DAY 12:

Task 7: Merging Two Sorted Linked Lists

You are provided with the heads of two sorted linked lists. The lists are sorted in ascending order. Create a merged linked list in ascending order from the two input lists without using any extra space (i.e., do not create any new nodes).

EXPLANATION:

Steps Explained

1. Initialization:

- Create a dummy node that serves as the head of the merged list.
- Initialize a current pointer to keep track of the last node added to the merged list.

2. Merge Process:

- While both input lists are not empty:
  - Compare the values of the current nodes of the two lists.
  - Append the node with the smaller value to the merged list.
  - Move the corresponding pointer to the next node in that list.
  - Update the current pointer to the node that was just added to the merged list.

3. Handling Remaining Nodes:

- After one of the input lists becomes empty, append the remaining nodes of the non-empty list to the merged list.

4. Return:

- Return the next node of the dummy node, which represents the head of the merged list.

Example Workflow

Consider two sorted linked lists:

- List 1: 8 -> 16 -> 12
- List 2: 5 -> 25 -> 24

1. Initialization:

- Create a dummy node with a value of -1.

- Set the current pointer to the dummy node.

2. Merge Process:

- Compare the values: 5 < 8. Append 5 to the merged list.

- Move pointer of List 2 to 25.

- Compare the values: 8 < 25. Append 8 to the merged list.

- Move pointer of List 1 to 16.

- Compare the values: 16 < 25. Append 16 to the merged list.

- Move pointer of List 1 to 12.

- Compare the values: 12 < 25. Append 12 to the merged list.

- Move pointer of List 2 to 24.

- Compare the values: 24 < null. Append 24 to the merged list.

3. Return:

- Return the next node of the dummy node, which is the head of the merged list (5 -> 8 -> 12 -> 16 -> 24 -> 25).

Conclusion

The provided algorithm efficiently merges two sorted linked lists into a single sorted linked list by comparing the values of nodes and appropriately linking them together. This approach ensures that the merged list maintains the sorted order while combining the elements from the input lists.