

## LinkedList

01 October 2022 13:01

Arrays  
Hashmap

LinkedList

Stack

Queue

Tree

Heap

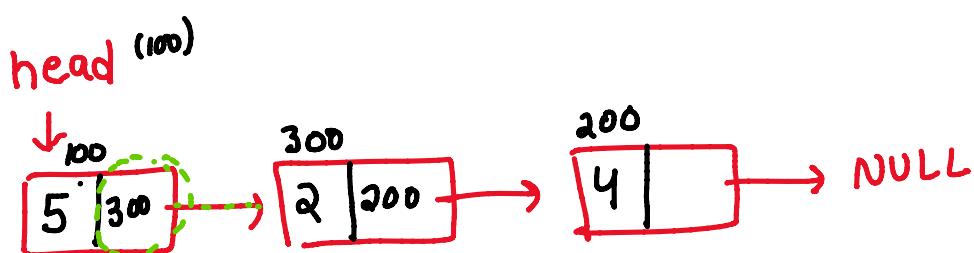
Graphs

Node



address of next node.

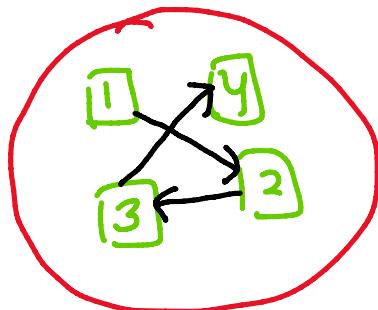
→ collection of nodes.



insert  
delete  
search

## Array

|   |   |   |   |   |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| 1 | 5 | 3 | 4 | 2 |



