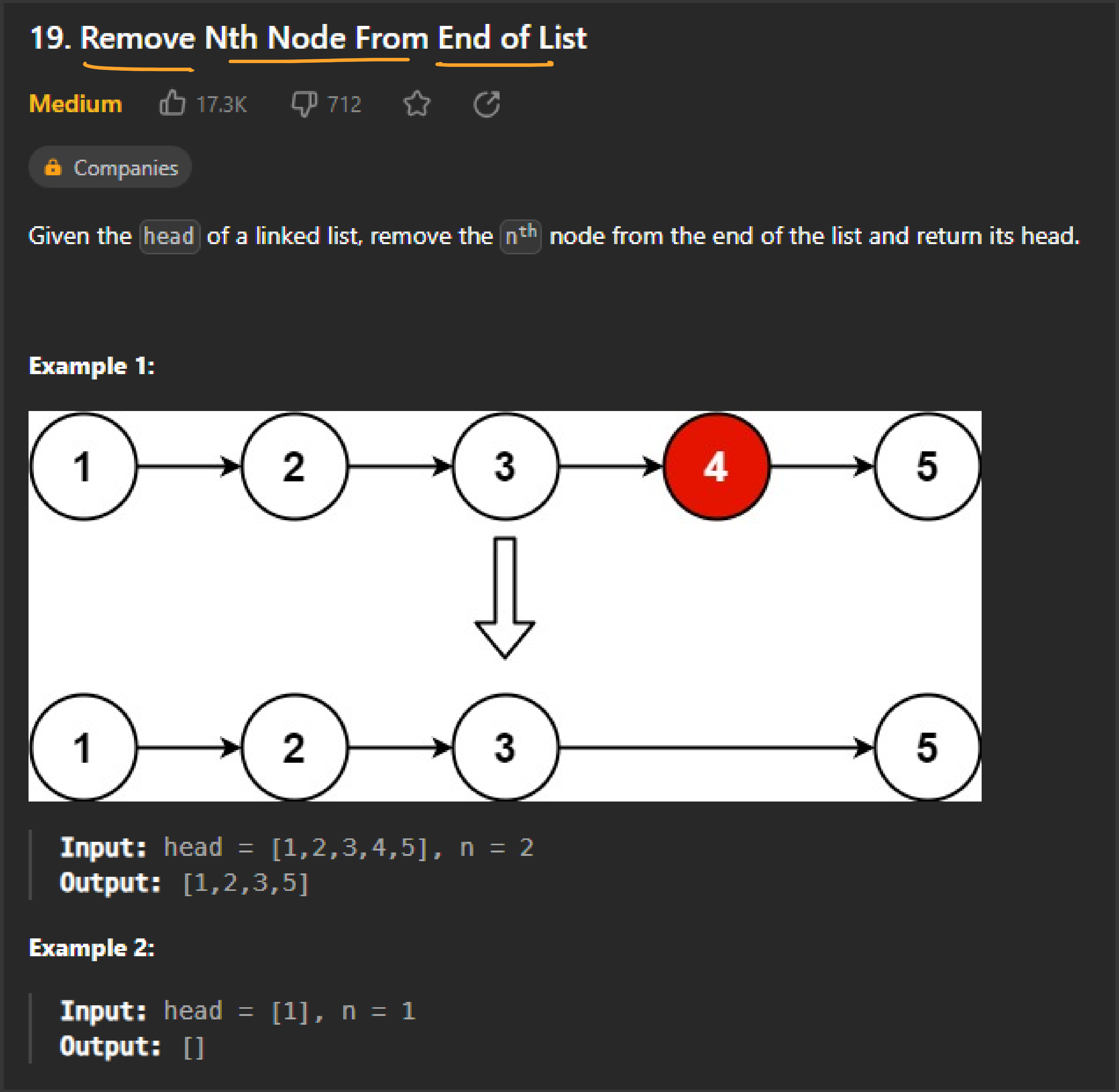


19. Remove Nth Node From End of List

Medium 17.3K 712

Companies

Given the `head` of a linked list, remove the n^{th} node from the end of the list and return its head.

Example 1:

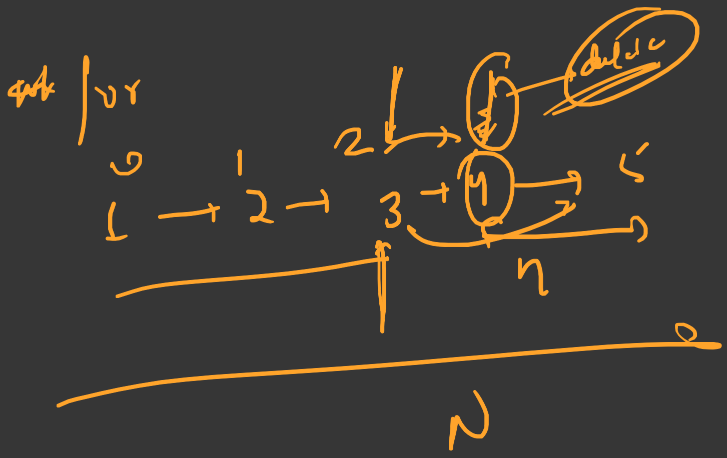
> **Input:** head = [1,2,3,4,5], n = 2
> **Output:** [1,2,3,5]

