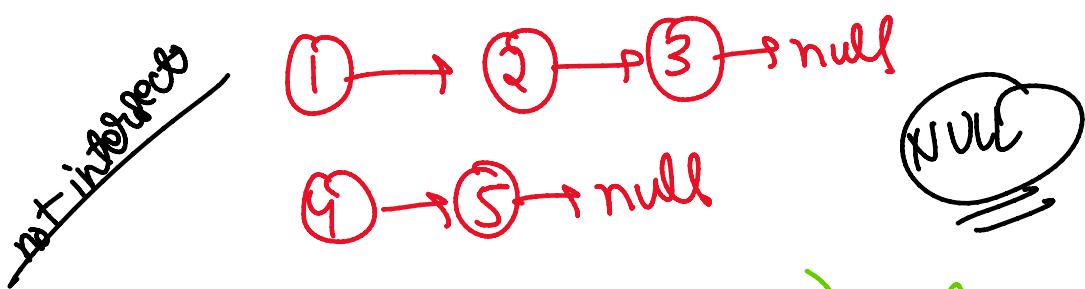
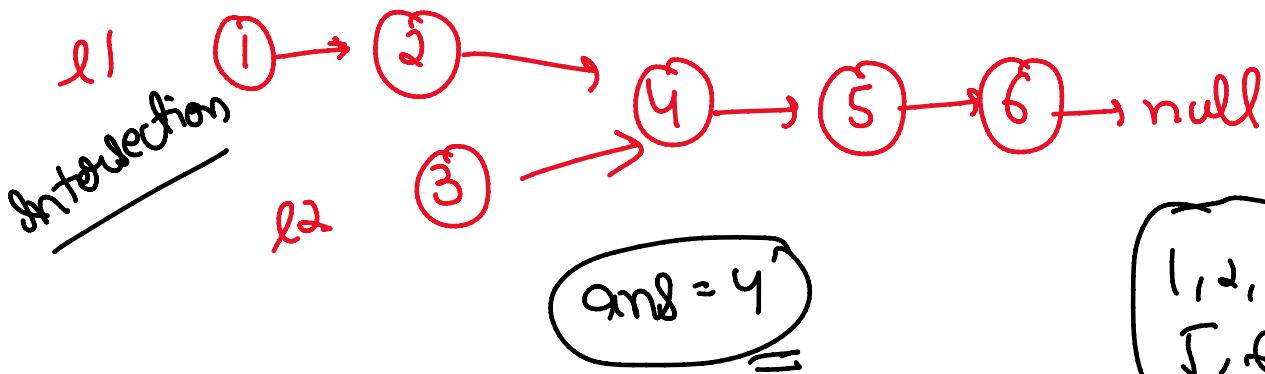
02 September 2023 14:25