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

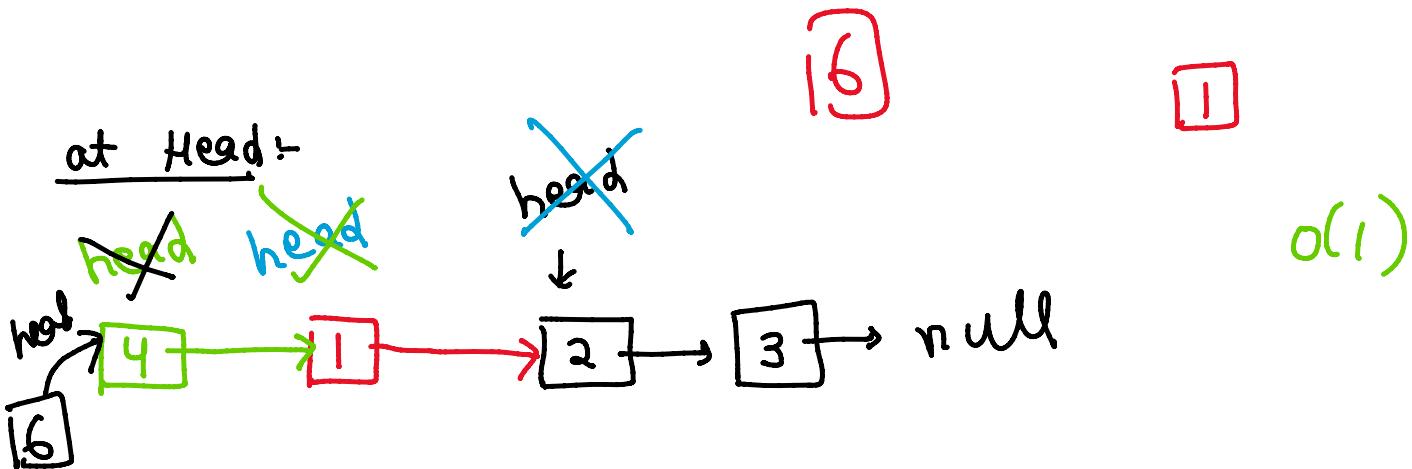
C++  
→  
NULL

Java  
.null

## Insertion in LinkedList

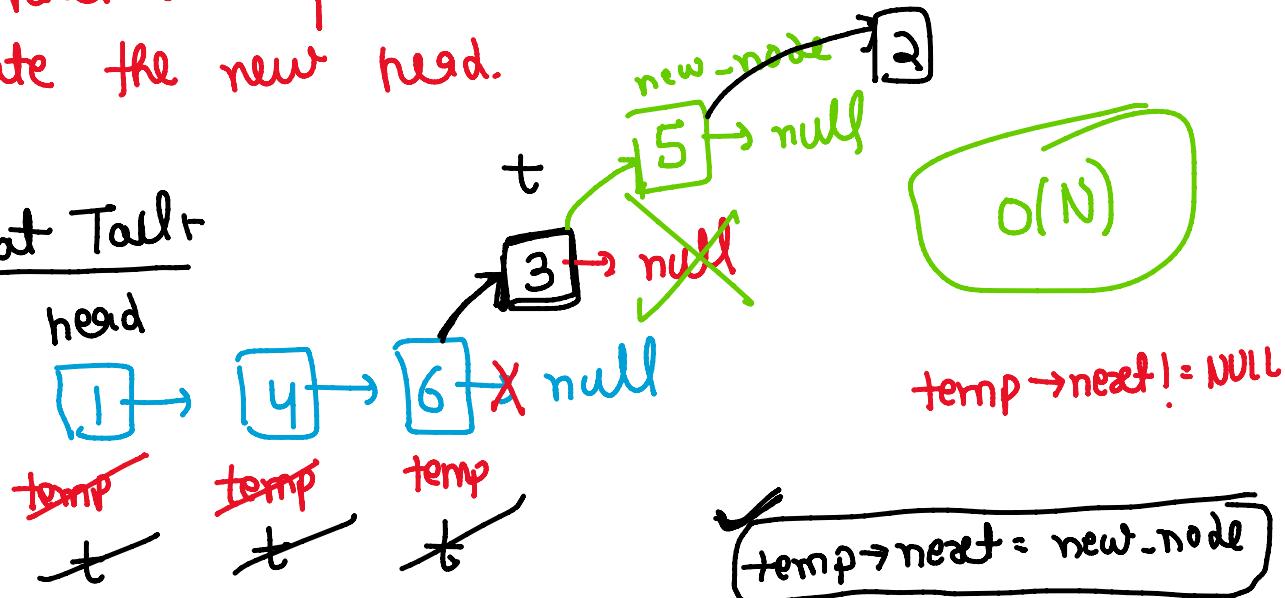
01 October 2022 13:18

- at Head
- at Tail
- at any position

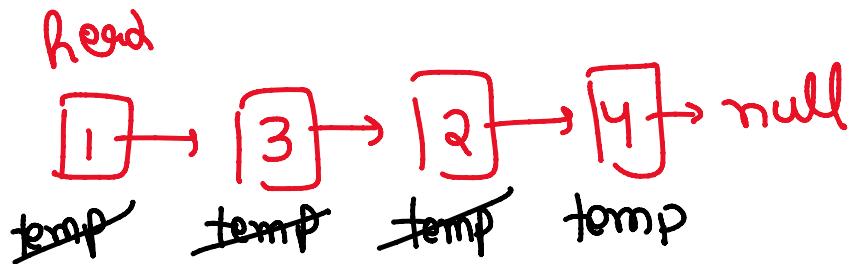


- ✓ 1. Create a node
- ✓ 2. nod1.next will point to head
- ✓ 3. Update the new head.

## Insert at Tail

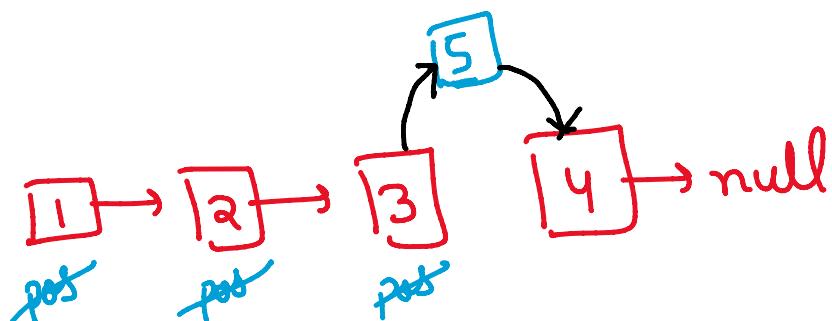


## Print linkedlist



1 → 3 → 2 → 4

at any given position



pos = 4  
val = 5

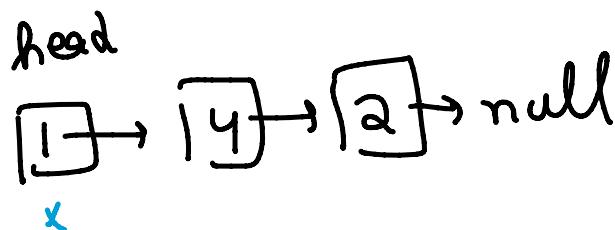
## Deletion in linkedlist

01 October 2022 13:47

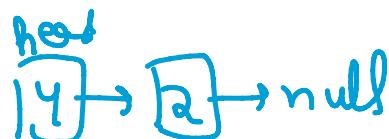
- at head
- at tail
- at any given position.

head  
↓  
NULL

at head



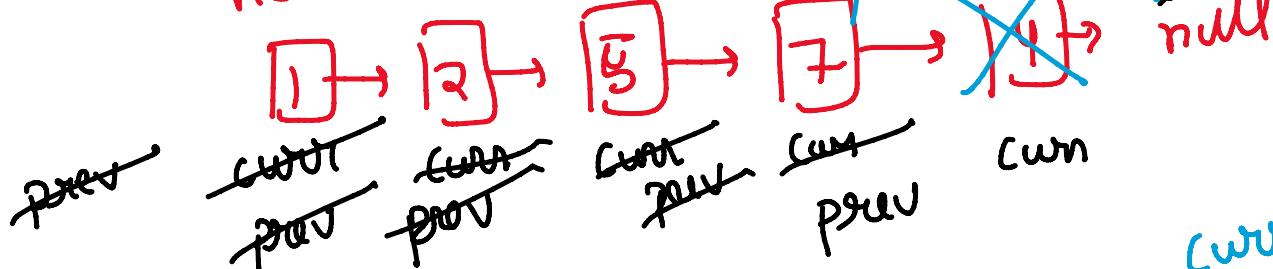
O(1)



$(curr \rightarrow next) = NULL$

at tail

head



O(N)

