

## 148. Sort List

```
class Solution {  
public:  
    ListNode* merge(ListNode* l1, ListNode* l2) {  
        ListNode dummy(0);  
        ListNode* tail = &dummy;  
  
        while (l1 && l2) {  
            if (l1->val < l2->val) {  
                tail->next = l1;  
                l1 = l1->next;  
            } else {  
                tail->next = l2;  
                l2 = l2->next;  
            }  
            tail = tail->next;  
        }  
        tail->next = l1 ? l1 : l2;  
        return dummy.next;  
    }  
  
    ListNode* sortList(ListNode* head) {  
        if (!head || !head->next) return head;  
  
        ListNode* slow = head, *fast = head, *prev = nullptr;  
        while (fast && fast->next) {  
            prev = slow;
```

```

        slow = slow->next;

        fast = fast->next->next;

    }

    prev->next = nullptr;

    ListNode* left = sortList(head);

    ListNode* right = sortList(slow);

    return merge(left, right);

}

};

```

The screenshot displays a LeetCode submission for the "Sort List" problem. The submission is accepted, with a runtime of 9 ms (89.79% efficiency) and memory usage of 56.88 MB (93.05% efficiency). The C++ code defines a singly-linked list structure and implements a merge sort algorithm to sort the list. The test result shows the input [4,2,1,3] and the output [1,2,3,4].

**Runtime Graph:** The graph shows the execution time of the code across different input sizes. The x-axis represents time in milliseconds (ms), ranging from 0 to 75ms. The y-axis represents the percentage of test cases passed, ranging from 0% to 8%. The graph shows a peak in performance around 11ms, indicating that the code is efficient for most inputs.

**Code:** The code is written in C++ and defines a singly-linked list structure. It includes a `ListNode` struct with an `int` value and a `next` pointer. The `merge` function is used to merge two sorted lists. The `sortList` function implements a merge sort algorithm to sort the list.

**Test Result:** The test result shows the input [4,2,1,3] and the output [1,2,3,4]. The submission is accepted, indicating that the code correctly sorts the list.