floyd cycle detection

Intersection of two linkedlist

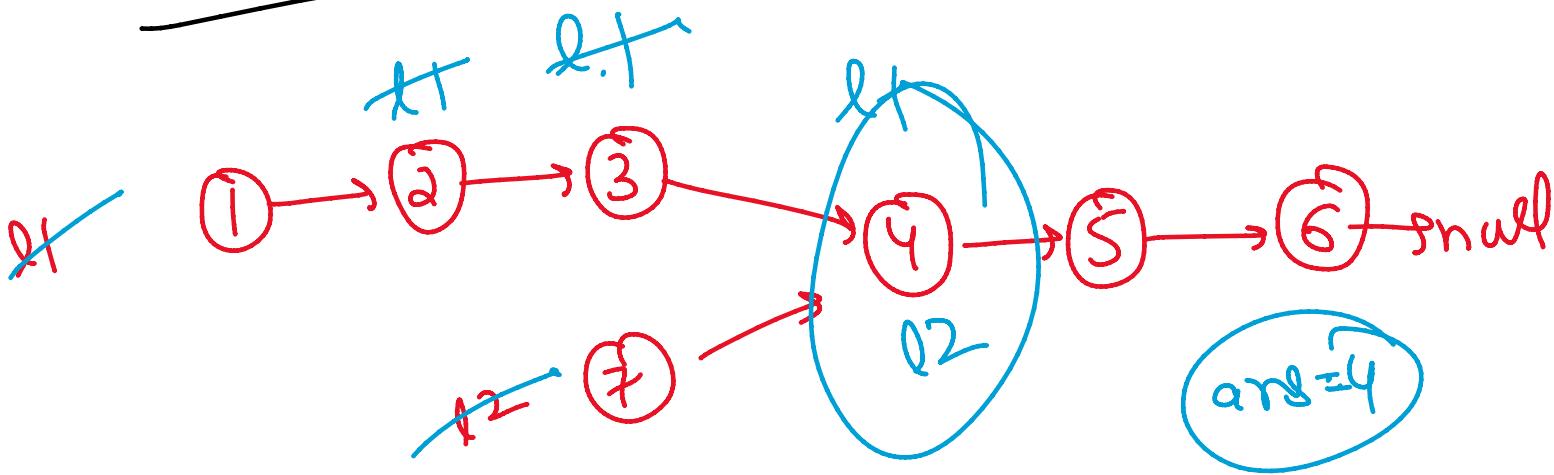
02 September 2023 15:03



→ using set. Time $\in O(N+m)$, Space $\in O(N+m)$

→ Two loops. Time $\in O(N \times m)$ Space $\in O(1)$

→ ? Time $\in O(N+m)$, Space $\in O(1)$



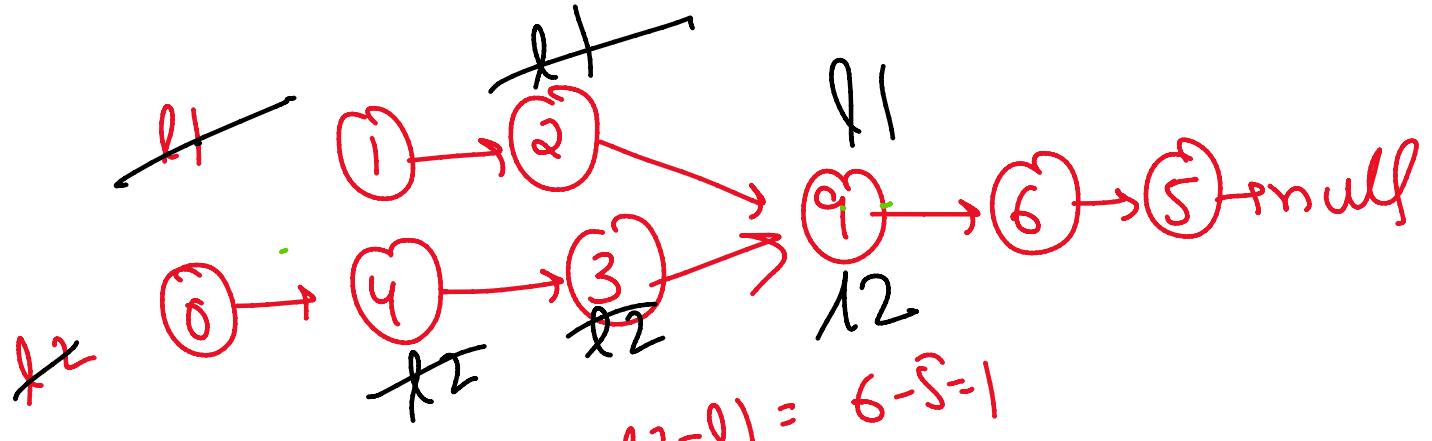
Length of $l1 = 6$

diff: $l1 - l2$

length of $l_1 = 6$