$(curr = head,$   
 $prev = NULL)$



O(N) }  
O(N) }

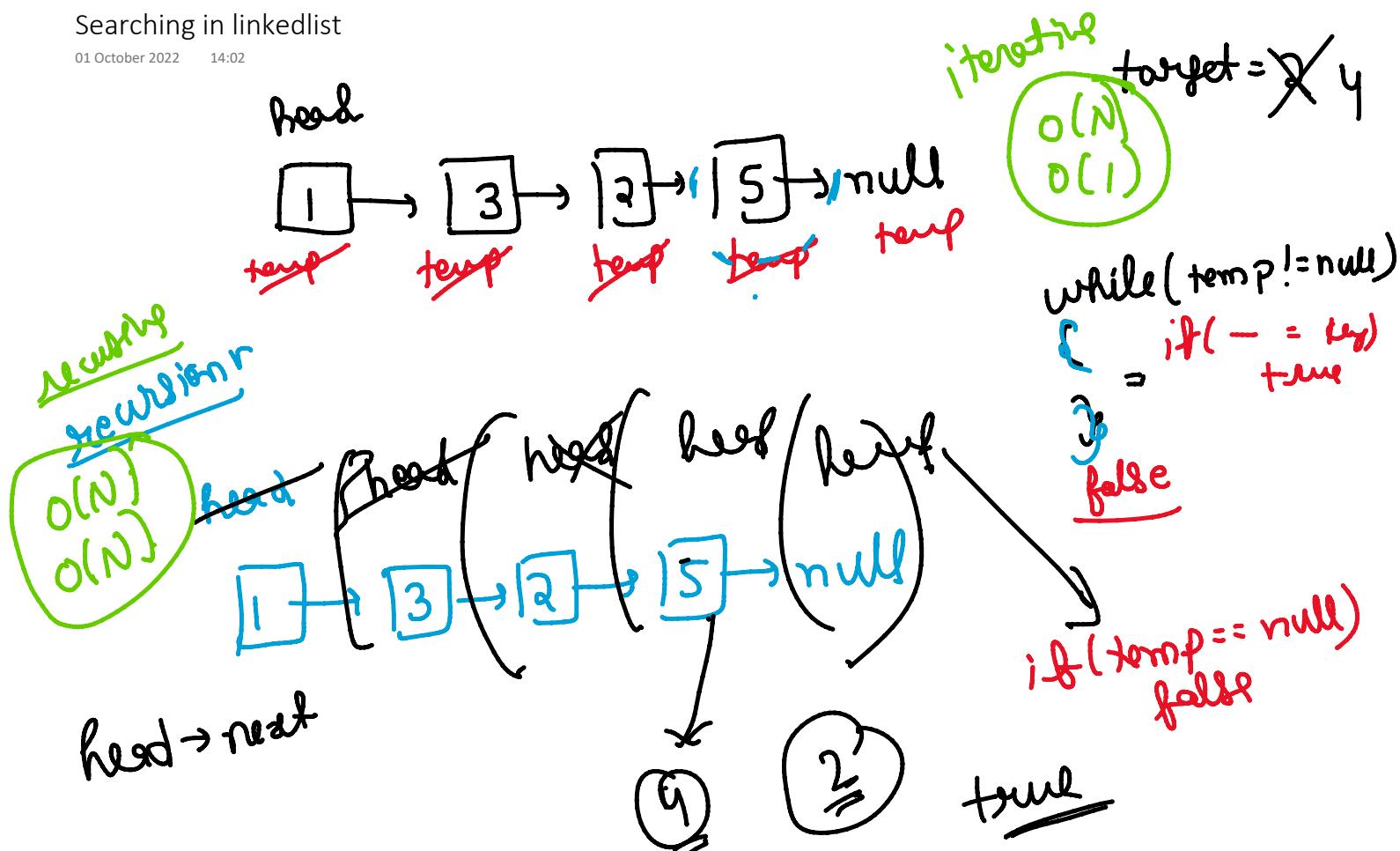
• next.next

✓ head  
[1] → null

cur. next. next

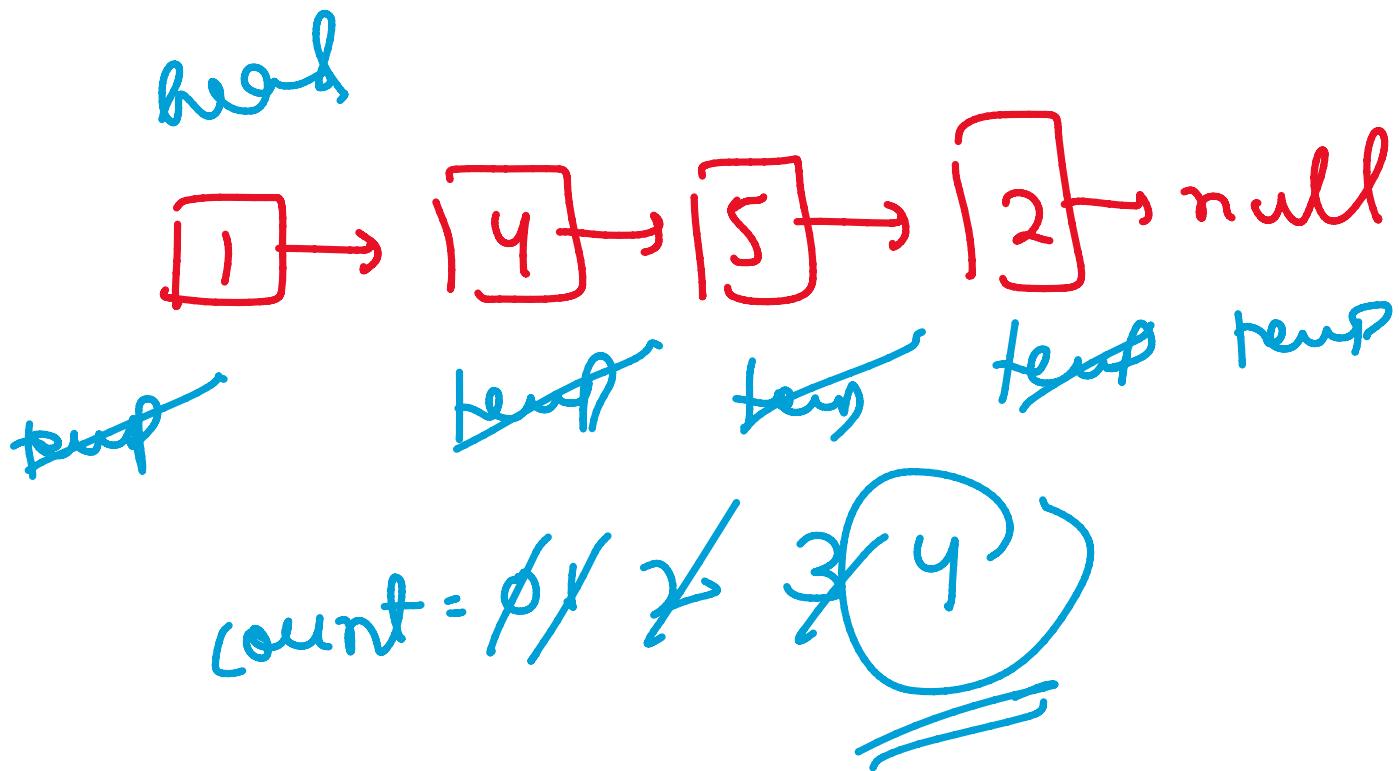
## Searching in linkedlist

01 October 2022 14:02



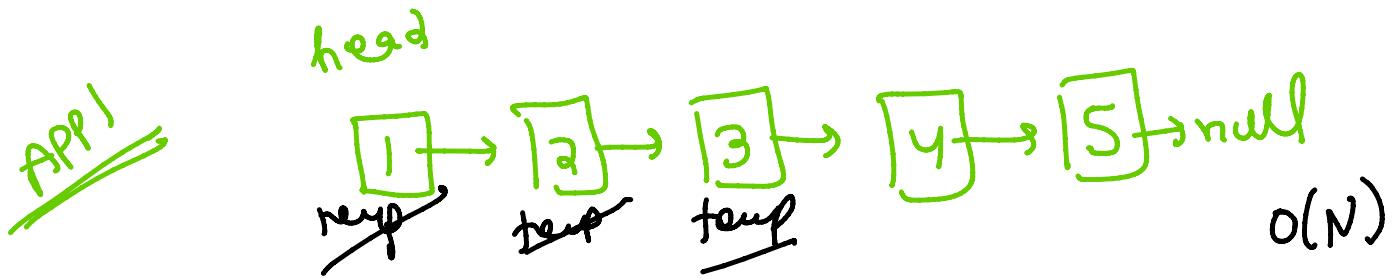
