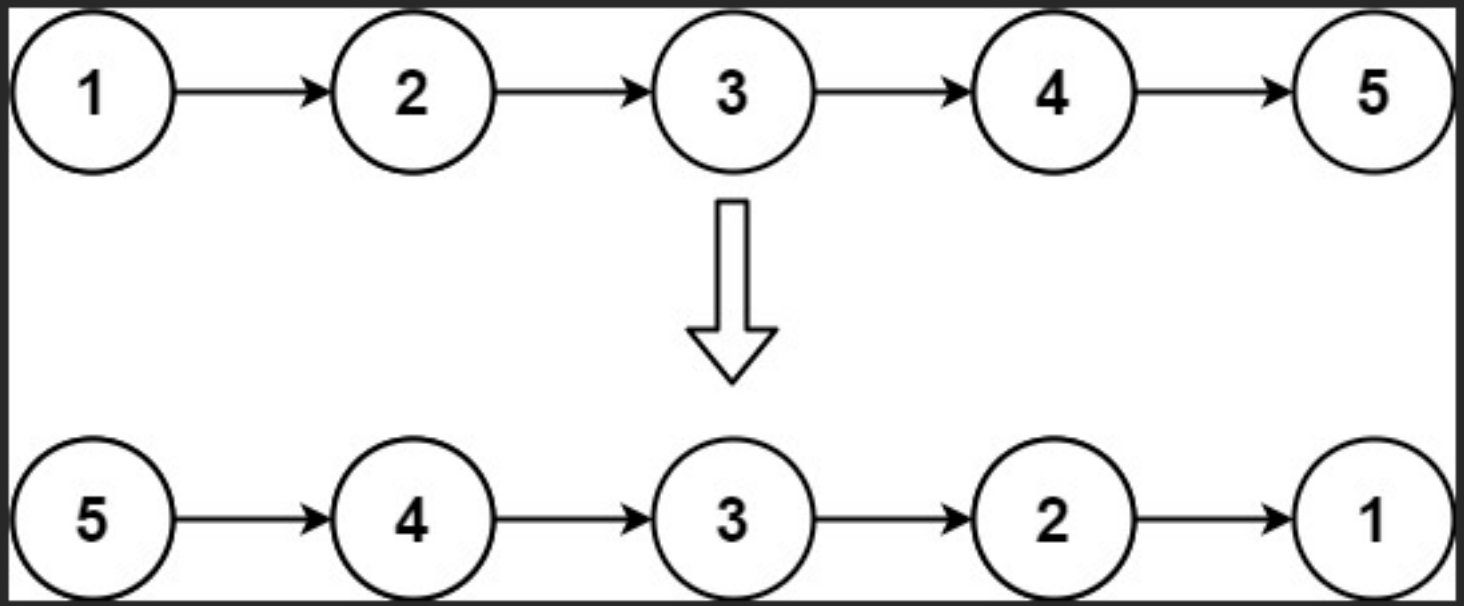
Reverse the Linked List

Given the `head` of a singly linked list, reverse the list, and return *the reversed list*.

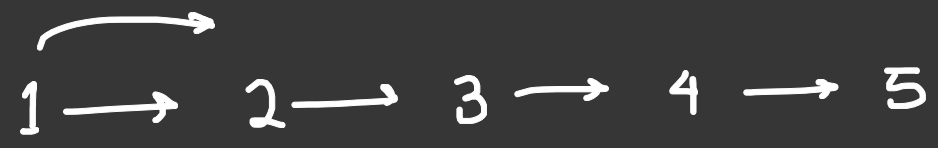
Example 1:

