Find The Sum Of Last N Nodes Of The Linked List

⊙ solved by	`Senan`
⊙ Platform	`GeeksForGeeks`
↔ difficulty	`Easy`
≔ tags	`Linked List`
🗣 language	`C++`
📅 solved on	@23/10/2024
🔗 link	https://www.geeksforgeeks.org/problems/find-the-sum-of-last-n-nodes-of-the-linked-list/1
☑ Completion	✅

Intuition

To find the sum of the last `n` nodes in a singly linked list, the idea is to use two pointers, `slow` and `fast`. The `fast` pointer is advanced by `n` steps, then both pointers are moved together one step at a time until the `fast` pointer reaches the end. At that point, the `slow` pointer will be at the beginning of the last `n` nodes. We can then sum the values from the `slow` pointer onwards.

Approach

1. **Two Pointers (Fast and Slow)**:
   - Start both pointers at the head of the linked list.
   - Move the `fast` pointer `n` steps ahead.
   - Then move both `slow` and `fast` one step at a time until the `fast` pointer reaches the end of the list.
   - Now, the `slow` pointer points to the start of the last `n` nodes.
2. **Summing the Last `n` Nodes**:
   - Once the `slow` pointer is positioned at the start of the last `n` nodes, traverse the list from there, summing the values of the nodes until the end of the list.

Complexity

Time Complexity:

- **O(L)** where `L` is the total number of nodes in the list. We traverse the list once to position the `fast` pointer and again to sum the last `n` nodes. This is a linear pass through the list.

Space Complexity:

- **O(1)** because we only use a few extra pointers for traversal, so the space complexity is constant.

Code

```cpp
class Solution {
  public:
    int sumOfLastN_Nodes(struct Node* head, int n) {
        Node* slow = head;
        Node* fast = head;
        int sum = 0;

        while(n > 0 && fast != nullptr){
            fast = fast->next;
            n--;
        }
        while(fast != nullptr){
            slow = slow->next;
            fast = fast->next;
        }

        while(slow != nullptr){
            sum += slow->data;
            slow = slow->next;
        }

        return sum;
    }
};
```