length of $l_2 = 4$

$$\text{diff} = \underline{x}^{l_1 - l_2} = 2$$

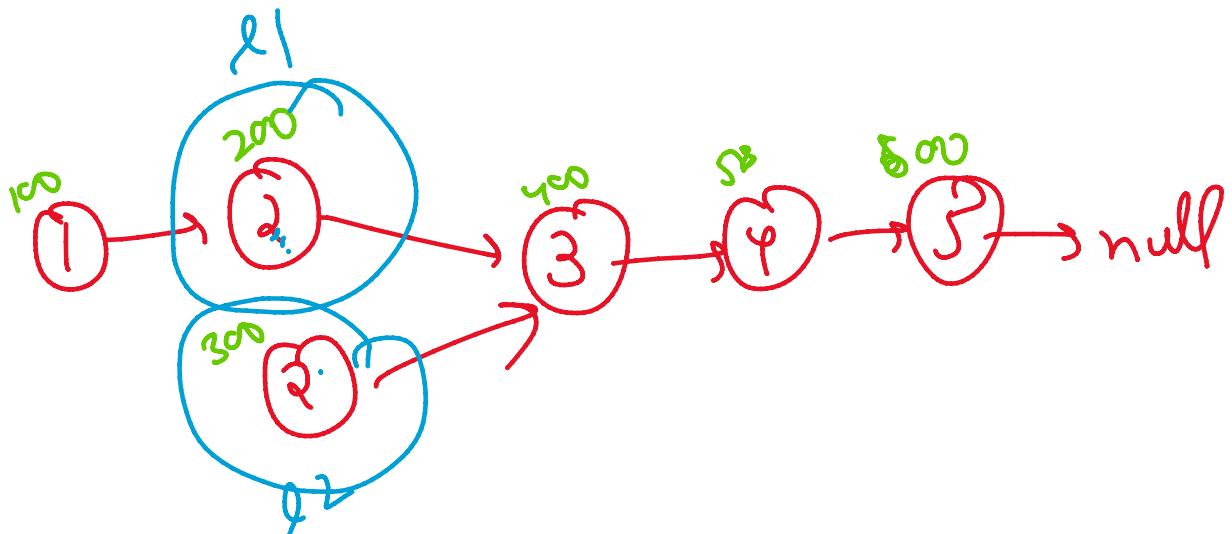
Time + $O(n+m)$, Space + $O(1)$



$$l_1 = 6 \\ l_2 = 5$$

$$\text{diff} = (l_2 - l_1) = 6 - 5 = 1$$

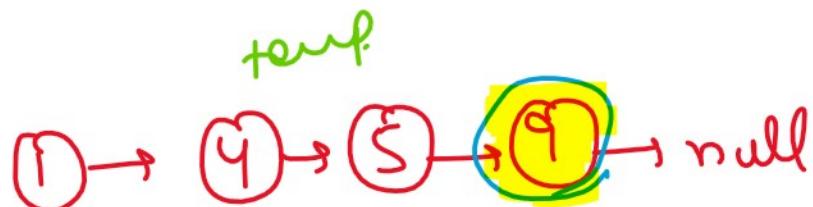
9 and



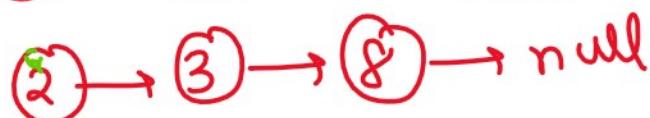
Merge 2 sorted lists

02 September 2023 15:27

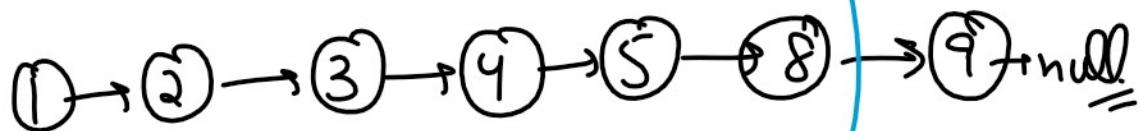
