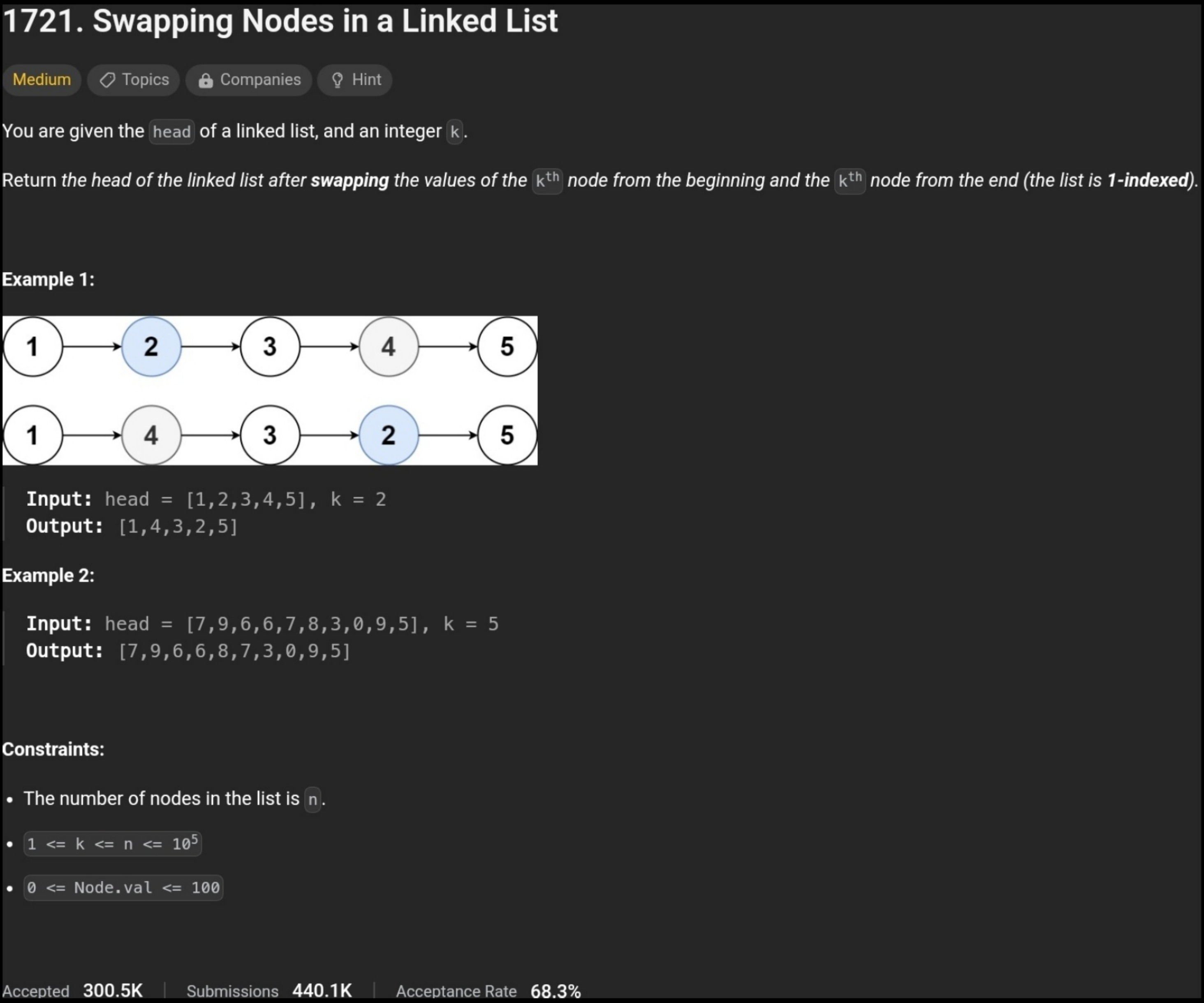


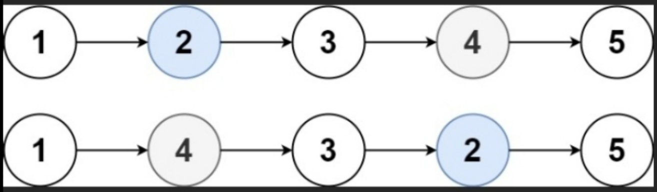
1721. Swapping Nodes in a Linked List

Medium · Topics · Companies · Hint

You are given the `head` of a linked list, and an integer `k`.

