

**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. Reverse 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**

```
</> Code
Python ▾ Auto
1 class ListNode:
2     def __init__(self, val=0, next=None):
3         self.val = val
4         self.next = next
5
6 class Solution:
7     def reverseList(self, head):
8         prev = None # Initialize previous pointer
9         current = head # Current pointer starts at head
10
11         while current: # Loop until current becomes None
12             next_node = current.next # Store the next node
13             current.next = prev # Reverse the link
14             prev = current # Move prev forward
15             current = next_node # Move current forward
16
17         return prev # New head of the reversed list
```

Saved Ln 17, Col 53

- **Output**

Testcase

Test Result

Accepted Runtime: 0 ms

• Case 1

• Case 2

• Case 3

Input

head =  
[1,2,3,4,5]

Output

[5,4,3,2,1]

Expected

[5,4,3,2,1]

- **Screenshot Submission**

Reverse Linked List - LeetCode Merge Two Sorted Lists - LeetCode

leetcode.com/problems/reverse-linked-list/submissions/1589642015/

Problem List

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 28 / 28 testcases passed  
56nSNTt1m submitted at Mar 29, 2025 09:42

Editorial Solution

Runtime 0 ms Beat: 100.00% Memory 14.42 MB Beat: 51.07%

Analyze Complexity

100%  
0% 50% 100%  
1ms 2ms 3ms 4ms

Code Python

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

class Solution:
    def reverseList(self, head):
        prev = None # Initialize previous pointer
        current = head # Current pointer starts at head
        while current: # Loop until current becomes None
            next_node = current.next # Store the next node
            current.next = prev # Reverse the link
            prev = current # Move prev forward
            current = next_node # Move current forward
        return prev # New head of the reversed list
```

Testcase Test Result

Accepted Runtime: 0 ms

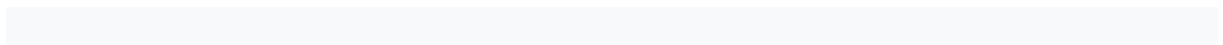
• Case 1

• Case 2

• Case 3

Input

head =  
[1,2,3,4,5]



\

## 2. Merge Two Sorted 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**

```
Code
Python Auto
1 # Definition for singly-linked list.
2 class ListNode:
3     def __init__(self, val=0, next=None):
4         self.val = val # Store the value of the node
5         self.next = next # Pointer to the next node
6
7 class Solution:
8     def mergeTwoLists(self, list1, list2):
9         if list1 is None: # If list1 is empty, return list2
10             return list2
11         if list2 is None: # If list2 is empty, return list1
12             return list1
13
14         if list1.val < list2.val:
15             list1.next = self.mergeTwoLists(list1.next, list2) # Recursively merge remaining nodes
16             return list1 # Return the smaller node
17         else:
18             list2.next = self.mergeTwoLists(list1, list2.next) # Recursively merge remaining nodes
19             return list2 # Return the smaller node
```

Saved Ln 19, Col 52

- **Output**

**Accepted** Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

list1 =  
[1, 2, 4]

list2 =  
[1, 3, 4]

Output

[1, 1, 2, 3, 4, 4]

Expected

[1, 1, 2, 3, 4, 4]

♥ Contribute a testcase

- **Screenshot Submission**

Reverse Linked List - LeetCode Merge Two Sorted Lists - LeetCode

leetcode.com/problems/merge-two-sorted-lists/submissions/1589646533/

Problem List > Run Submit

Description Accepted Editorial Solutions Submissions

All Submissions

Accepted 208 / 208 testcases passed  
56sINTITM submitted at Mar 29, 2025 09:54

Runtime 0 ms Beats 100.00% Memory 12.73 MB Beats 7.22%

Runtime

0 ms Beats 100.00%

Memory

12.73 MB Beats 7.22%

Code Python

```
# Definition for singly-linked list.
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val # Store the value of the node
        self.next = next # Pointer to the next node

class Solution:
    def mergeTwoLists(self, list1, list2):
```

Testcase > Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

list1 =  
[1, 2, 4]

list2 =  
[1, 3, 4]

Output

[1, 1, 2, 3, 4, 4]

Expected

[1, 1, 2, 3, 4, 4]

27°C Carah

Search

ENG 09:56 29/03/2025

