25. Reverse Nodes in k-Group

Hard 12.8K 628

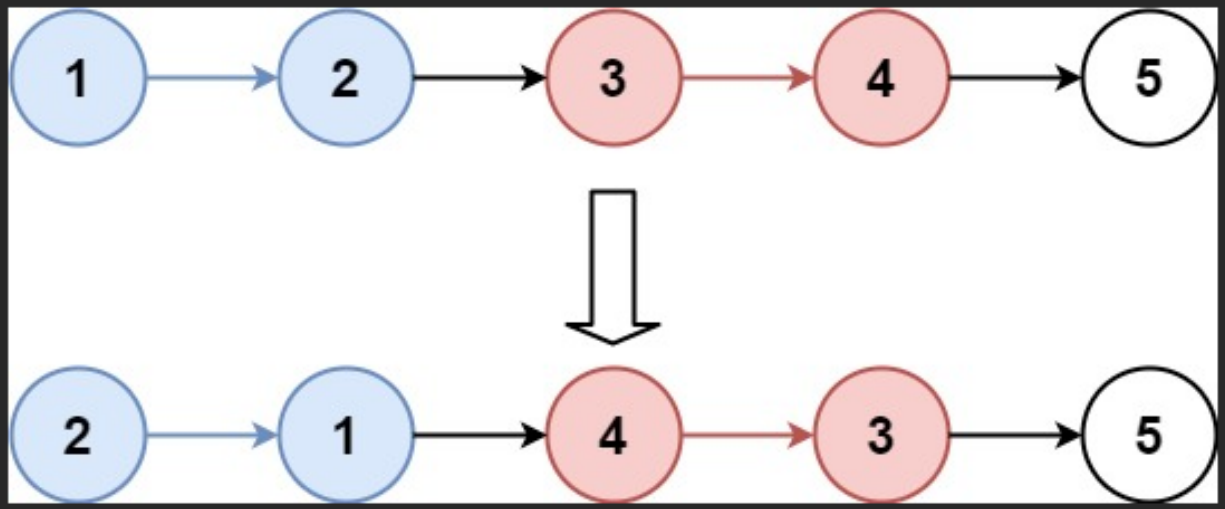
Companies

Given the `head` of a linked list, reverse the nodes of the list `k` at a time, and return *the modified list*.

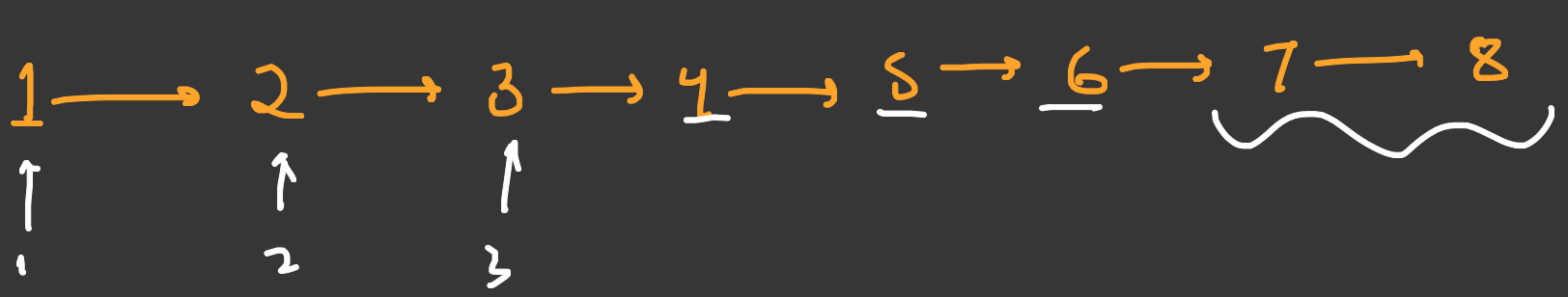
`k` is a positive integer and is less than or equal to the length of the linked list. If the number of nodes is not a multiple of `k` then left-out nodes, in the end, should remain as it is.

