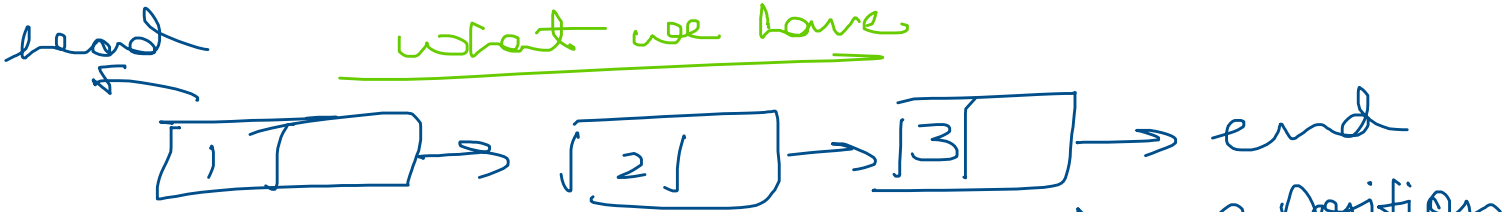
Linked Lists, July - 20

20 July 2020 18:47

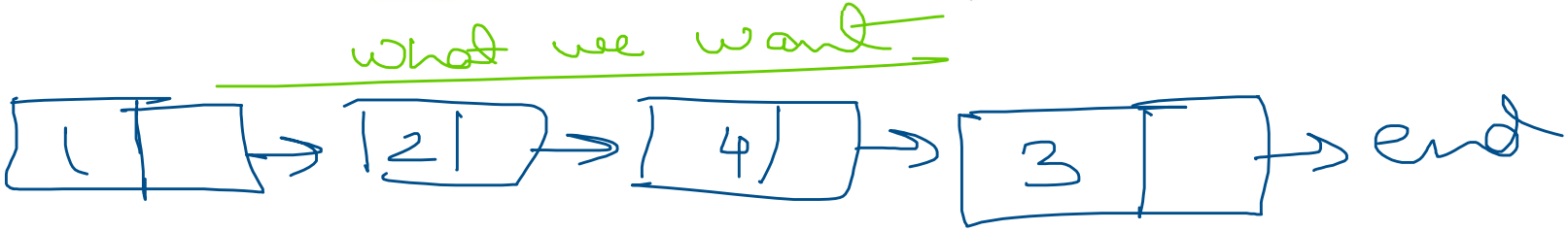
Insertion at nth position ✓
Delete a LL
Delete a node from beginning of a LL
Delete a node from the end of a LL
Delete a node from the nth position of a LL
Find the middle element of a LL
Reverse a Linked List

H.W

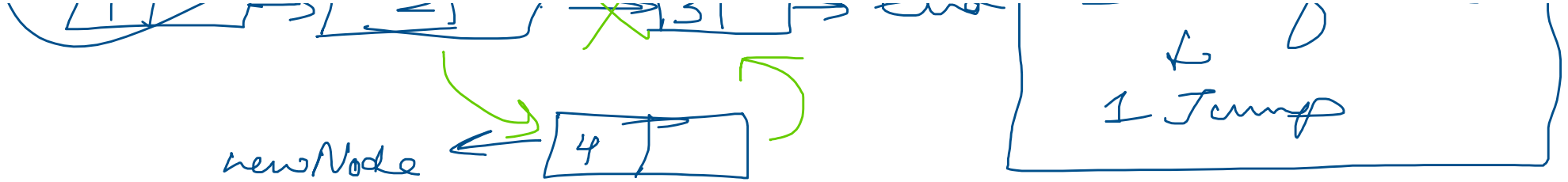
3 Insertion at nth position



insertAtPos (4 , 3)
value position



Steps



newNode.next = temp.next;

temp.next = newNode;

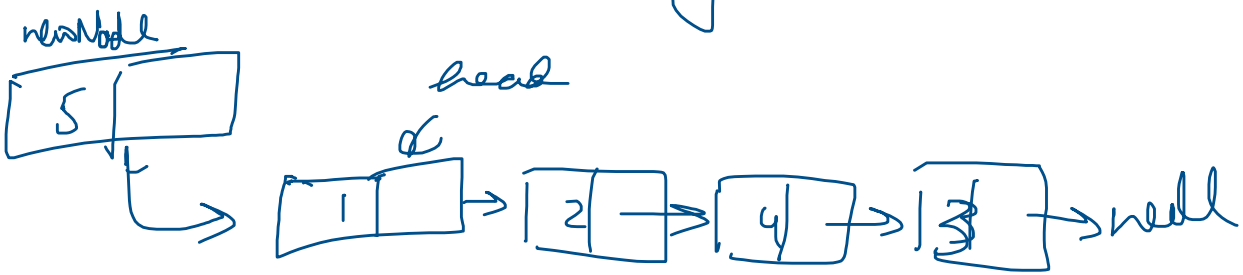
n-2 Jumps

Steps:

1. Add newNode
2. Create temp = head;
3. Take n-2 Jumps
4. newNode.next = temp.next;
5. temp.next = newNode;

Insert At Position 1 case:

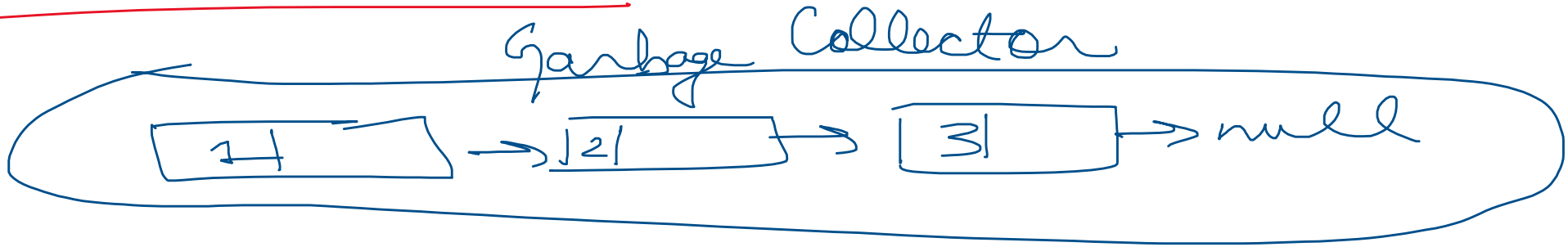
Handle separately



IAP(5, 1)

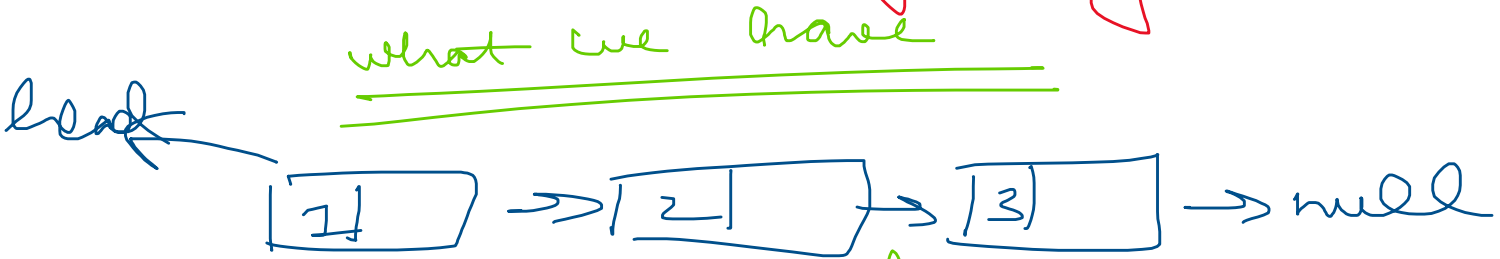
1. newNode.next = head;