# Count nodes

01 October 2022 14:25



## Middle of linkedlist

01 October 2022 14:32



$$\text{count} = 5$$

$$= 2 + 1 = \underline{\underline{3}}$$

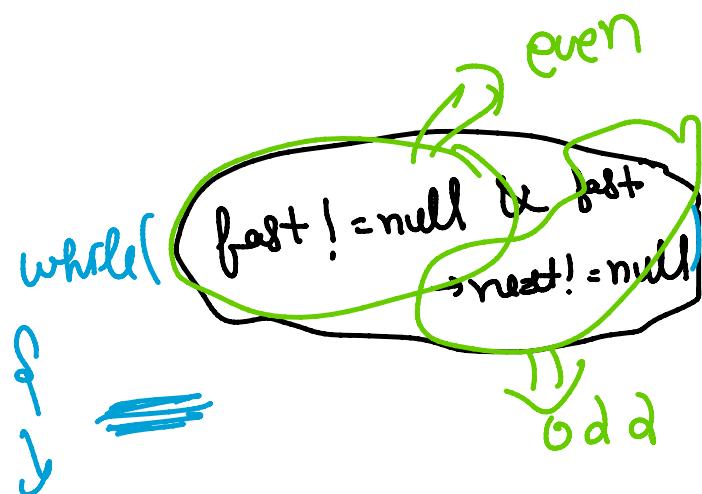
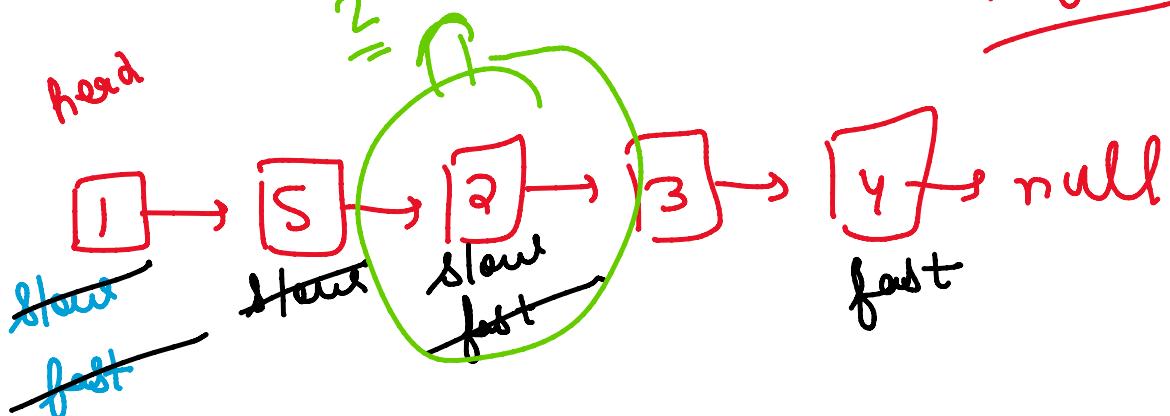
2 passes

