Insertion Sort for Singly Linked List

Problem Statement

Given the head of a singly linked list, sort the list using the insertion sort algorithm and return the head of the sorted list.

Definition

Singly Linked List: A data structure where each node contains a value and a reference (pointer) to the next node in the list.

Insertion Sort Algorithm

Insertion sort works by building a sorted portion of the list one element at a time. It follows these steps:

1. **Initialization:**
    - A dummy node is created to serve as the starting point of the sorted list.

2. **Iteration:**
    - Iterate through each node in the original list.
    - For each node, remove it from the original list and find its correct position in the sorted portion of the list.
    - Insert the node into the correct position within the sorted list.

3. **Completion:**
    - Repeat the process until all nodes from the original list have been inserted into the sorted list.
    - Return the head of the sorted list, which starts after the dummy node.

Example 1

- **Input:** [4, 2, 1, 3]
- **Output:** [1, 2, 3, 4]

Example 2

- **Input:** [-1, 5, 3, 4, 0]
- **Output:** [-1, 0, 3, 4, 5]

Constraints

- The number of nodes in the list is between 1 and 5000.
- Node values range from -5000 to 5000.

Detailed Steps

1. **Create a Dummy Node:**
    - A dummy node is initialized with a value of negative infinity (-∞). This dummy node helps manage the sorted portion of the list.

2. **Process Each Node:**
    - Initialize current to the head of the original list.
    - *For each node pointed to by current:*
        - *Save the Next Node:* Store the reference to the next node.
        - *Find Insertion Point:* Traverse the sorted portion of the list to find where the current node should be inserted.
        - *Insert Node:* Place the current node into its correct position within the sorted portion.
        - *Move to Next Node:* Update current to the saved next node.

3. **Return the Sorted List:**
    - After processing all nodes, return the list starting from dummy.next, which represents the head of the sorted list.

Code Explanation

Class Definition:

- *ListNode:* A class representing a node in a singly linked list, with attributes val (value) and next (pointer to the next node).

Solution Class:

- *insertionSortList:* A method that takes the head of the list, applies insertion sort, and returns the head of the sorted list.

Key Points

- *Efficiency:* Insertion sort is efficient for small or partially sorted lists but has a time complexity of $O(n^2)$ in the average and worst cases.

- *Space Complexity:* The algorithm sorts the list in-place, so the space complexity is $O(1)$.

This approach ensures that the linked list is sorted efficiently using the properties of the insertion sort algorithm, and the sorting is done in-place, minimizing additional space usage.