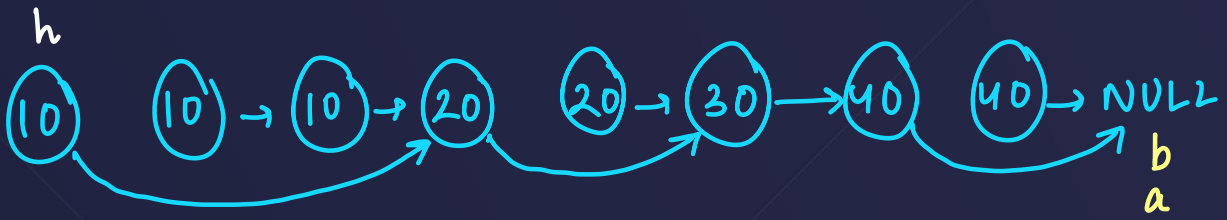
Linked List

Part - 3

Raghav Garg

Ques: Remove Duplicates from Sorted List [Leetcode - 83]

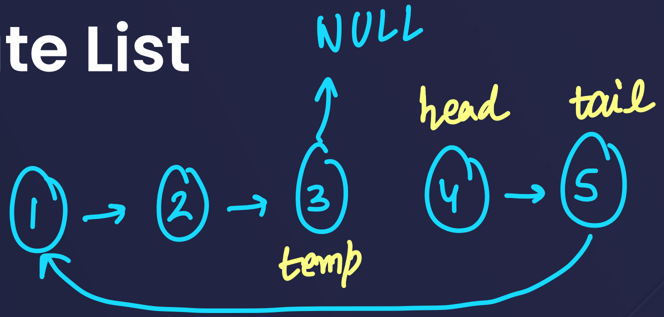


```
Node* a = head;
Node* b = head→next;
```

```
while (b != NULL){
    while (b != NULL && b→val == a→val)
        b = b→next;

    a→next = b;

    a = b;

    if (b != NULL) b = b→next;
}
```

Ques: Rotate List

[Leetcode - 61]



tail → next = head

head = temp → next

temp → next = NULL

1 2 3

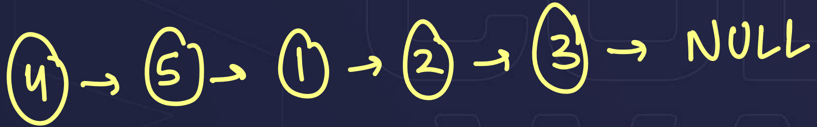
K = 2

K = 90

$\boxed{K > n}$

K = 10

K = K % n

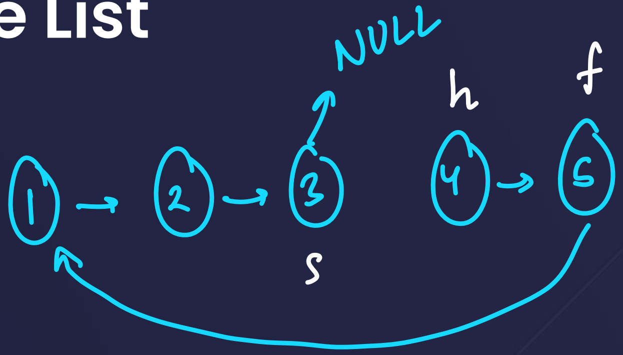


Ques: Rotate List

[Leetcode - 61]

$\boxed{M-2}$

× ×



$k = 2$

when → $\boxed{k < n}$

Steps

1) s = f = h

2) move fast k steps ahead

3) move slow & fast till f.next → null

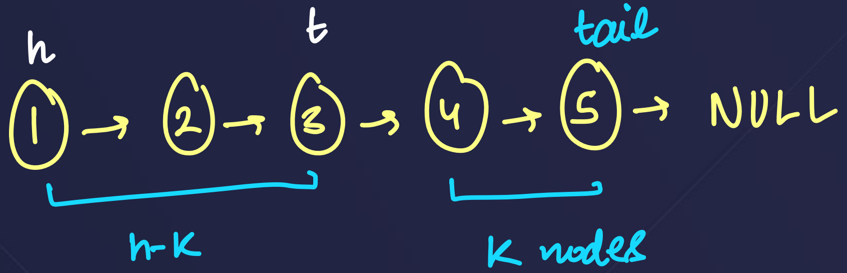
4) f → next = head

5) h = s → next

6) s → next = NULL

Ques: Rotate List

[Leetcode - 61]



$K = K \% n$

$K = \cancel{12}$ 2

$n = 5$

Ques: Spiral Matrix IV [Leetcode - 2326]