Return *the head of the linked list after* ***swapping*** *the values of the* k^{th} *node from the beginning and the* k^{th} *node from the end (the list is* ***1-indexed****).*

Example 1:



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

Example 2:

```
Input: head = [7,9,6,6,7,8,3,0,9,5], k = 5
Output: [7,9,6,6,8,7,3,0,9,5]
```

Constraints:

- The number of nodes in the list is `n`.
- `1 <= k <= n <= 10^5`
- `0 <= Node.val <= 100`

Accepted 300.5K | Submissions 440.1K | Acceptance Rate 68.3%

Approach 1: Two passes

1^{st} pass : find length of list

2^{nd} pass : keep pointers on required nodes

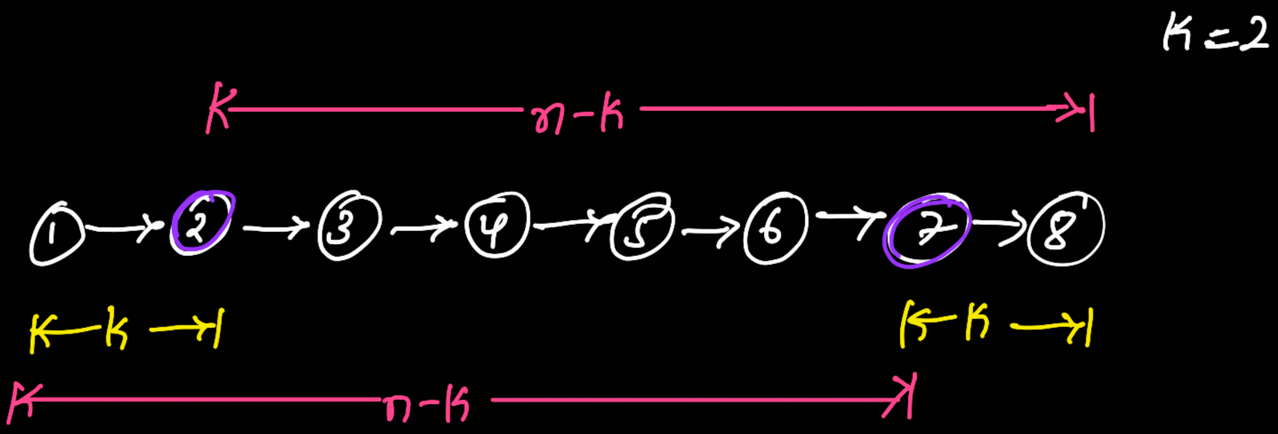
now swap the values.

