

50. Insertion Sort List Given the head of a singly linked list, sort the list using insertion sort, and return the sorted list's head. The steps of the insertion sort algorithm: 1. Insertion sort iterates, consuming one input element each repetition and growing a sorted output list. 2. At each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list and inserts it there. 3. It repeats until no input elements remain. The following is a graphical example of the insertion sort algorithm. The partially sorted list (black) initially contains only the first element in the list. One element (red) is removed from the input data and inserted in-place into the sorted list with each iteration.

Program:

```
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next

def insertionSortList(head):
    # Create a dummy node to act as the starting point of the sorted list
    dummy = ListNode(0)
    current = head

    while current:
        # At each iteration, current points to the node to be inserted
        prev_node = dummy
        next_node = dummy.next

        # Find the correct position to insert the current node
        while next_node:
            if current.val < next_node.val:
                break
            prev_node = next_node
            next_node = next_node.next

        # Insert current node in the sorted list
        next_iter = current.next
        current.next = next_node
        prev_node.next = current

        # Move to the next element in the original list
        current = next_iter

    return dummy.next

# Helper function to print the linked list
def printList(head):
    while head:
        print(head.val, end=" -> ")
        head = head.next
    print("None")
```

```
# Example usage
# Creating a linked list: 4 -> 2 -> 1 -> 3
head = ListNode(4)
head.next = ListNode(2)
head.next.next = ListNode(1)
head.next.next.next = ListNode(3)
```

```
print("Original List:")
printList(head)
```

```
sorted_head = insertionSortList(head)
```

```
print("Sorted List:")
printList(sorted_head)
```

Output:

```
"C:\Program Files\Python312\python.exe" "C:\Work Space\DAA COADS.PYTHON\program 50.py"
Original List:
4 -> 2 -> 1 -> 3 -> None
Sorted List:
1 -> 2 -> 3 -> 4 -> None

Process finished with exit code 0
```

TIME COMPLEXITY:

$F(n)=O(n \log n)$