N-th node

To remove the n-th node from the end of a singly linked list in one pass, the most efficient approach is the two-pointer method (also called "fast and slow pointers").

Key Idea: 1. Use two pointers: fast and slow.
2. Move the fast pointer n steps ahead first.
3. Then move both fast and slow together until fast reaches the end.
4. At the point, slow is right before the node to be removed.
5. Modify slow → next to skip the target node.

Steps: 1. Use a dummy node before head to handle edge cases (e.g., removing the first node).
2. Initialise both fast and slow to dummy.
3. Move fast n+1 steps forward (to create a gap of n between fast and slow.
4. Move both pointers until fast hits nullptr.
5. Do slow → next = slow → next → next; to remove the node

Example: For head = [1, 2, 3, 4, 5], n = 2:
- After gap: fast at 3 slow at 1
- After traversal: fast at end, slow at 3
- Remove 4 (slow → next)

Time Complexity: · O(sz) one traversal
· O(1) space