You may not alter the values in the list's nodes, only nodes themselves may be changed.

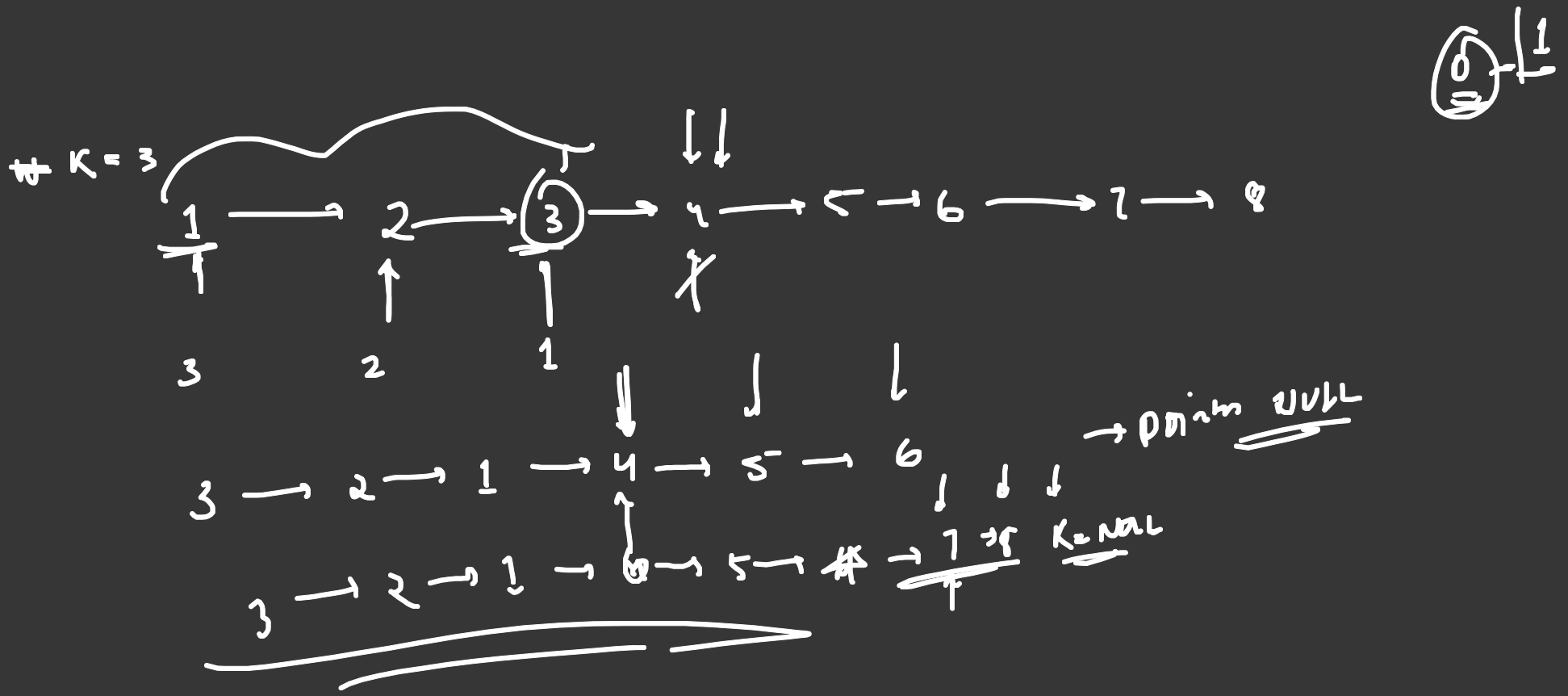
Example 1:




```
Input: head = [1,2,3,4,5], k = 2
Output: [2,1,4,3,5]
```