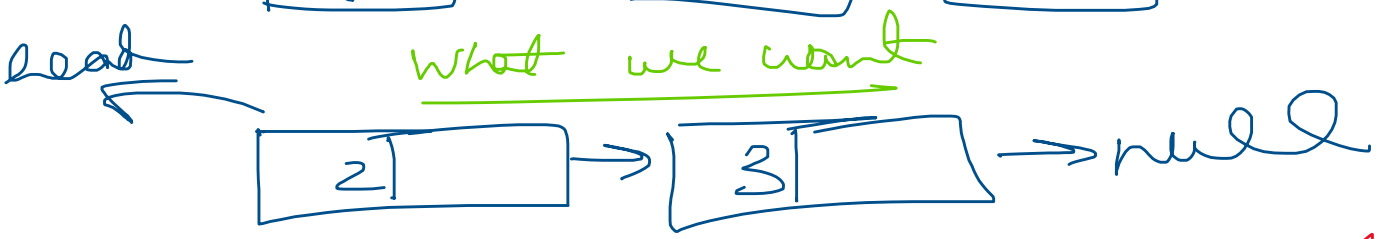
Delete a LL



head = null

Delete Node at Beginning of LL

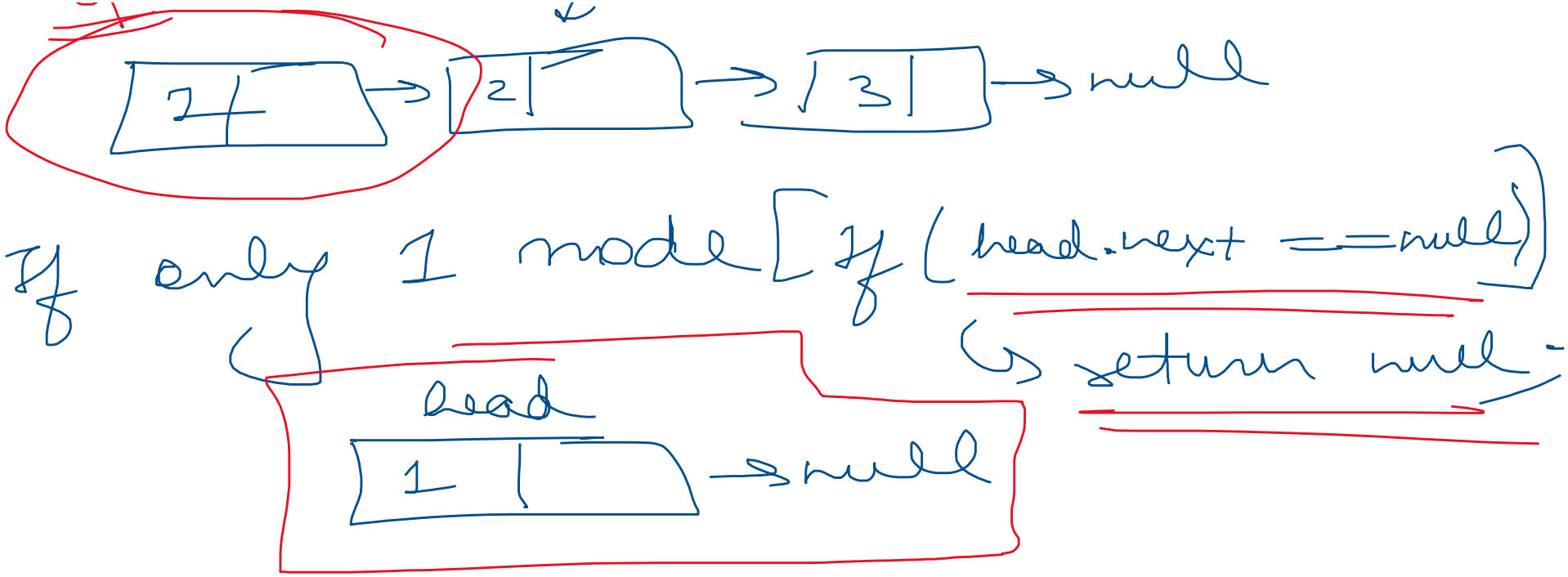




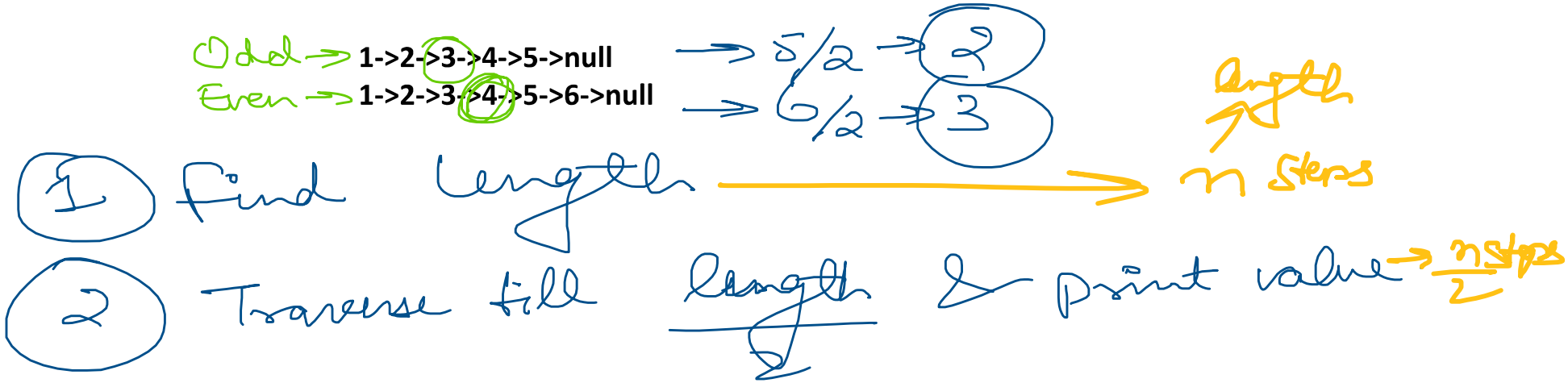
Steps

head = head . next

GC

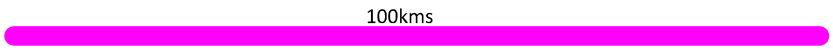


Print the Middle Element of a LL (Second in Case of even elements)