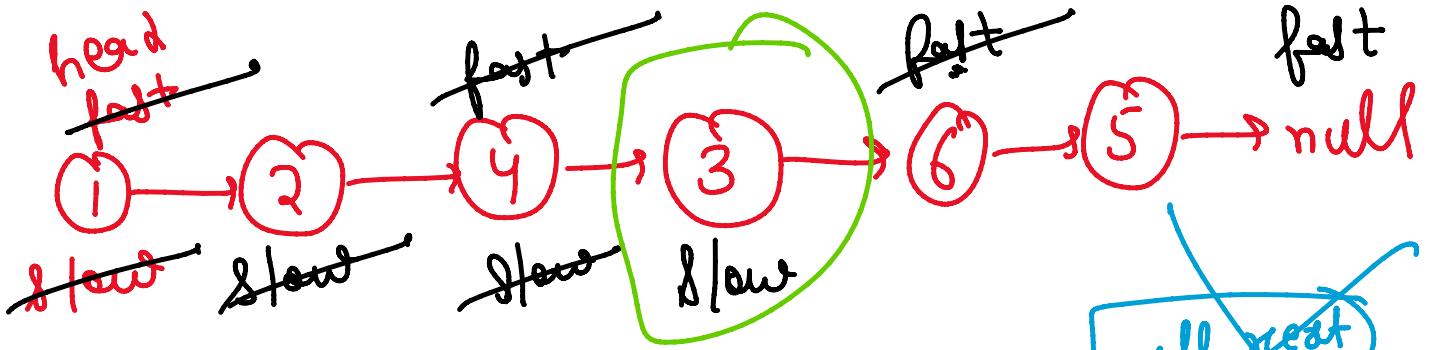
$$\text{count} = 6$$

$$= 3 + 1 = \underline{\underline{4}}$$

$\downarrow$   
[3, 4]

App 2





~~fast != null~~

3

~~null.next~~

fast = null

fast == next == null

runner technique

2, 2 (fast)

Horse 1

2 (slow)