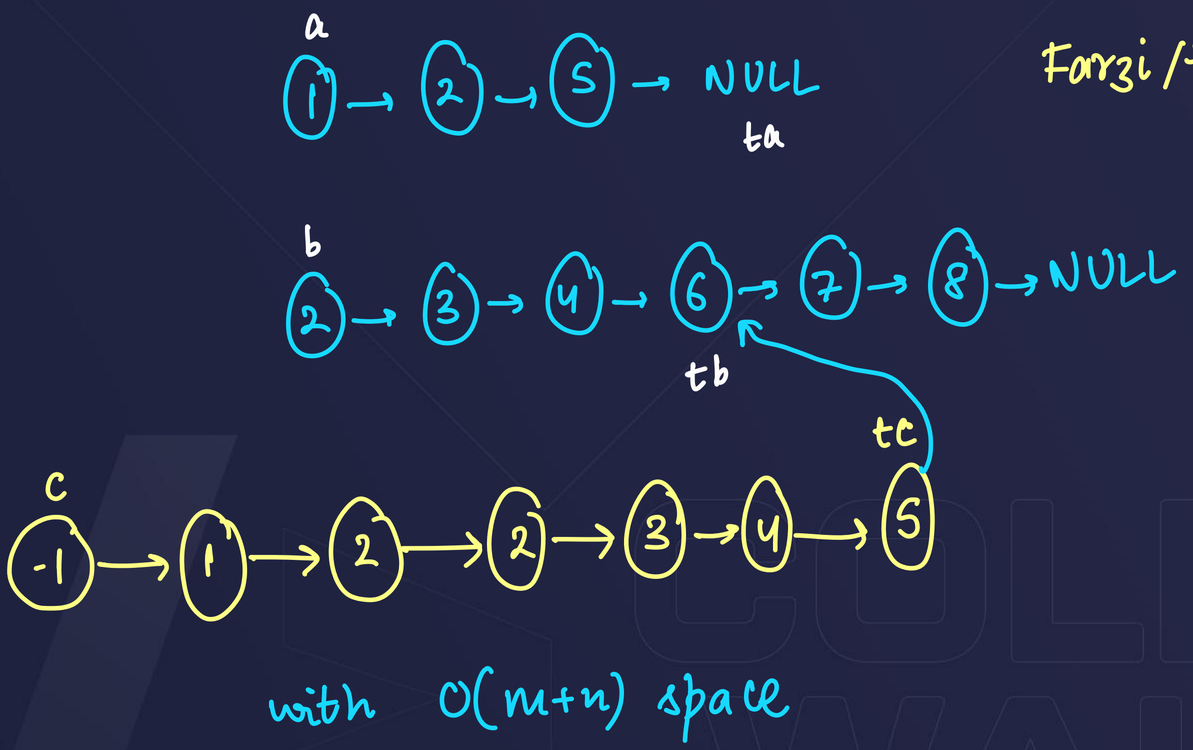
Spiral Matrix, Spiral Matrix II

$n = 3, m = 4$

$$\begin{tikzcd} \textcircled{1} \arrow[r] & \textcircled{2} \arrow[r] & \textcircled{3} \arrow[r] & \textcircled{4} \arrow[r] & \textcircled{5} \arrow[r] & \textcircled{6} \arrow[r] & \textcircled{7} \arrow[r] & \textcircled{8} \arrow[r] & \text{NULL} \end{tikzcd}$$

1	2	3	4
-1	-1	-1	5
-1	8	7	6

Ques: Merge 2 sorted lists [Leetcode - 21]

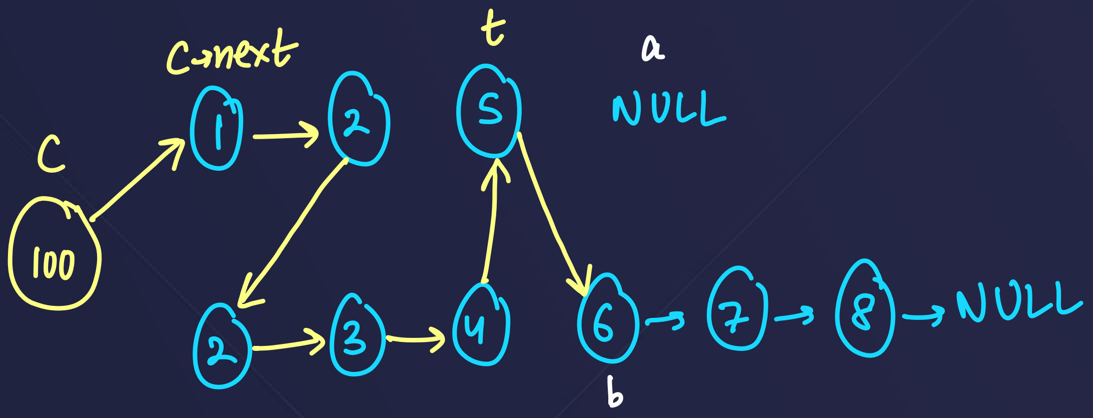


Farzi / Extra Node

T.C. = O(m+n)

S.C = O(m+n)/O(1)

Ques: Merge 2 sorted lists O(1) space [Leetcode - 21]



```
if (a→val <= b→val)
    t→next = a;
    a = a→next
    t = t→next
```

Ques: Merge k sorted lists [Leetcode - 23]

a

$$\begin{tikzcd} 1 \arrow[r] & 4 \arrow[r] & 5 \end{tikzcd}$$

b

$$\begin{tikzcd} 1 \arrow[r] & 3 \arrow[r] & 4 \end{tikzcd}$$

] merge(a, b)

c

$$\begin{tikzcd} 2 \arrow[r] & 6 \end{tikzcd}$$

d

$$\begin{tikzcd} 1 \arrow[r] & 1 \arrow[r] & 3 \arrow[r] & 4 \arrow[r] & 4 \arrow[r] & 5 \end{tikzcd}$$

