Reverse Linked List II

For this problem the task is to reverse a portion of the linked list between positions left and right (1-indexed).

Key Idea (One Pass Solution):

We can reverse only the sublist between left and right while keeping the rest of the list intact.

Steps:

1. Use a dummy node to handle edge cases where left = 1 (reversal starts at head).
2. Traverse to the node just before left (prev pointer).
3. Reverse nodes from left to right using standard in-place reversal.
4. Reconnect the reversed sublist back to the main list.

Visualization:

For head = [1, 2, 3, 4, 5], left = 2, right = 4.

Before reversal: 1 → 2 → 3 → 4 → 5

Reverse sublist 2-4: 1 → 4 → 3 → 2 → 5

Complexity:

- Time Complexity: O(n) (single traversal)
- Space Complexity: O(1) (in-place reversal)