Horse 2

mid

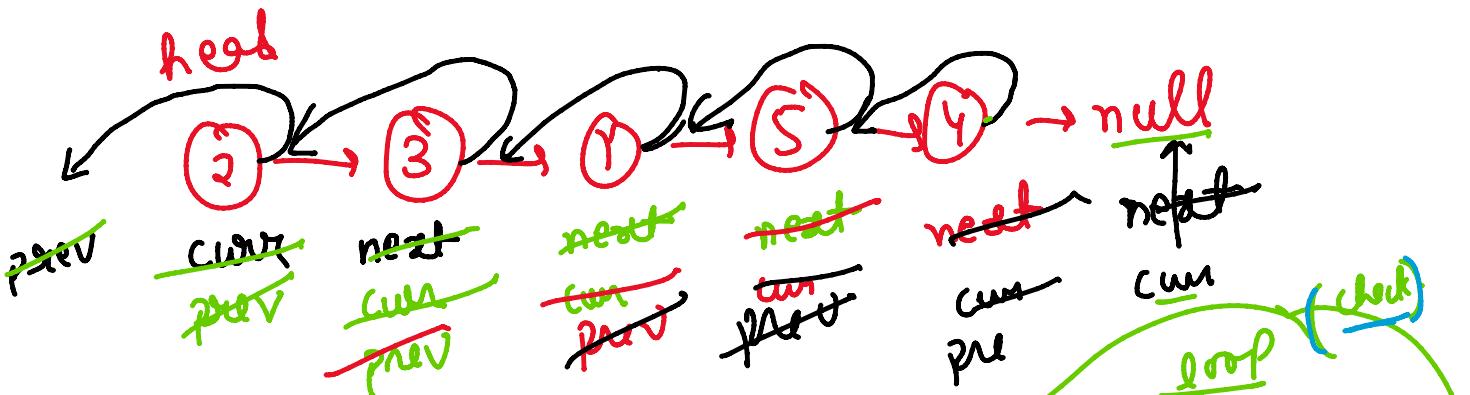
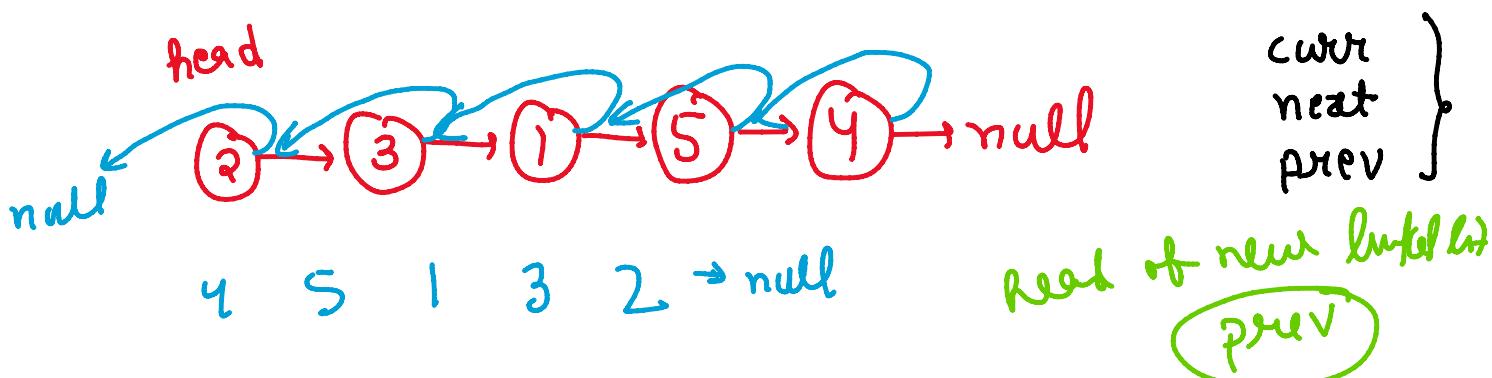
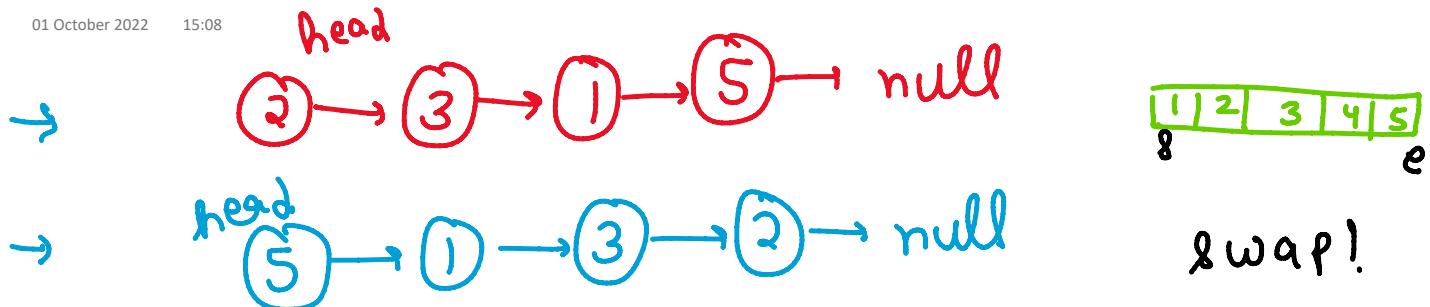
end point

single path

$O(N)$   
 $O(1)$

## Reverse a linkedlist

01 October 2022 15:08

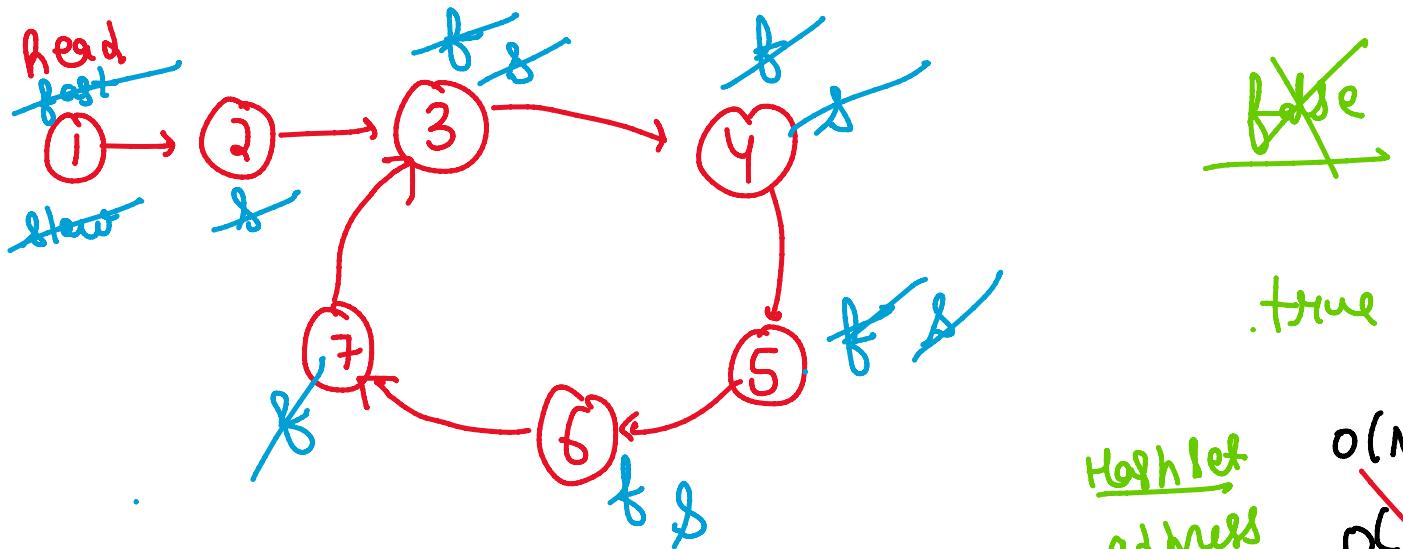


$O(N)$   
 $O(1)$

\*  $N = \text{curr} \rightarrow \text{next}$   
!  $\text{curr} \rightarrow \text{next} = \text{prev}$   
 $\text{prev} = (\text{curr})$   
 $\text{curr} = N$

