**TASK REPORT**

**ALGORITHMS AND DATA STRUCTURE**

**Linked List Problems**

**Created by:**

**Onic Agustino**

**L200234275**

**X**

**Date : March 29, 2025**

**INFORMATICS ENGINEERING**

**FACULTY OF COMMUNICATION AND INFORMATICS**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2024/2025**

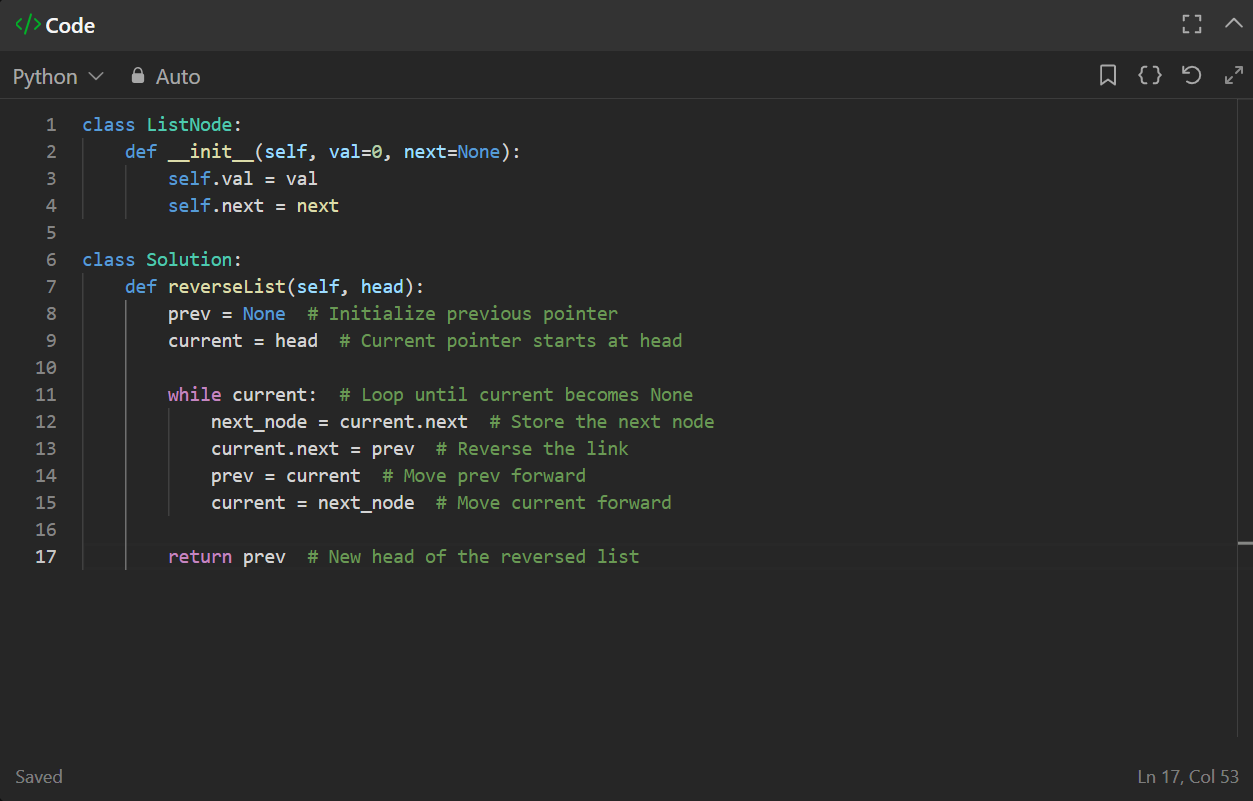
**Leetcode**

1. **Reserve Linked List**

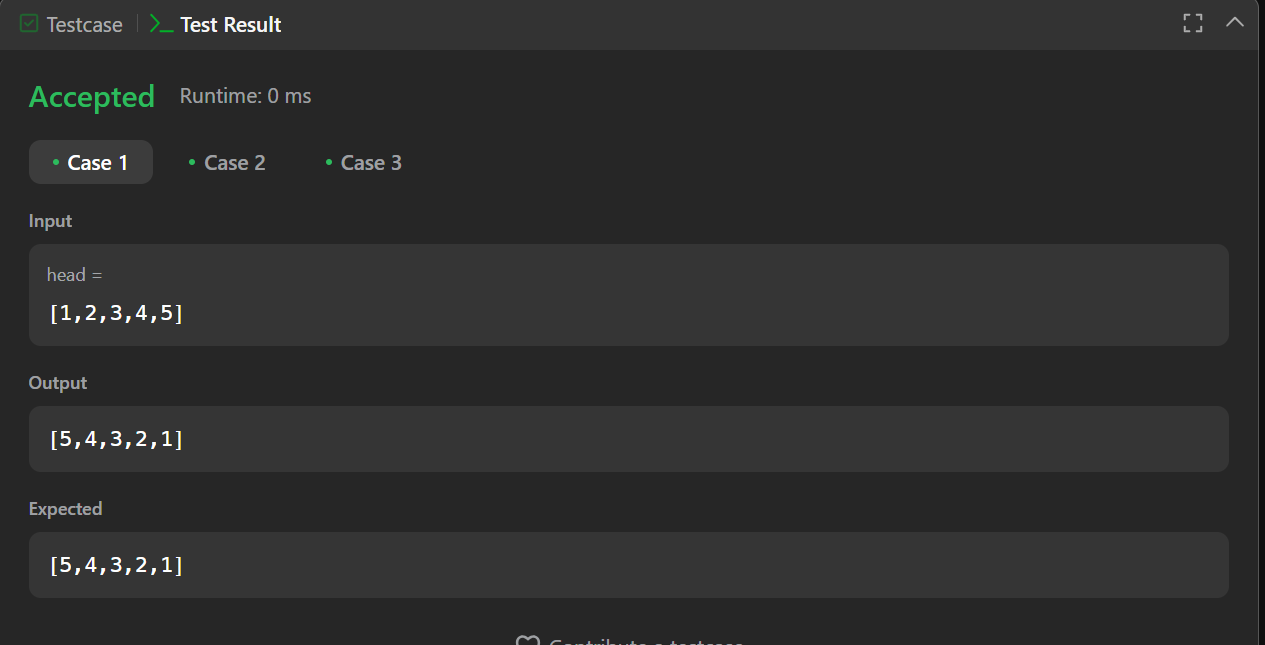
* **Problem Explanation**