c

$$\begin{tikzcd} 2 \arrow[r] & 6 \end{tikzcd}$$

$$\begin{tikzcd} 1 \arrow[r] & 2 \arrow[r] & 3 \end{tikzcd}$$

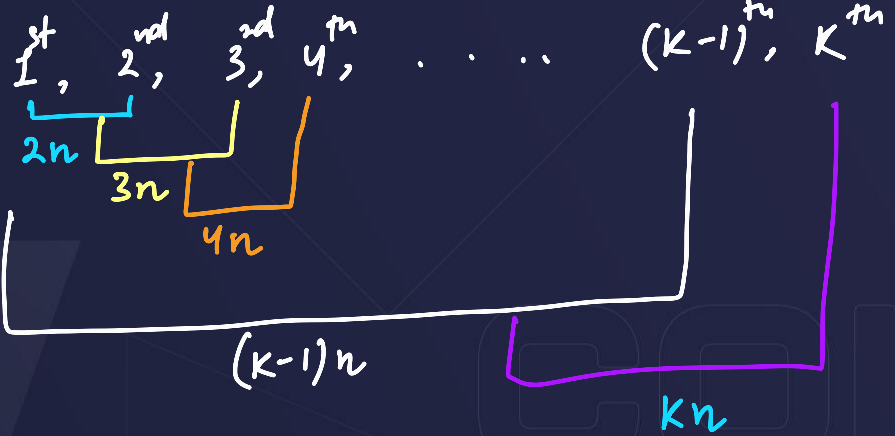
$$\begin{tikzcd} 4 \arrow[r] & 5 \arrow[r] & 5 & 8 \arrow[l] \arrow[r] & 9 \end{tikzcd}$$

$$\begin{tikzcd} 19 \arrow[r] & 29 \arrow[r] & 100 \arrow[r] & 1000 \end{tikzcd}$$

Ques: Merge k sorted lists [Leetcode - 23]

Time Complexity

→ K lists & each LL has on an average 'n' elements



$tno = 2n + 3n + 4n + \ldots\ldots (K-1)n + Kn$

$= n\left[2+3+4+\ldots\ldots K\right]$

$tno = n\left[\underline{1+2\cdots K} - 1\right]$

$= n\cdot\frac{K(K+1)}{2} - n$

$T.C. = O(n\cdot K^2)$

Ques: Merge k sorted lists [Leetcode - 23]

1, 2, 3, 4, 5, 6, 7, 8

$$\text{TNO} = n\cdot\frac{K(K+1)}{2} - n = n\cdot\frac{\overset{4}{\cancel{8}}\cdot 9}{\cancel{2}} - n = \boxed{35n}$$

$$\underbrace{1,\ 2}_{2n},\quad \underbrace{3,\ 4}_{2n},\quad \underbrace{5,\ 6}_{2n},\quad \underbrace{7,\ 8}_{2n}$$

$$\underbrace{a \qquad b}_{4n} \qquad \underbrace{c \qquad d}_{4n}$$

$$\underbrace{e \qquad\qquad f}_{8n}$$

$$g$$

$$\text{T.N.O} = \underline{8n} + \underline{8n} + \underline{8n}$$

$$= 24n$$

$$= O(n\cdot K\cdot \log K)$$

Ques: Merge k sorted lists [Leetcode - 23]

{ 9 } , }

, , a , b

7, 8 3, 4

,

5, 6 1, 2

$\underbrace{\qquad}$

c , d , e , f

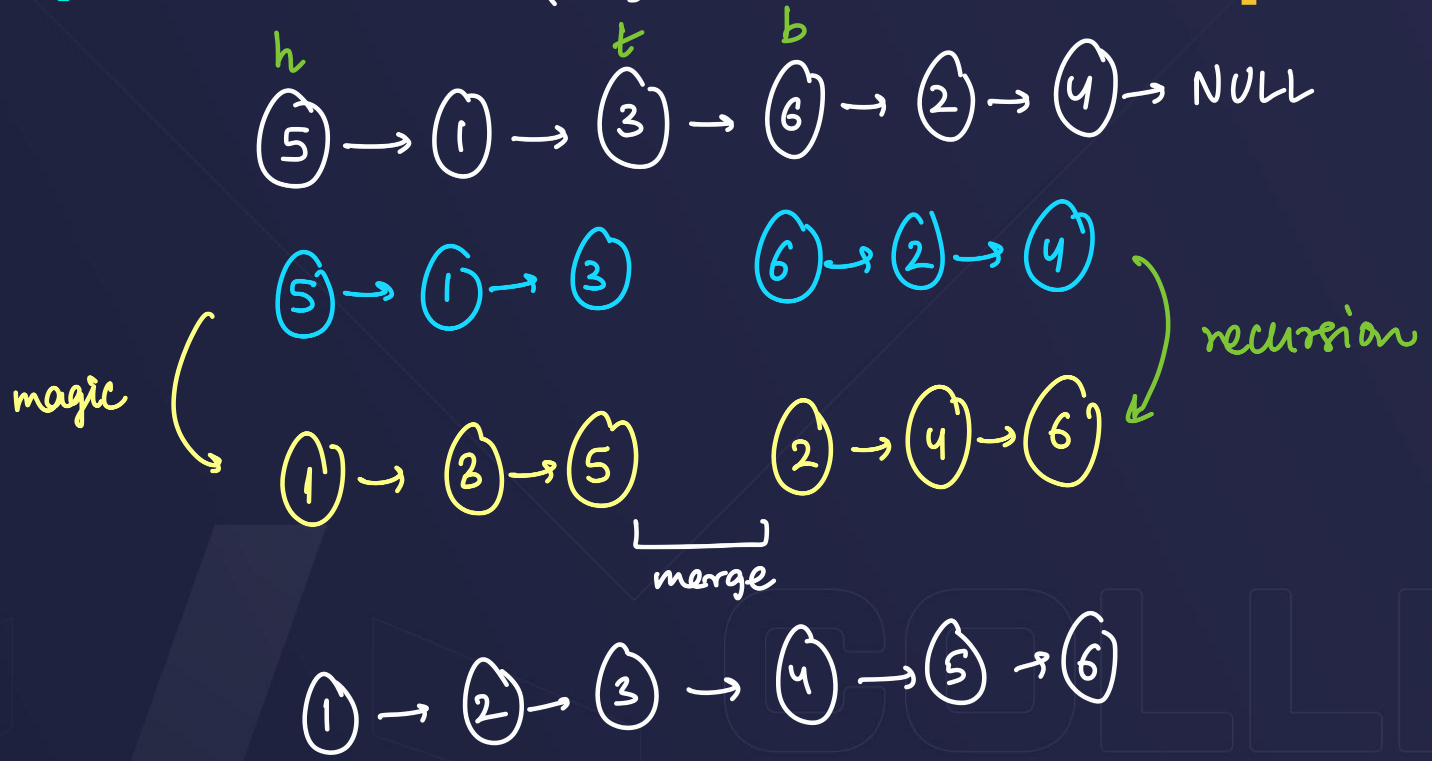
Ques: Merge k sorted lists [Leetcode - 23]