l_1



l_2



ans



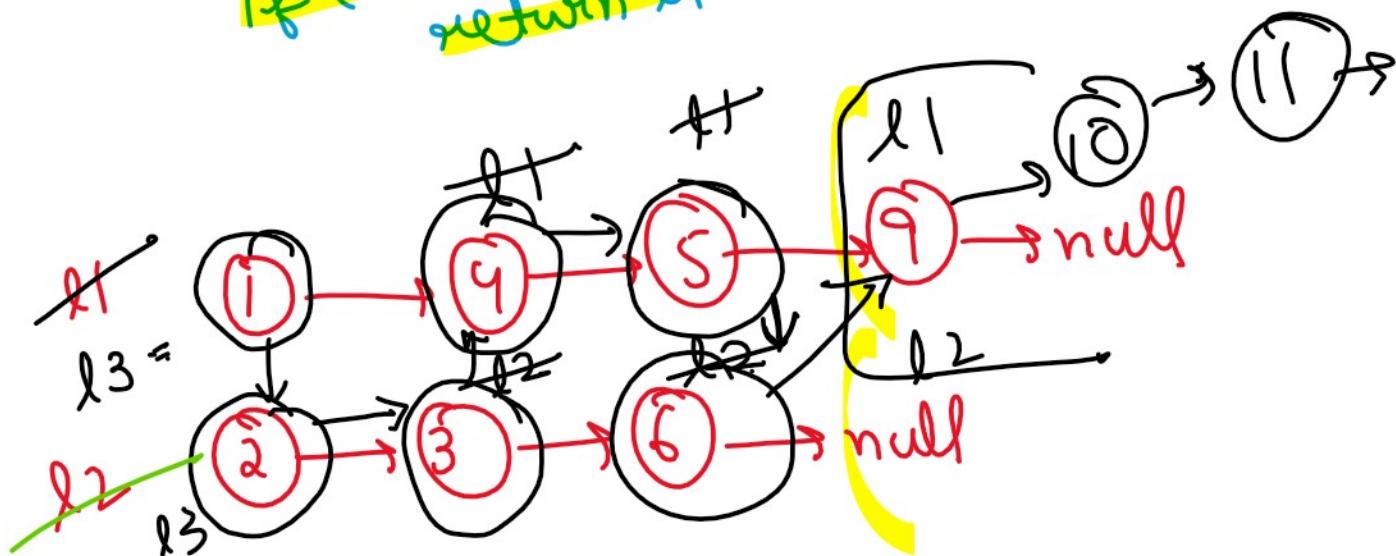
Base Case

```

if ( $l_1 == \text{null}$ )
    return  $l_2$ 
if ( $l_2 == \text{null}$ )
    return  $l_1$ 

```

H.W. iterative
recursive



$O(m+N)$
 $O(m+N)$

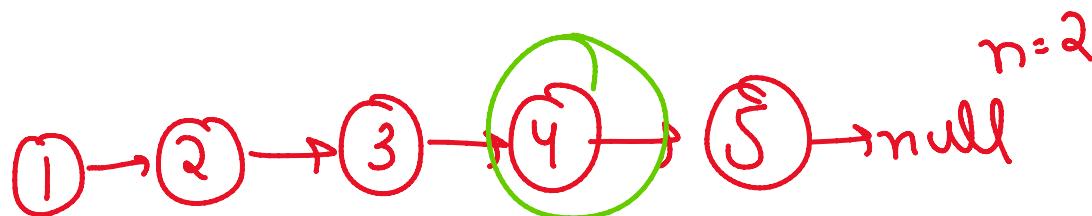
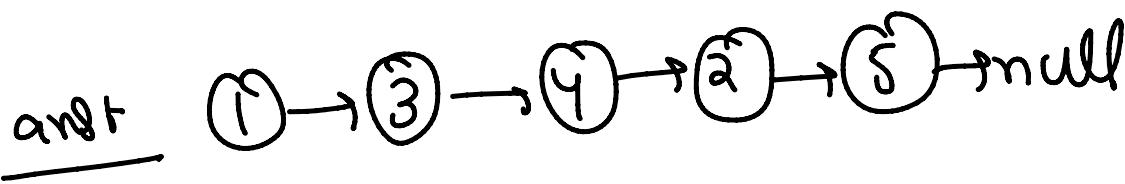
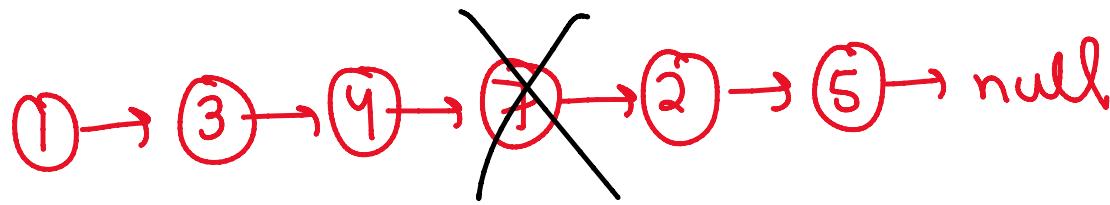
merge(l1, l2)