Given the head of a singly linked list, reverse the list and return its new head. The input is a linked list, and the output is linked list with the order of elements reversed.

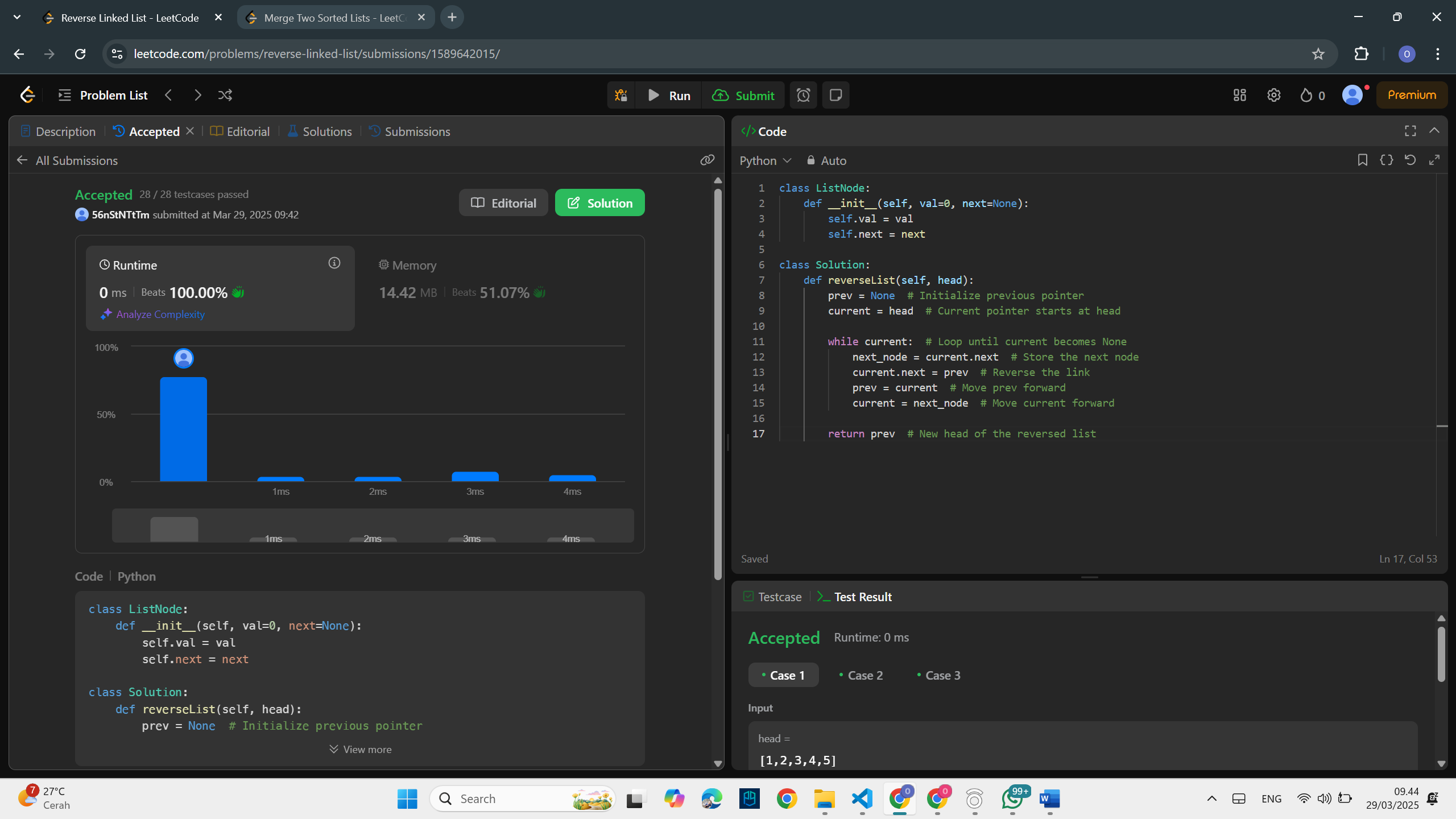
* **Solution Approach**
* We use iterative approach with two pointers : prev (stores the previous node) and curr (stores the current node).
* We traverse through the linked list while updating pointers to reverse the direction of each node.
* At the end of the traversal, prev will be the new head of the reversed list.
* **Code Implementation**