Example 2:

> **Input:** head = [1], n = 1
> **Output:** []

Brute force:

Idea → find the len of the linked list $N - n = X$



3

$O(N) + O(X)$

$O(N + N - n)$

$O(2N - n)$

$5 - 2 = 3$

$O(2N)$

Worst Case

• find the length of linked list $N - n = x$

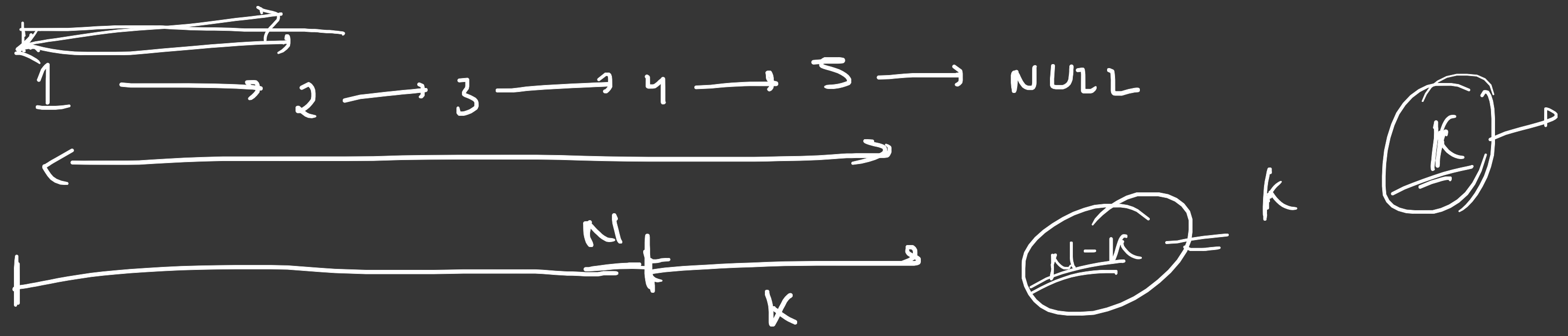
for(i=0; i<x; i++)   3 → 5

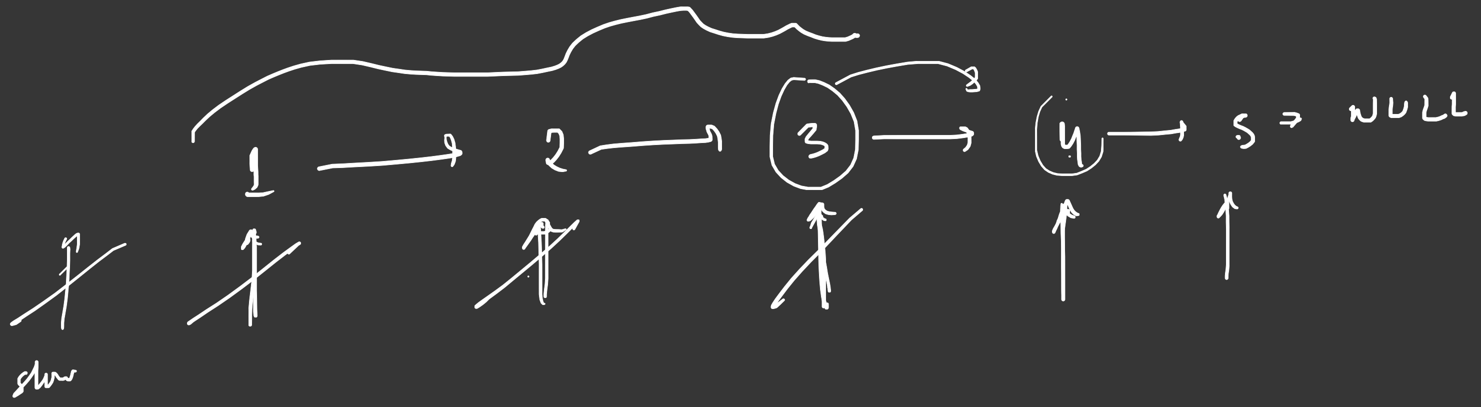
temp → next   2 ~~next~~   temp → next → next

todelete = temp → next

delete the todelete;

Idea





- Idea is that we make two pointer, one slow and fast idea is that we move fast pointer K time faster than slow and as fast reach end slow → next would me node which is to be deleted.

Psudocode:

```
start = new Node() // dummy Node

slow = start, fast = start
for(i=0; i<n; i++)
    fast = fast → next;

while( fast → next != NULL)

    slow  slow → next

    fast = fast → next

slow → next = slow → next → next

return start → next;
```

Time Complexity :

$O(N)$ S.C. : $O(1)$