1, 2, 3 . . . . . . , 31, 32

$$n \cdot \frac{\overset{16}{\cancel{32}} \cdot 31}{\cancel{2}} - n = 495n$$

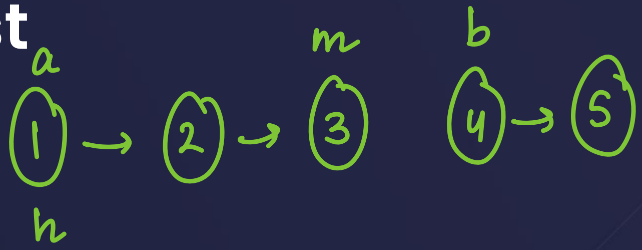
$$\rightarrow 32n + 32n + 32n + 32n + 32n = 160n$$

Ques: Sort List (merge Sort) O(1) [Leetcode - 148]



Ques: Sort List

[Leetcode - 148]



Ques: Partition List

[Leetcode - 86]

$$\begin{tikzcd}
\overset{\checkmark}{10} \arrow[r] & 80 \arrow[r] & \overset{\checkmark}{40} \arrow[r] & \overset{\checkmark}{30} \arrow[r] & 60 \arrow[r] & 70 \arrow[r] & \text{NULL}
\end{tikzcd}$$

$x = 41$

$$\begin{tikzcd}
10 \arrow[r] & 40 \arrow[r] & 30 \arrow[r] & 80 \arrow[r] & 60 \arrow[r] & 70 \arrow[r] & \text{NULL}
\end{tikzcd}$$

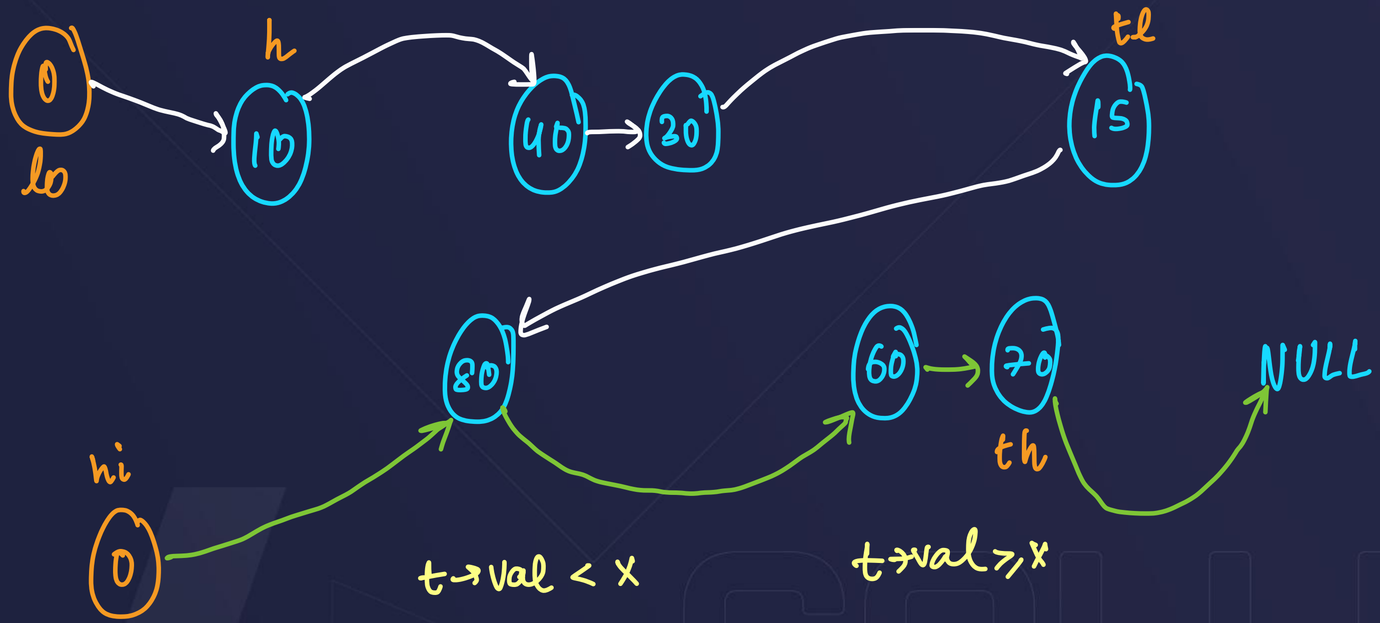
Quick Sort

$\underline{50}$, 40, 40, $\underline{30}$, 80, 10, 60

$\underline{30}$, 40, 40, $\underline{50}$, 80, 10, 60

Ques: Partition List

[Leetcode - 86]