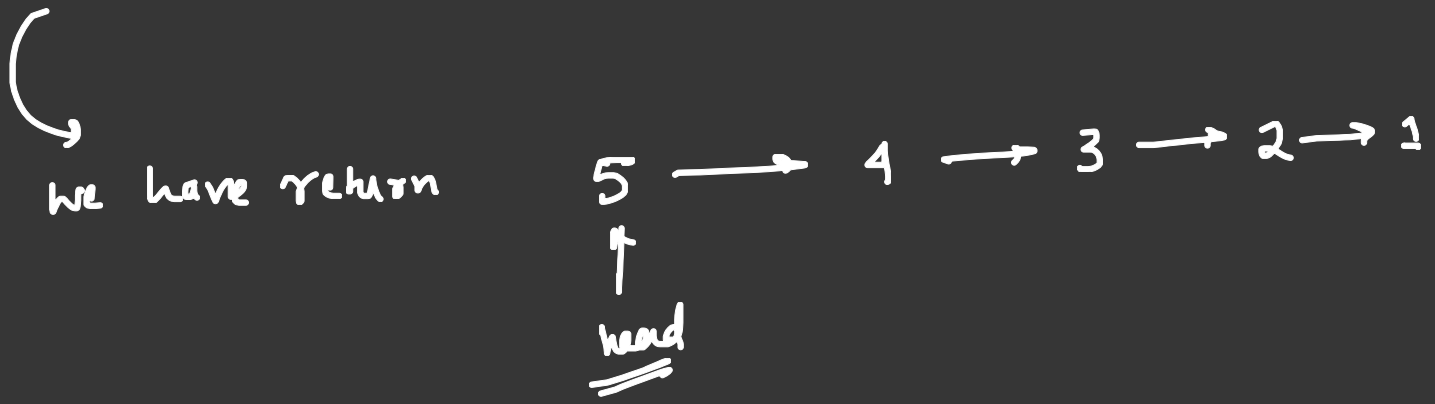


> **Input:** head = [1,2,3,4,5]
> **Output:** [5,4,3,2,1]

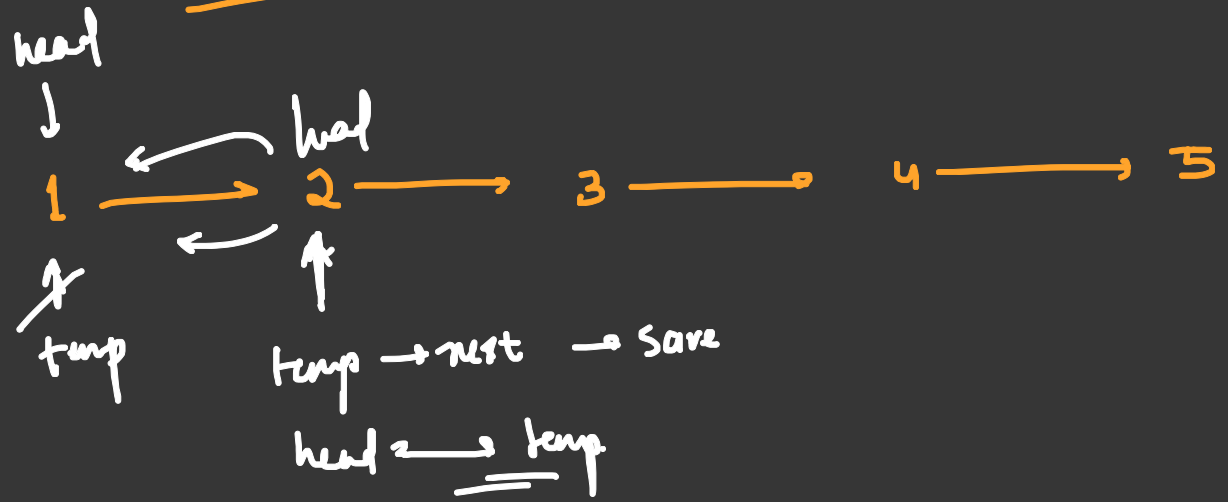
Explanation:

Given, head → node





Brute : First approach come to mind



Current

head → 1 → 2

1 → 3

3 → 4

4 → 5

5 → NULL

1 → null

2 → 1

3 → 2

4 → 3

5 → 4

head

We require prev, curr, next

iterator

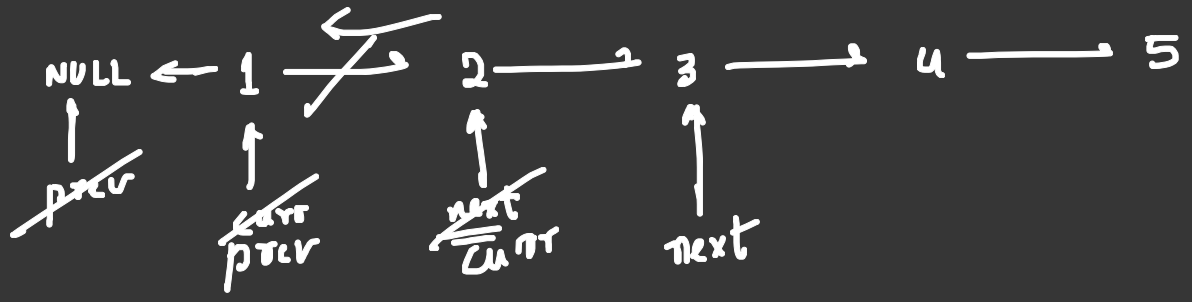
1

prev = null

curr = head

curr → next = prev

A



curr → next = prev

prev = curr

curr = next

1 → NULL

So, iterative method is:

- We keep 3 pointer prev, curr, next. We iterate on curr.