1 → 2 → 3 → 4 → 5 → 6 → 7 → 8
K = 3
1
2
3

3 → 2 → 1 → 6 → 5 → 4 → 7 → 8


K = 3
1 → 2 → 3 → 4 → 5 → 6 → 7 → 8
3
2
1
3 → 2 → 1 → 4 → 5 → 6
→ prim null
3 → 2 → 1 → 6 → 5 → 4 → 7 → K.Null

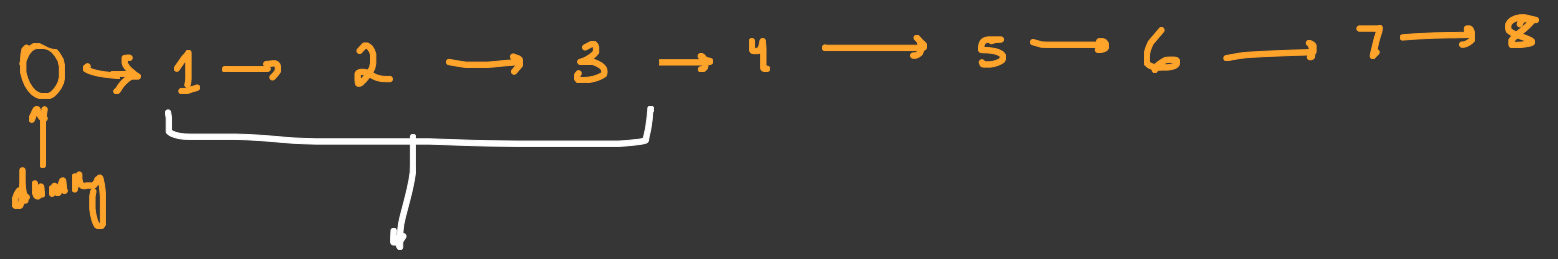
Approach:

→ Iterative → we have to reverse k size linked list.

k = 3

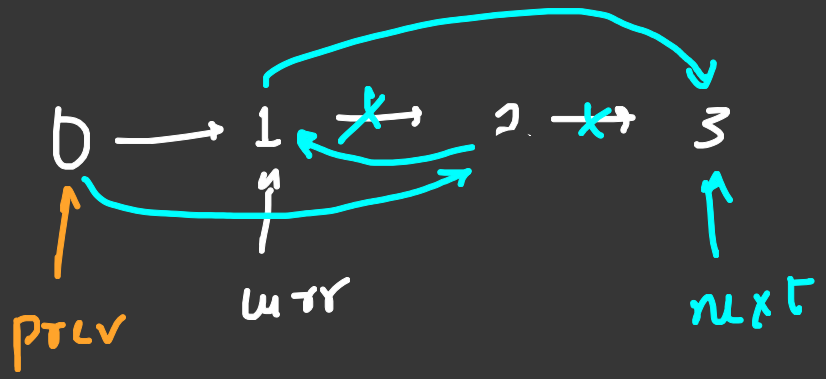
1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 len = 8

→ First we have to find len so that we know how many k size reverse grp is there.