t→val < x

tl→next = t

t = t→next

tl = tl→next

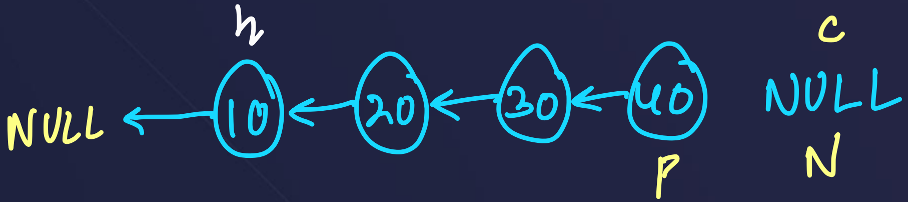
t→val >= x

th→next = t

t = t→next

th = th→next

Ques: Reverse Linked List (Iterative) [Leetcode - 206]



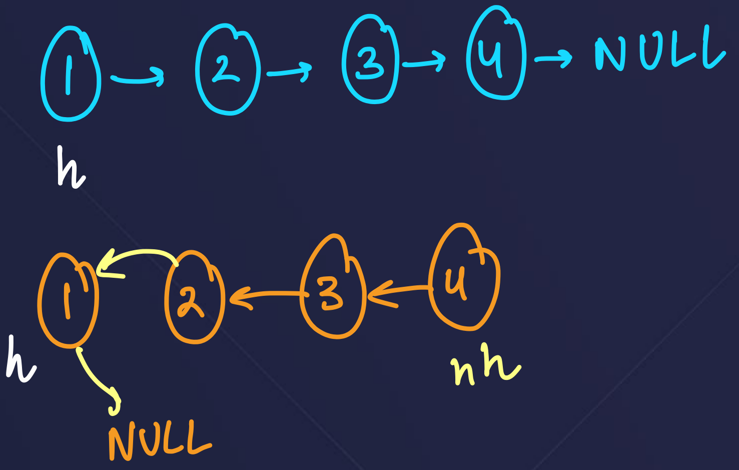
prev , curr, Next

```
while (curr){
    Next = curr → next
    curr → next = prev
    prev = curr
    curr = Next
}
return prev;
```

Time Complexity → O(n)

Space Complexity → O(1)

Ques: Reverse Linked List (Recursive) [Leetcode - 206]



T.C. = O(n)

S.C. = O(n)

nhead = reverse(head→next);

h→next→next = h

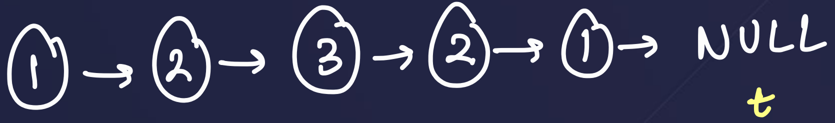
h→next = NULL

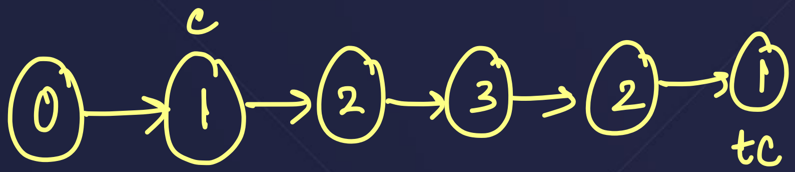
return newhead

Ques: Palindrome Linked List [Leetcode - 234]

M-I : Bad [T.C. = $O(n^2)$] → getNodeAt(idx) S.C = $O(1)$

M-2 :





c = reverse(c)

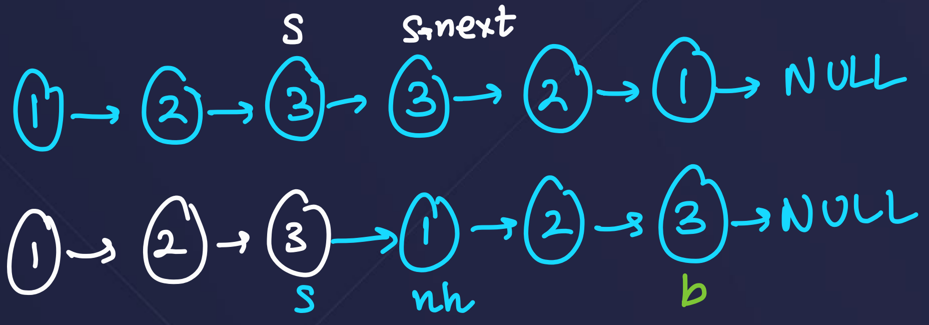
→ check every element of original & duplicate(reversed)

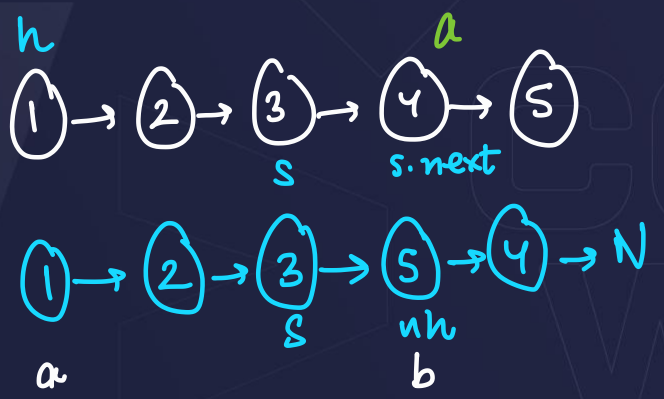
Okayish Method → T.C. = $O(n)$
S.C. = $O(n)$