## Detect Cycle

01 October 2022 15:31

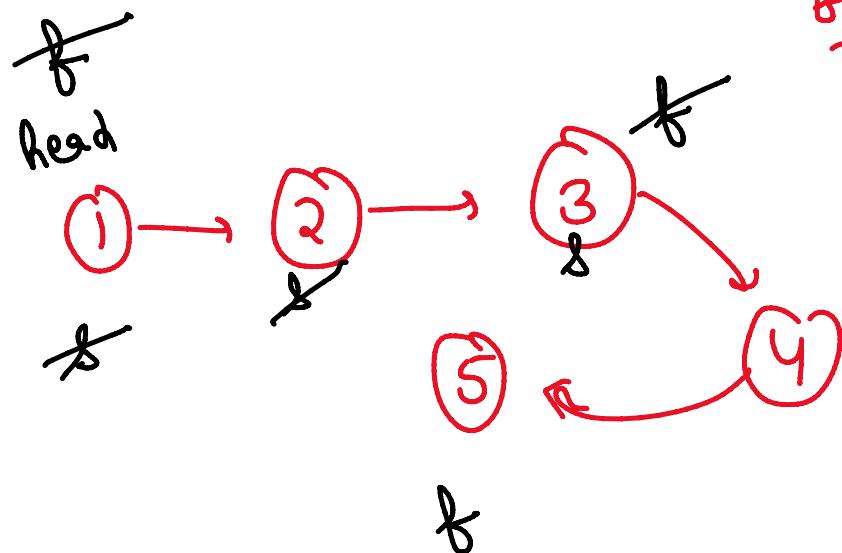


fast.next == null || fast = null

Floyd cycle detection

Hash set address

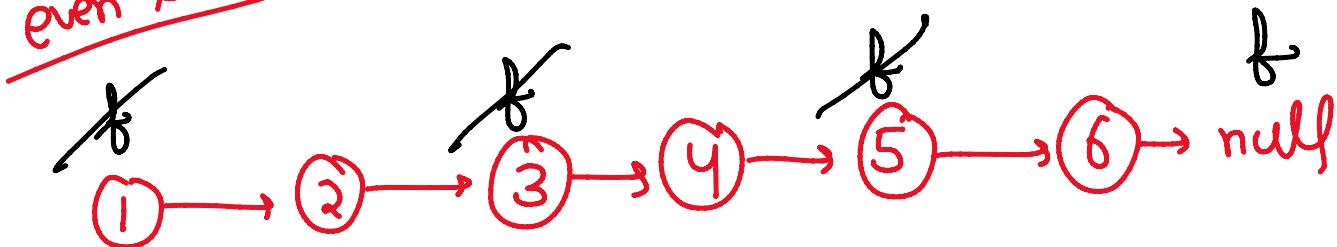
~~O(N)~~  
~~O(N)~~  
~~O(1)~~



false