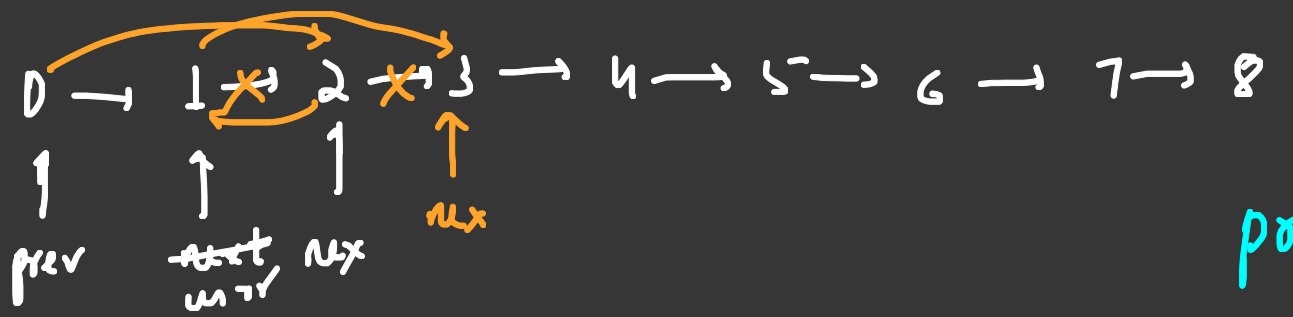
For K size we have
to do K-1 operations

2 → 1



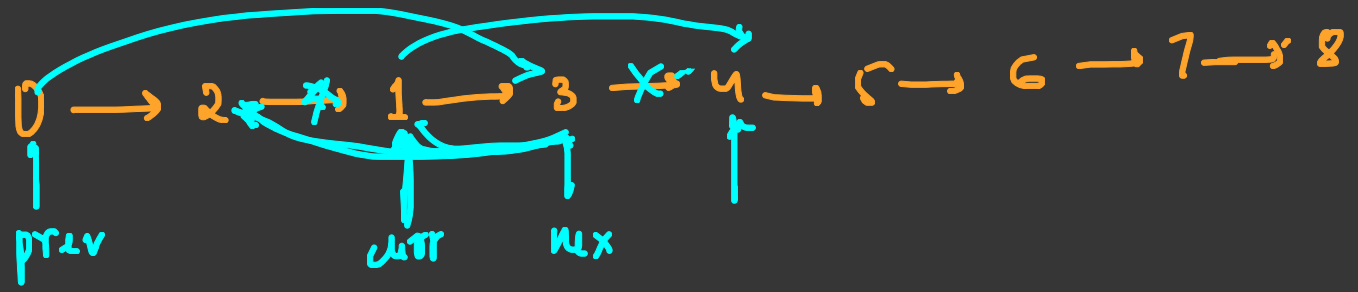

```
curr → next = next → next
next → next = prev → next
prev → next = nx
nx = curr → next;
```

2 → 1 → 3 → 4



prev 3

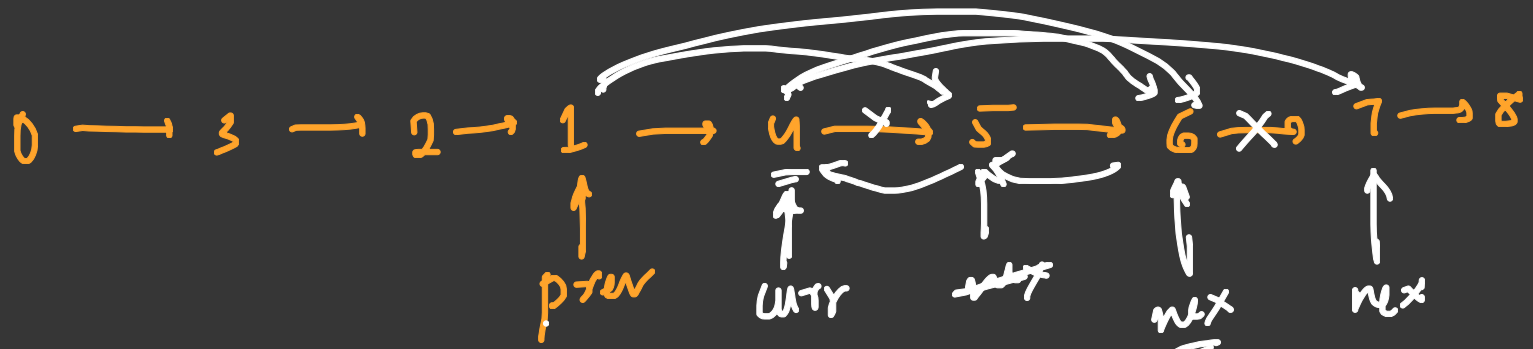
curr → nxt



5 → 4

1 → 5 → 4 → 6

prev = curr



1 → 6 → 5 → 4 → 7