Ques: Palindrome Linked List [Leetcode - 234]

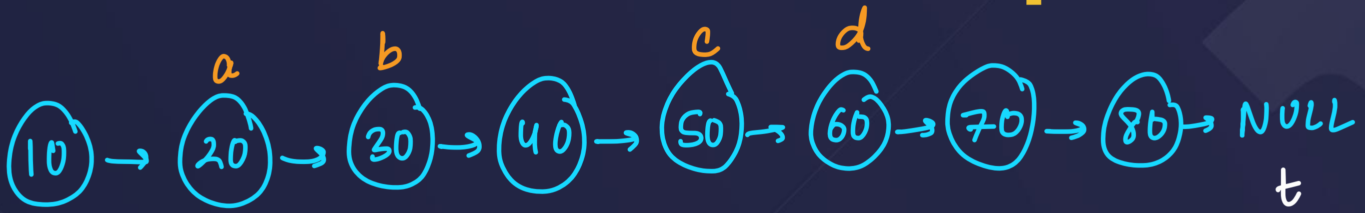
M-3 : T.C. = O(n) , S.C. = O(1)

Hint : if the first & second halves of LL are reverse of each other → true





Ques: Reverse Linked List II [Leetcode - 92]



```
while(temp){
    if (n == l-1) a = temp
    if(n == l) b = temp
    if (n == r) c = temp
    if (n == r+1) d = temp
    temp = temp -> next
    n++
}
```

temp = head

n = ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9

l = 3

r = 5

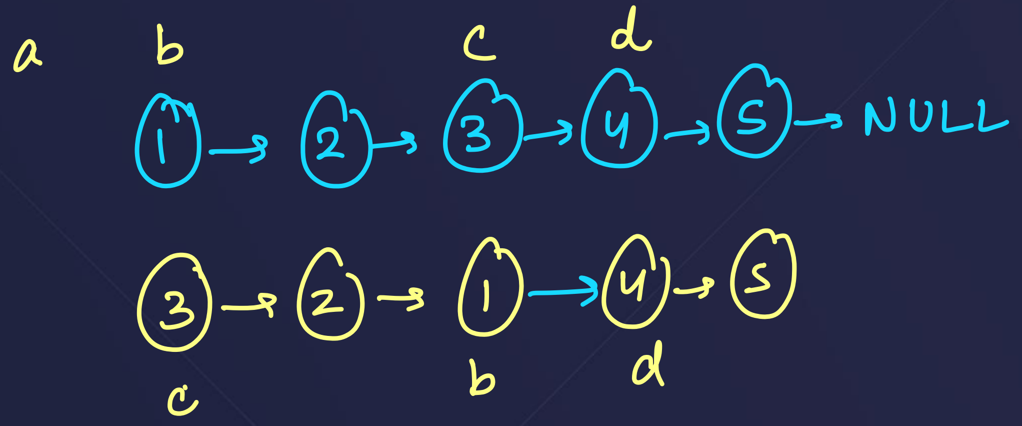
Hint : Break the list into 3 lists -

$\boxed{1 \text{ to } l-1}$, $\boxed{l \text{ to } r}$, $\boxed{r+1 \text{ to } n}$

↓

reverse

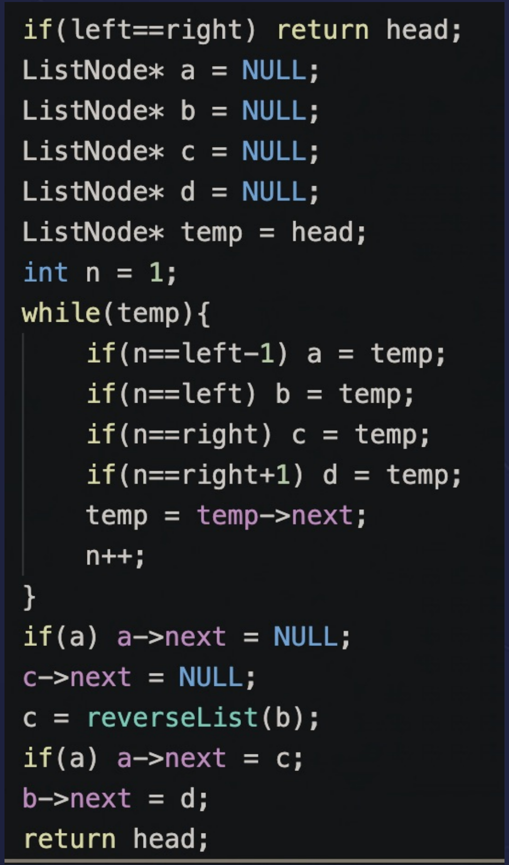
Ques: Reverse Linked List II [Leetcode - 92]