****

* **Output**

****

* **Screenshot Submission**

****

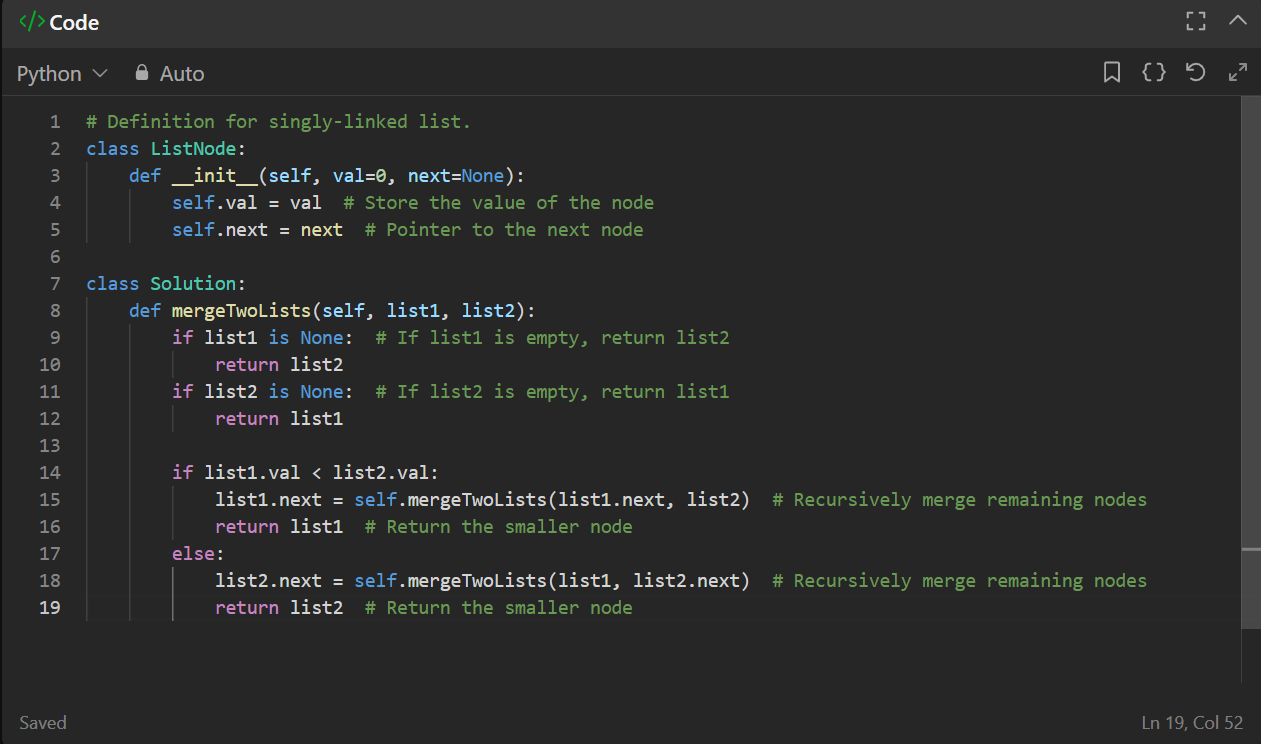
\

1. **Merge Two Sortes Lists**