Change curr → next = Prev

Prev = Curr

Curr = next

next = Curr → next

Lastly move head to last.

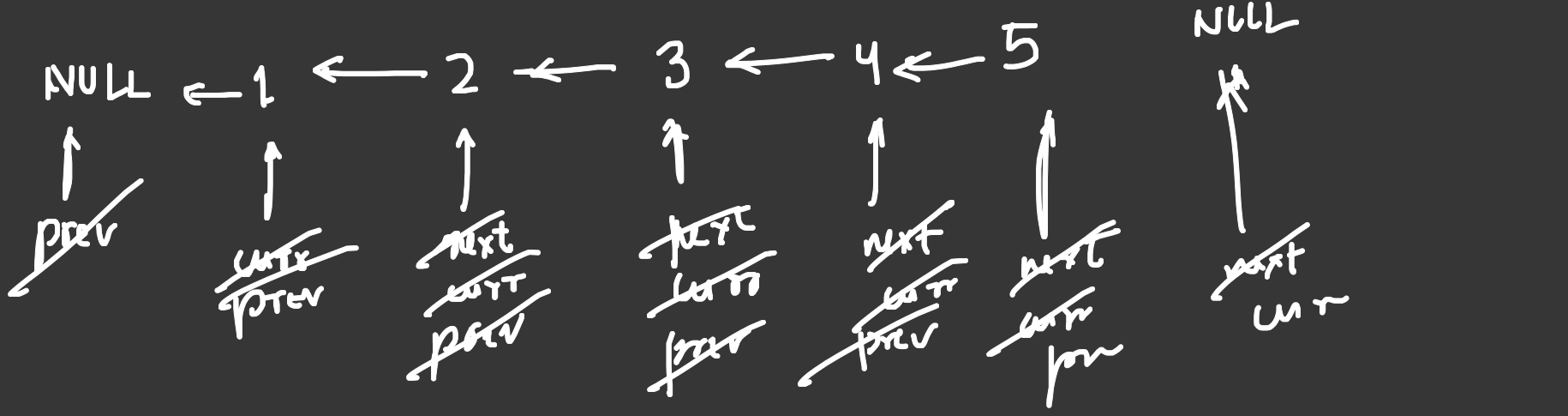
prev

curr → next = prev

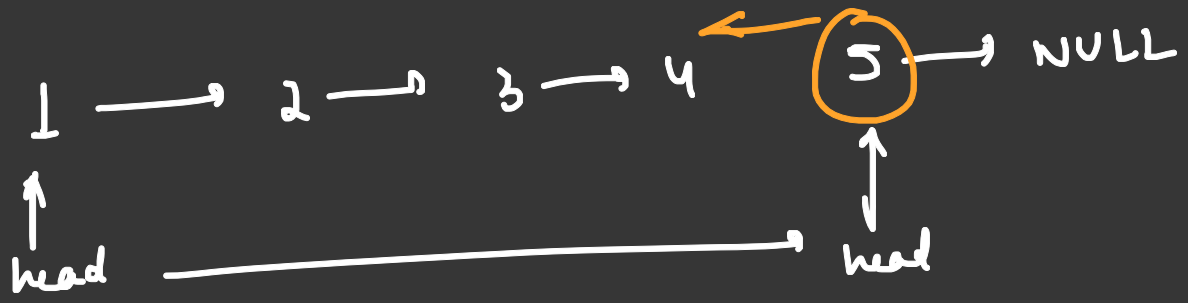
prev = curr

curr = next

next = curr → next



Recursively:



recursively go to end. → reverse the node

for eg:

4 → next → next = head

4 → head → next → next = head

head → next = NULL

return new head

T.C. = O(N)

S.C → O(N)