Middle of the Linked List LeetCode

Approach 1: This is the brute force approach to find the middle of the linked list.

This approach involves traversing the linked list to find the middle node. It calculates the middle index by dividing the total length of the list by 2. Then, it iterates through the list until it reaches the node at the middle index.

- **Time Complexity: O(N), where N is the number of nodes in the linked list. In the worst case, it needs to traverse the entire list to find the middle.**
- **Space Complexity: O(1), as it uses a constant amount of extra space for variables.**

Approach 2: Optimized way to find the middle using the two-pointer approach.

This approach is more optimized and uses two pointers, a slow pointer and a fast pointer. The fast pointer moves twice as fast as the slow pointer. When the fast pointer reaches the end of the list, the slow pointer will be at the middle node.

- **Time Complexity: O(N), where N is the number of nodes in the linked list. Both pointers need to traverse the entire list, but the fast pointer moves at twice the speed.**
- **Space Complexity: O(1), as it uses a constant amount of extra space for the pointers.**