Total Steps → $\frac{3n}{2}$

Print the Middle Element of a LL (Second in Case of even elements) - Method 2



fptr

sptr

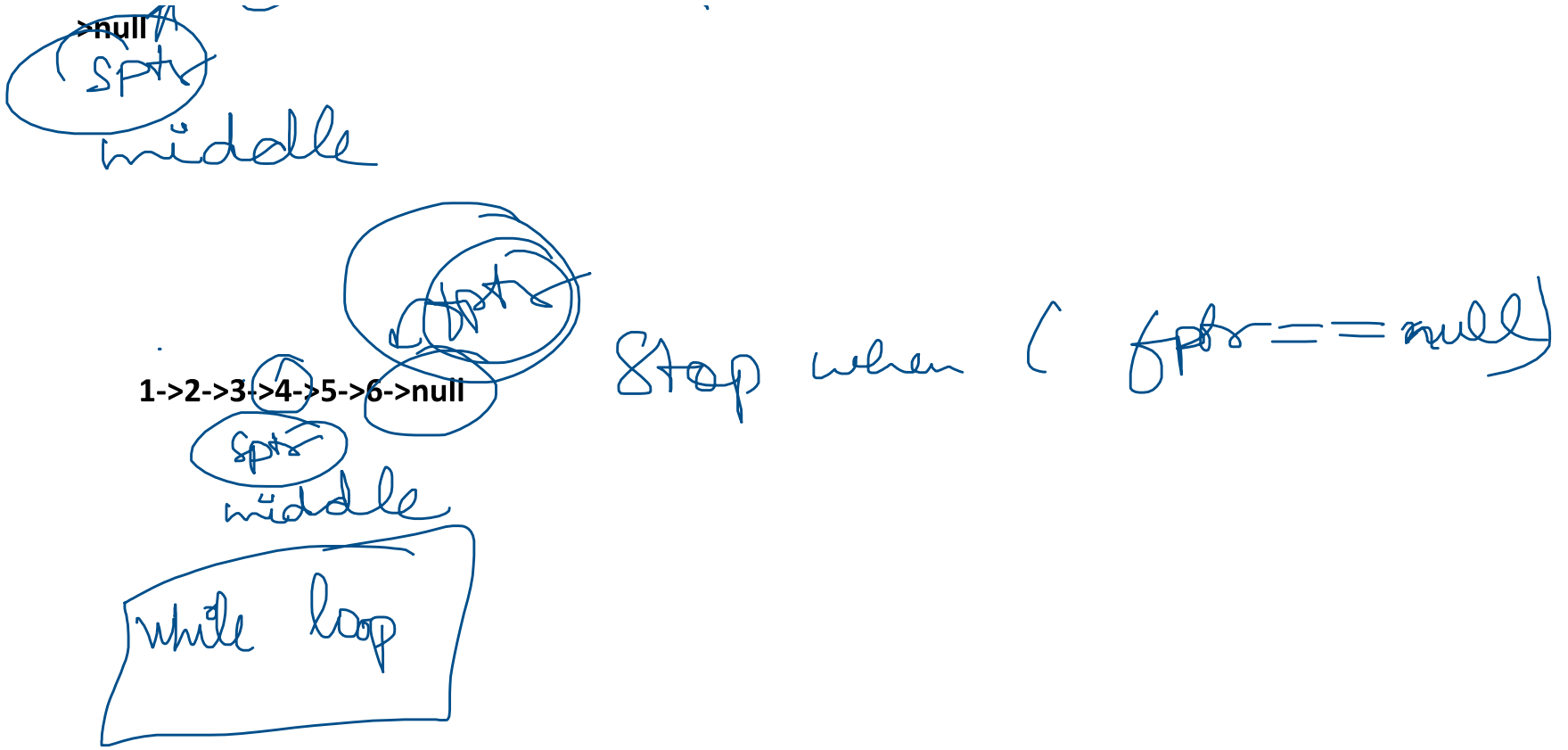
1->2->3->4->5->null

Fptr takes 2 jumps at a time.
Sptr takes 1 jump at a time.

1->2->3->4->5->

fptr

Stop when (fptr.next == null)


->null
sptr
middle
sptr
1->2->3->4->5->6->null
Stop when (fptr == null)
sptr
middle
while loop