Problem 148: Sort List

Problem Information

Difficulty: Medium
Acceptance Rate: 63.22%
Paid Only: No
Tags: Linked List, Two Pointers, Divide and Conquer, Sorting, Merge Sort

Problem Description

Given the `head` of a linked list, return _the list after sorting it in**ascending order**_.

Example 1:

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

Example 2:

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

Example 3:

Input: head = [] **Output:** []

Constraints:

* The number of nodes in the list is in the range `[0, 5 * 104]`. * `-105 <= Node.val <= 105`

Follow up: Can you sort the linked list in `O(n logn)` time and `O(1)` memory (i.e. constant space)?

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* sortList(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 sortList(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 sortList(self, head: Optional[ListNode]) -> Optional[ListNode]:
```