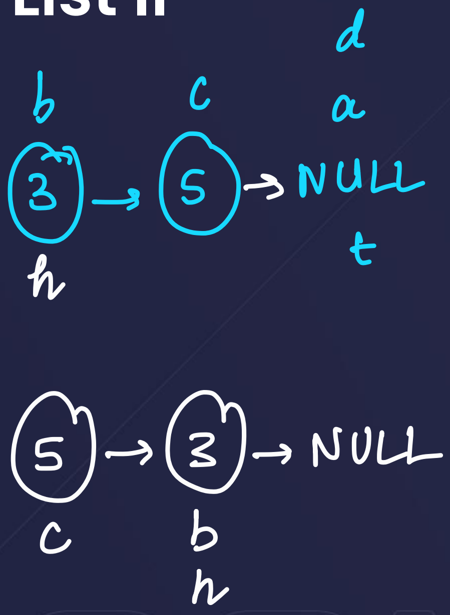
left = 1

right = 3

Ques: Reverse Linked List II [Leetcode - 92]



```
if(left==right) return head;
ListNode* a = NULL;
ListNode* b = NULL;
ListNode* c = NULL;
ListNode* d = NULL;
ListNode* temp = head;
int n = 1;
while(temp){
    if(n==left-1) a = temp;
    if(n==left) b = temp;
    if(n==right) c = temp;
    if(n==right+1) d = temp;
    temp = temp->next;
    n++;
}
if(a) a->next = NULL;
c->next = NULL;
c = reverseList(b);
if(a) a->next = c;
b->next = d;
return head;
```

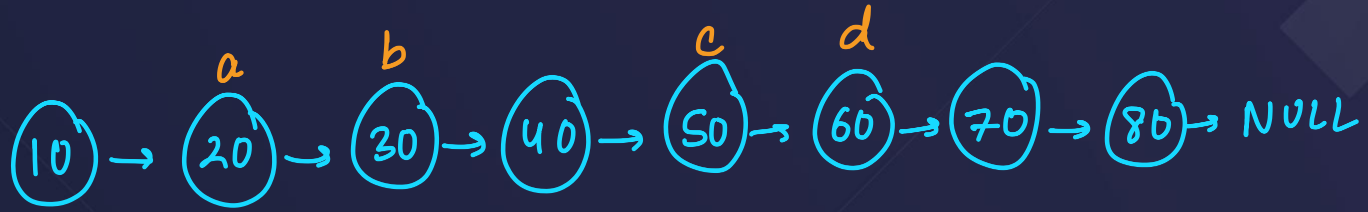


left = 1
right = 2

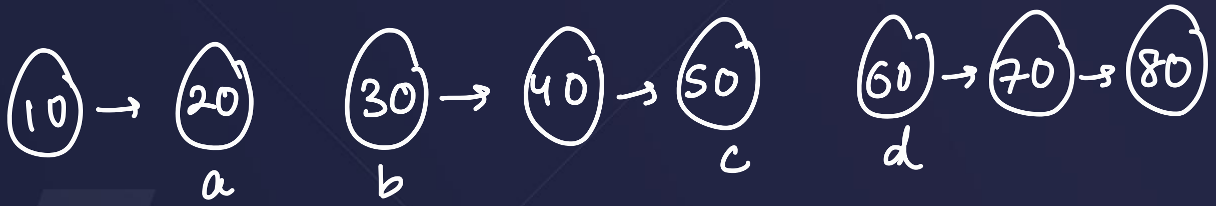
n = ~~1~~ 2

Ques: Reverse Linked List II

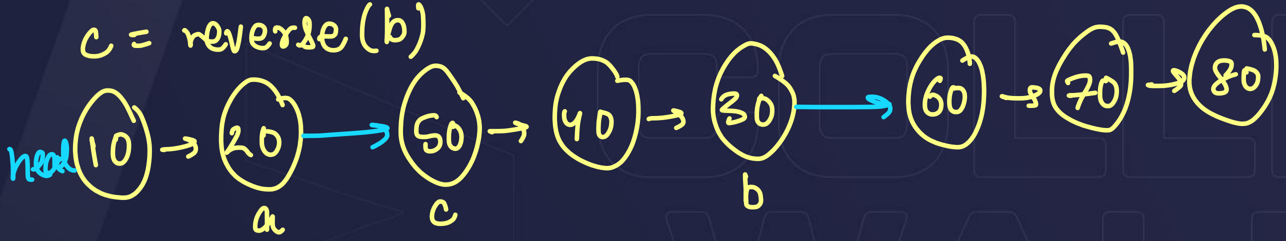
[Leetcode - 92]



a→next = NULL , c→next = NULL ,



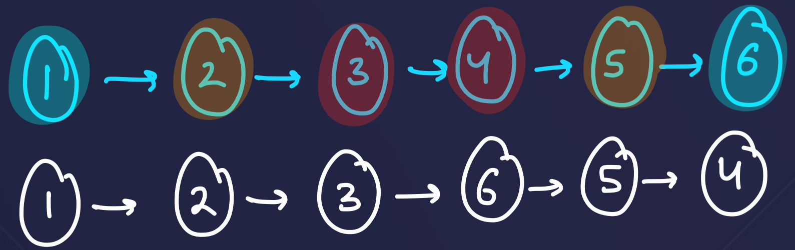
c = reverse(b)



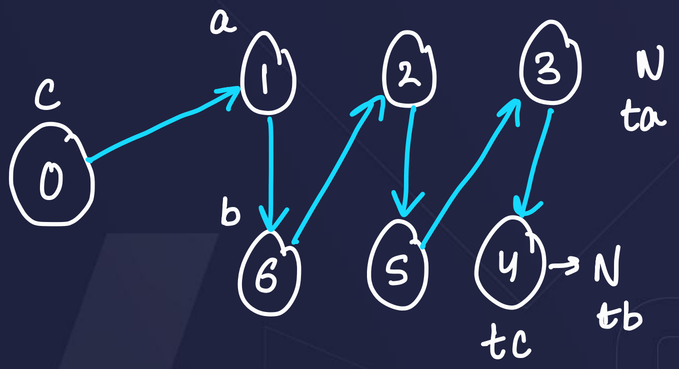
a→next = c , b→next = d

Ques: Reorder List

[Leetcode - 143]