X

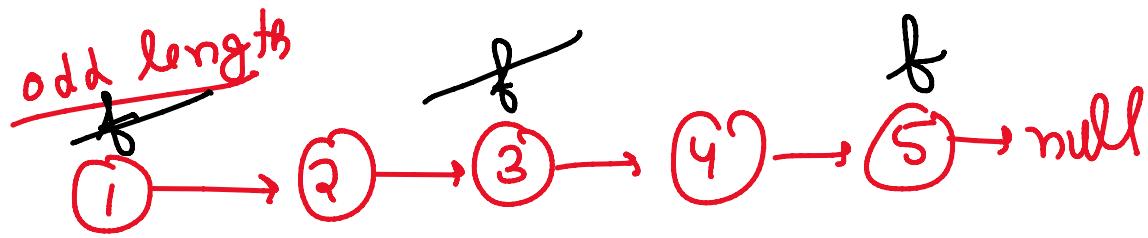
even length



odd length

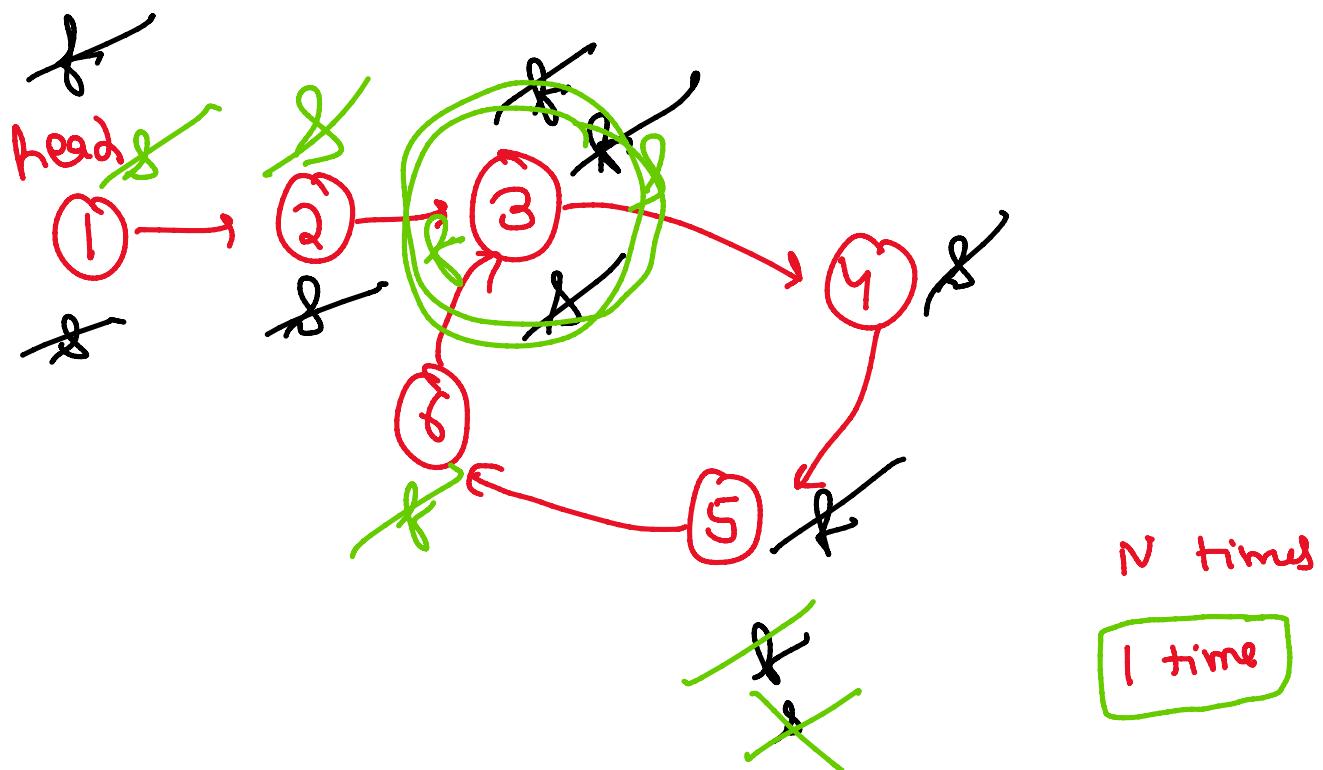
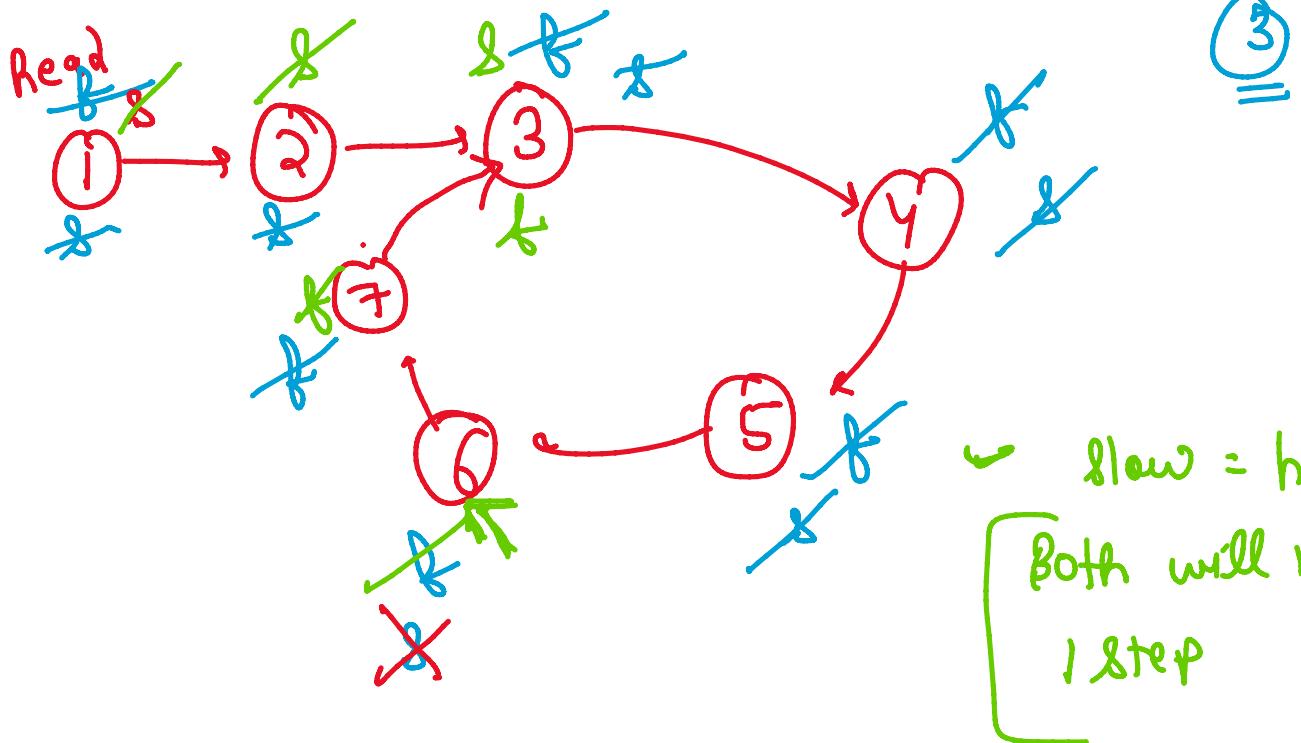
s

f



## Detect Cycle ii

01 October 2022 15:45



→ Intersection of two linked list ↴