Rotate List

Problem Description

Given the head of a linked list, rotate the list to the right by k places.

Example

Example 1:

Input:

head = [1,2,3,4,5], k = 2

Output:

[4,5,1,2,3]

Example 2:

Input:

head = [0,1,2], k = 4

Output:

[2,0,1]

Constraints

- The number of nodes in the list is in the range [0, 500].
- Node values are in the range [-100, 100].
- k is in the range [0, 2 * 10^9].

Solution Approach

1. Check for base cases: If the linked list is empty, has only one node, or if k is 0, return the list as is.
2. Calculate the length of the linked list.
3. Find the actual rotation amount (k mod length of the list).
4. If the new rotation amount is 0, return the list as is.
5. Traverse to find the new tail and the node just before it.
6. Update pointers to perform rotation.