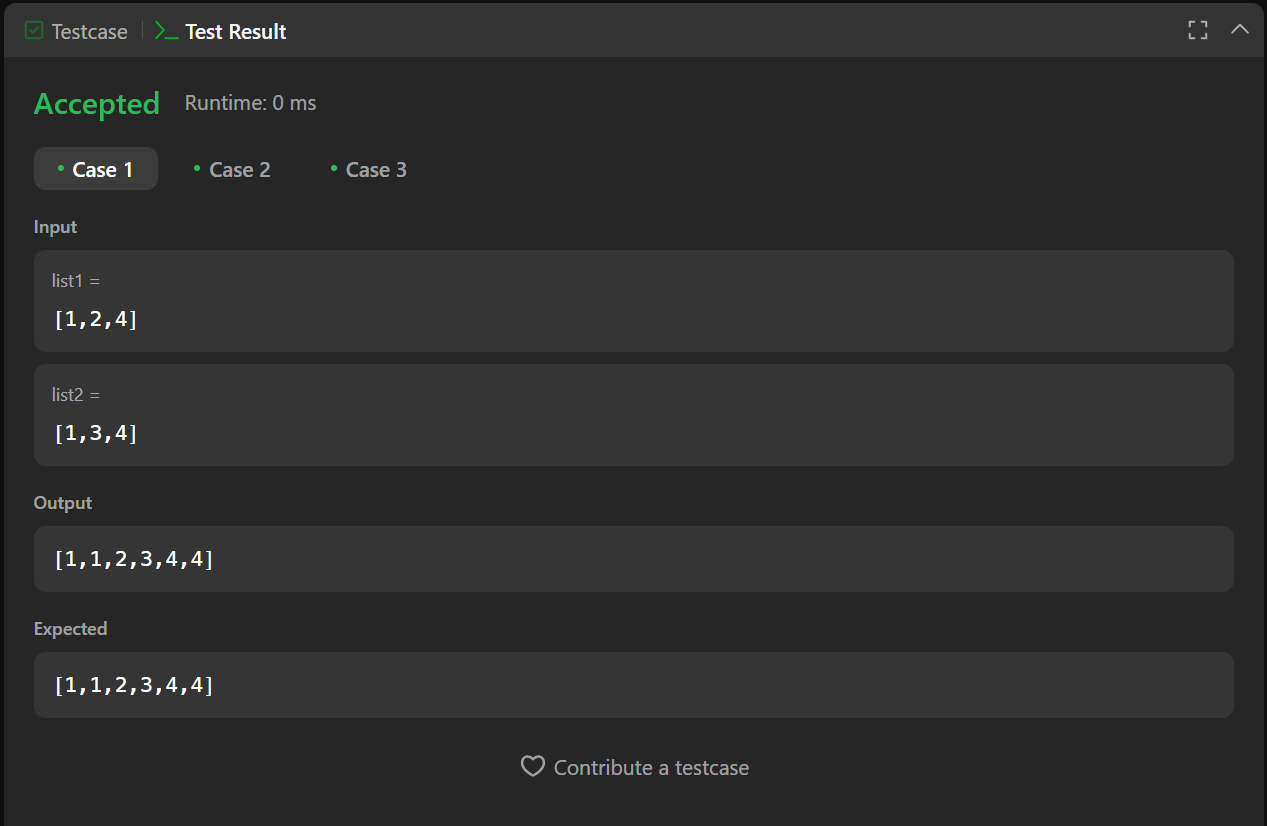
* **Problem Explanation**

Given the heads of two sorted linked list, merge them into one sorted linked list.