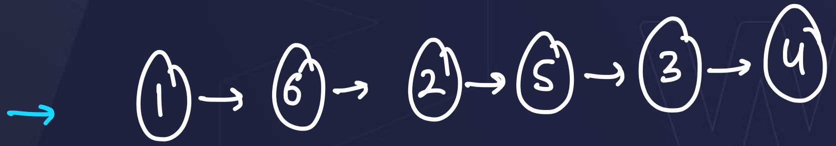
#Hints
1) Palindrome LL
2) Partition LL/ Merge 2 Sorted
3) Farzi Node



loop

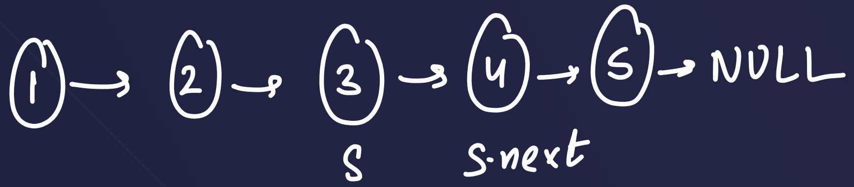
```
tc->next = ta
tc = tc->next
ta = ta->next
tc->next = tb
tc = tc->next
tb = tb->next
```

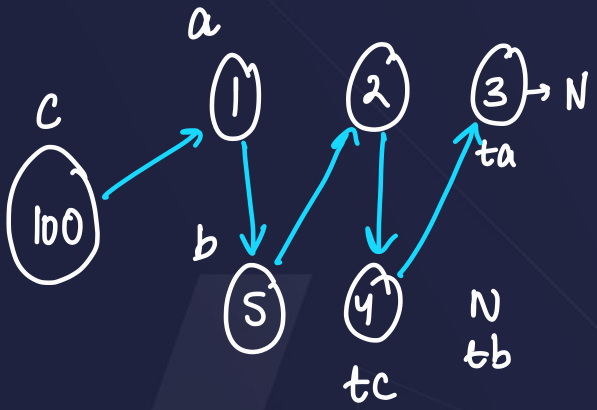
tc -> next = null

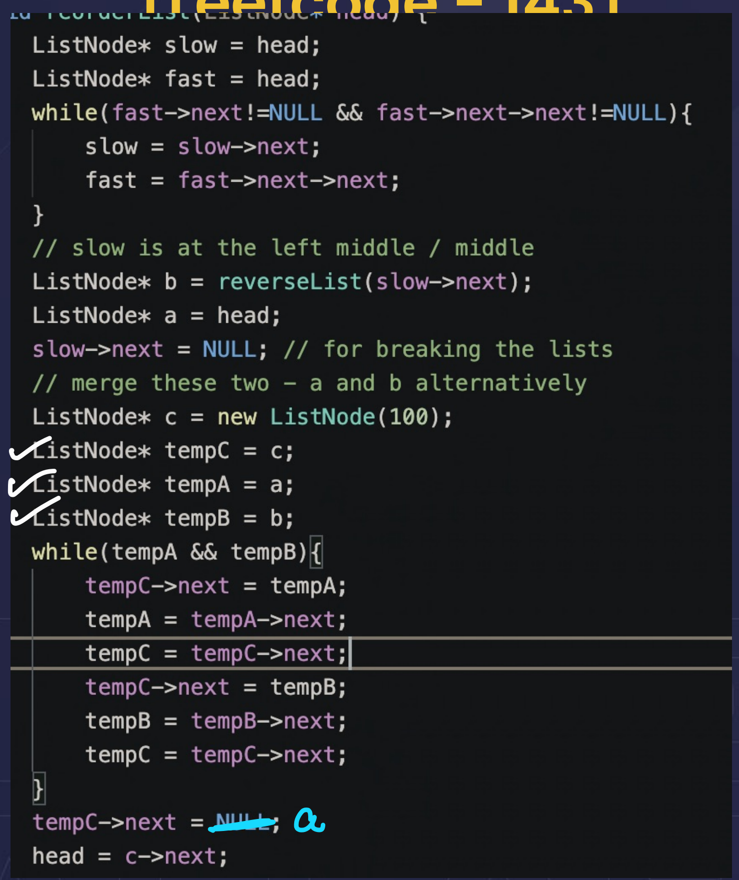


Ques: Reorder List

[Leetcode - 143]







```
id reorderList(ListNode* head) {
ListNode* slow = head;
ListNode* fast = head;
while(fast->next!=NULL && fast->next->next!=NULL){
    slow = slow->next;
    fast = fast->next->next;
}
// slow is at the left middle / middle
ListNode* b = reverseList(slow->next);
ListNode* a = head;
slow->next = NULL; // for breaking the lists
// merge these two - a and b alternatively
ListNode* c = new ListNode(100);
ListNode* tempC = c;
ListNode* tempA = a;
ListNode* tempB = b;
while(tempA && tempB){
    tempC->next = tempA;
    tempA = tempA->next;
    tempC = tempC->next;
    tempC->next = tempB;
    tempB = tempB->next;
    tempC = tempC->next;
}
tempC->next = ~~NULL~~; a
head = c->next;
```

Next Lecture

More problems on Linked Lists!