Sort Linked List of 0s, 1s & 2s CodeStudio

You are given a singly linked list containing values that are either 0, 1, or 2. Write a C++ program that sorts the linked list in ascending order.

Example:

1 -> 0 -> 2 -> 2 -> 0 -> 0 -> 1 -> 0 -> 2 -> NULL

Output:

0 -> 0 -> 0 -> 0 -> 1 -> 1 -> 2 -> 2 -> 2 -> NULL

Approach 1: Sort Linked List of 0s, 1s & 2s using Counting approach.

- Traverse the linked list and count the occurrences of 0s, 1s, and 2s.
- Modify the linked list while traversing again:
    - Set the current node's value to 0 while decrementing the count of 0s.
    - Set the current node's value to 1 while decrementing the count of 1s.
    - Set the current node's value to 2 while decrementing the count of 2s.
- **Time Complexity: O(n), where n is the number of nodes in the linked list.**
- **Space Complexity: O(1), as only a constant amount of extra space is used.**

Approach 2: Sort Linked List of 0s, 1s & 2s using Partition approach

- Create separate lists for 0s, 1s, and 2s.
- Traverse the original list and partition elements into these three lists based on their values.
- Reconnect the three lists in the sorted order: 0s, then 1s, and finally 2s.
- **Time Complexity: O(n), where n is the number of nodes in the linked list.**
- **Space Complexity: O(1), as only a constant amount of extra space is used.**