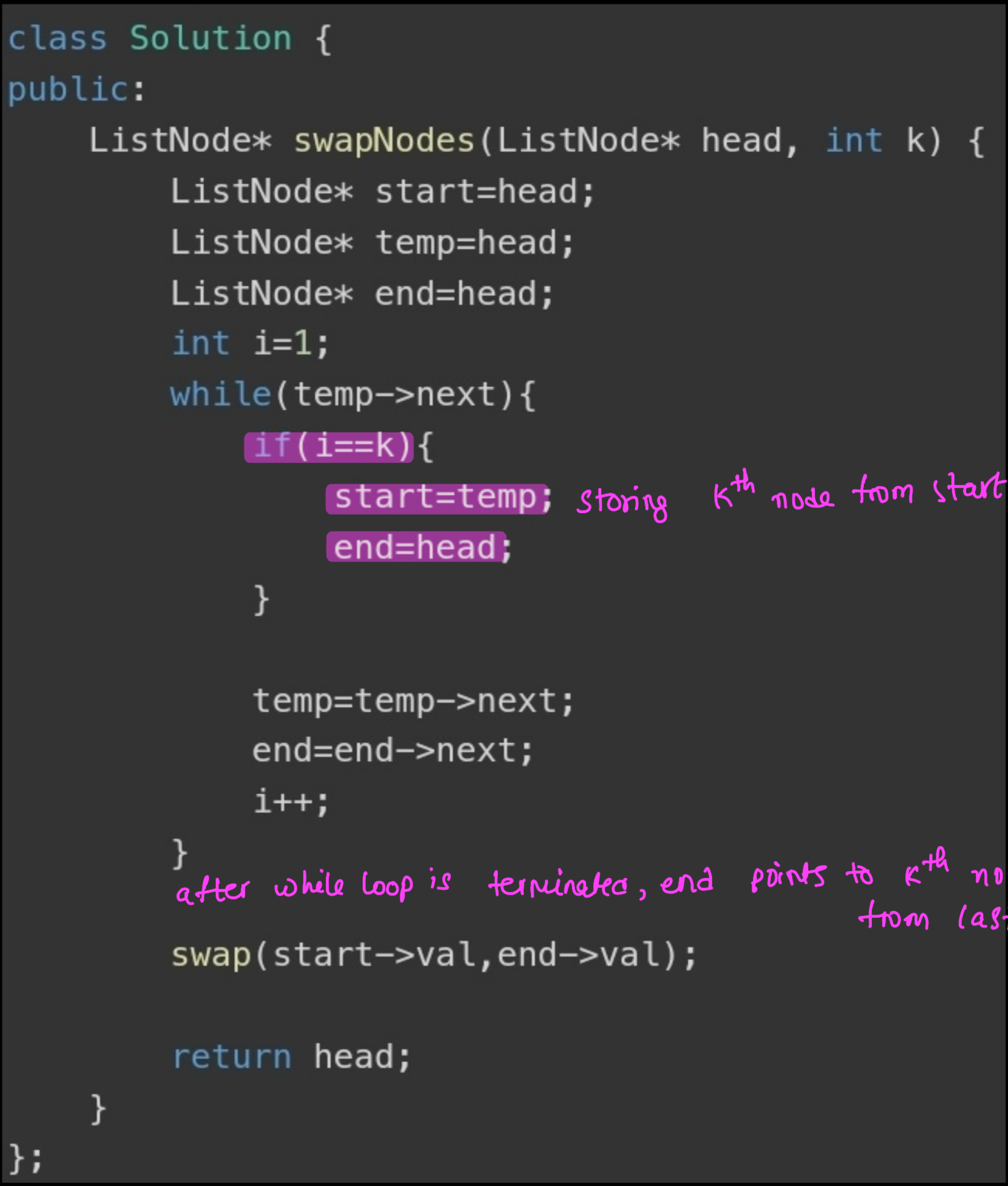
In 2^{nd} pass, we make k hops and mark the current node and then after making n-k+1 hops total we mark the current node.

T.C. : O(n) + O(n)

$T(n) : O(n) + O(n)$
$S(n) : O(1)$

Approach 2: One pass





```
class Solution {
public:
    ListNode* swapNodes(ListNode* head, int k) {
        ListNode* start=head;
        ListNode* temp=head;
        ListNode* end=head;
        int i=1;
        while(temp->next){
            if(i==k){
                start=temp;  storing kth node from start
                end=head;
            }

            temp=temp->next;
            end=end->next;
            i++;
        }
        after while loop is terminated, end points to kth node
                                                     from last
        swap(start->val,end->val);

        return head;
    }
};
```

$T(n) : O(n)$
$S(n) : O(1)$

note: if we are asked to swap nodes but not values, we can still do that by landing on nodes before the actual nodes.

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