$\checkmark l_1 \rightarrow \text{next}, l_2$
 $\checkmark l_1, l_2 \rightarrow \text{next}$

Remove Nth node from the last

02 September 2023 15:48

$n=3$



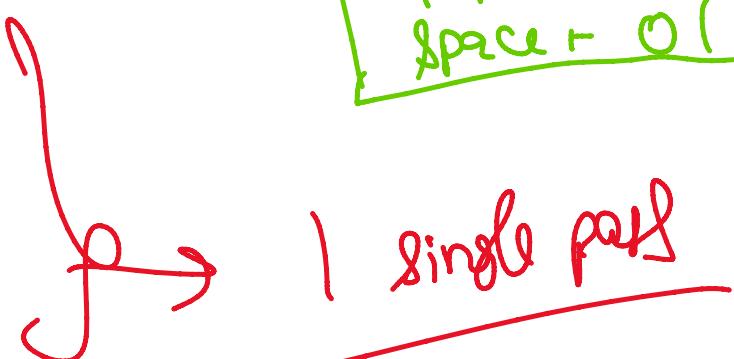
find length = 5

$$S-2 = 3 + 1$$

Time = $O(N)$
Space = $O(1)$

2 pass

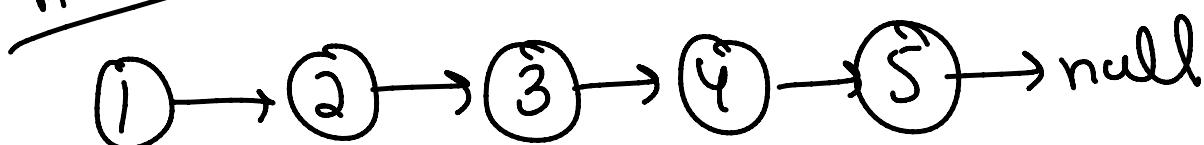
