Problem 147: Insertion Sort List

Problem Information

Difficulty: Medium
Acceptance Rate: 57.69%
Paid Only: No
Tags: Linked List, Sorting

Problem Description

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.

![](https://upload.wikimedia.org/wikipedia/commons/0/0f/Insertion-sort- example-300px.gif)

Example 1:

![](https://assets.leetcode.com/uploads/2021/03/04/sort1linked-list.jpg)

Input: head = [4,2,1,3] **Output:** [1,2,3,4]

Example 2:

![](https://assets.leetcode.com/uploads/2021/03/04/sort2linked-list.jpg)

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

Constraints:

* The number of nodes in the list is in the range `[1, 5000]`. * `-5000 <= Node.val <= 5000`

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
ListNode* insertionSortList(ListNode* head) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public ListNode insertionSortList(ListNode head) {
}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def insertionSortList(self, head: Optional[ListNode]) -> Optional[ListNode]:
```