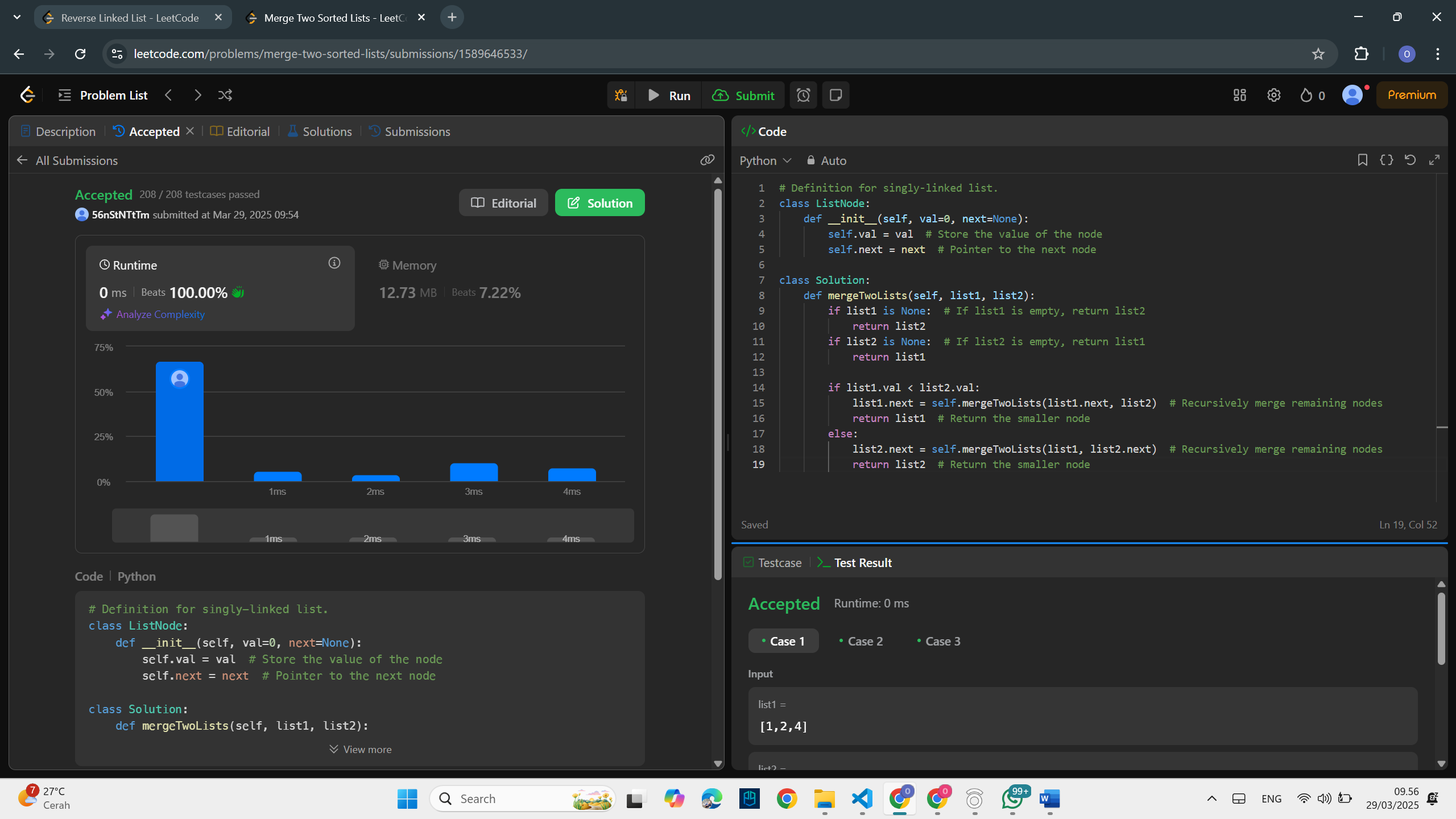
* **Solution Approach**
* We compare the values of nodes from both lists and attach the smaller one to the merged list
* This is done recursively by choosing the smaller node and calling the function for the remaining nodes.
* **Code Implementation**



* **Output**



* **Screenshot Submission**

****