1 → length
2 → removal



1 single pass

$n=2$

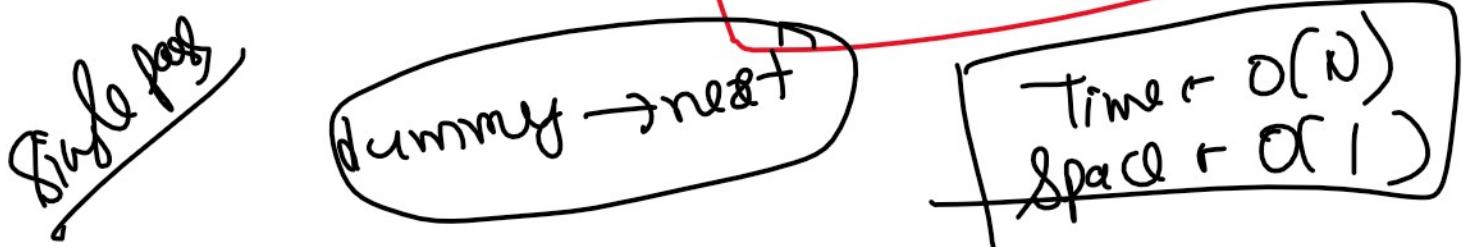
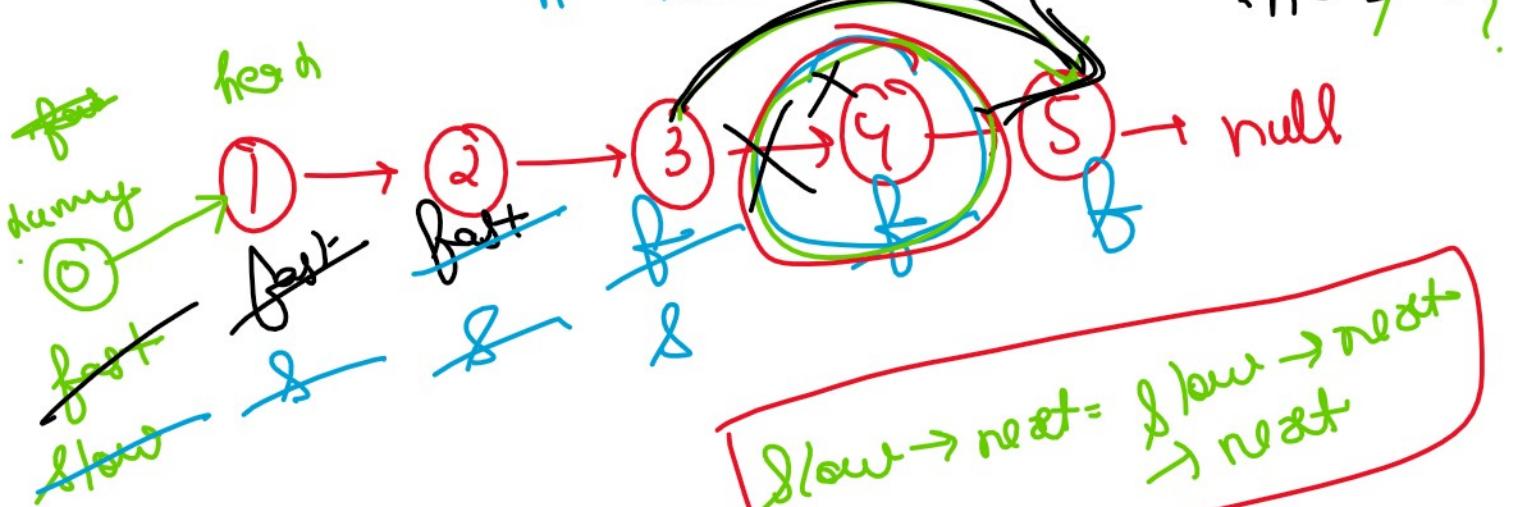
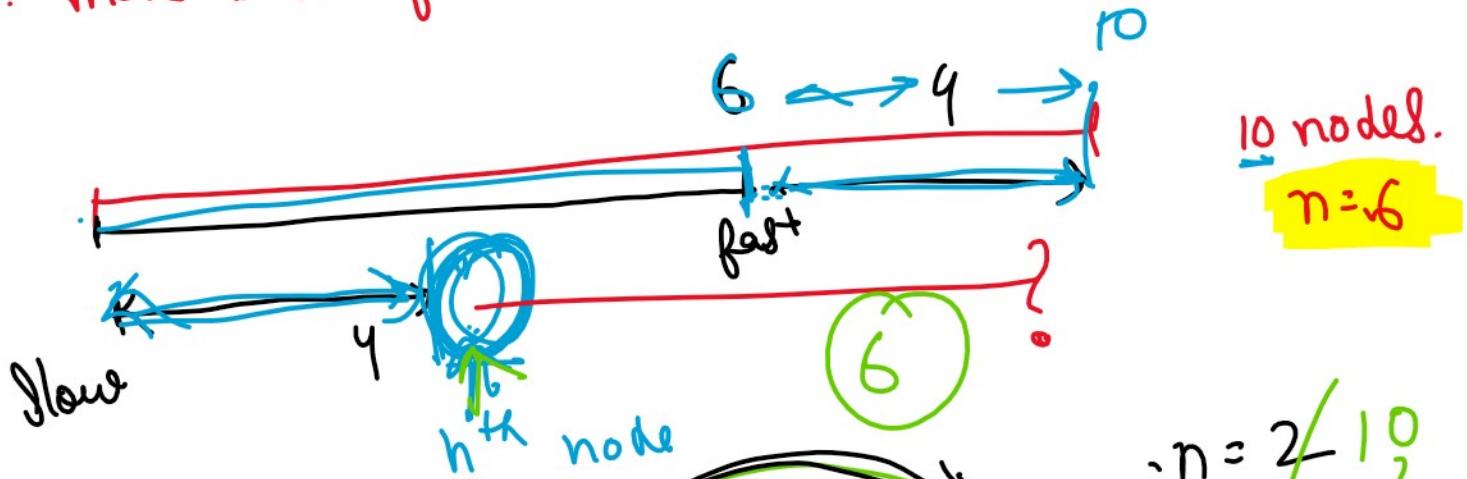
Read



1 move fast 'n' times

... back

1. move fast 'n' times
2. move both fast and slow one step each.



Time $\in O(N)$
Space $\in O(1)$