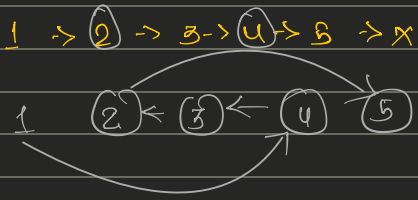
Swapping Nodes in a Linked List



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

(1) -> Null || head -> next == 0

X

1 -> X

Return head

(2) -> lpos , rpos k = 2

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

lpos

Rpos k = 2

lpos = rpos = 3

k = 5

k = 2 lpos = 2

Rpos = len - k + 1 = 5 - 2 + 1 = 4

lpos == rpos

Return head (No swap)

(3) -> len == 2

head

next node

1 -> 2 -> X

k = 1, 2

lpos

rpos

=> nextNode -> next = head
head = nextNode

(4) -> lpos = = 1 ——————> k = 1 or N

1 -> 2 -> 3 -> x k = 3
↑ rpos ↑ lpos

kabi aisa hua toh if (rpos < lpos){

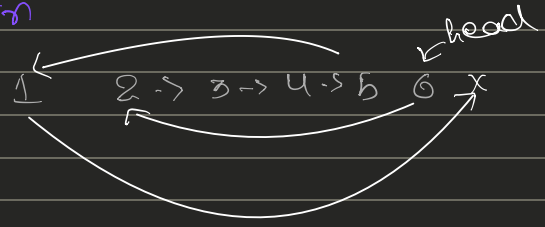
rpos -> ko -> Right ki taraf rakhna chahte hai

lpos -> ko -> left taraf

Swap (rpos, lpos)
}

lp rp rpos
1 -> 2 -> 3 -> 4 -> 5 -> 6 -> x
ln rn rsave

(1) -> rn -> next = ln -> next
rp -> next = ln
ln -> next = rsave
head = rn



=> 6 -> 2 -> 3 -> 4 -> 5 -> 1 -> x
head

(5) ⇒ 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → X

lp, LN, RN, k = 4

xb, lpos, rpos, xsave

No of Nodes b/w swap Nodes = 0

lp->next = xp->next

xn->next = xb

xp->next = xsave

1 → 2 → 3 ⟵ 4 ⟵ 5 → 6 → 7 → 8 → X

⇒ 1 → 2 → 3 → 5 → 4 → 6 → 7 → 8 → X

(6) ⇒ No of Nodes B/w swap Nodes ≥ 1

1 → 2 → 3 → 4 → 5 → 6 → X

LN, RN, lp, lpos, xb, Rpos, xsave

k = 2

lpos = 2

Rpos = 6 - 2 + 1 = 5

lp->next = xn

xn->next = LN->next

xp->next = LN

LN->next = xsave

1 2 3 → 4 5 6 → X

LN, xb, xN, xsave, lp

⇒ 1 → 5 → 3 → 4 → 2 → 6 → X

TC -> O(n)

Space -> O(1)

(N-2) -> Swap values

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

lpos rpos

Store lpos value

int l = lpos->val

lpos->val = rpos->val

rpos->val = l

It is like

Swapping

two Numbers