

## **75 Days of Code Day 31 206. Reverse Linked List (leetcode)**

**Type : Linked List**

**Medium**

**Given the head of a singly linked list, reverse the list, and return the reversed list.**

**Example 1:**

**Input: head = [1,2,3,4,5]**

**Output: [5,4,3,2,1]**

**Example 2:**

**Input: head = [1,2]**

**Output: [2,1]**

**Example 3:**

**Input: head = []**

**Output: []**

**Solution using array**

```
function reverseList(head: ListNode | null): ListNode | null {
    if(!head || !head.next){
        return head;
    }
    let arrayList = [];
    let temp :ListNode = head;
    while(temp){
        arrayList.push(temp.val);
        temp = temp.next;
    }

    let newReverseList = new ListNode(arrayList[arrayList.length-1]);
    let current = newReverseList;
    for(let index =arrayList.length-2; index>=0 ;index-- ){
        current.next = new ListNode(arrayList[index]);
        current = current.next;
    }

    return newReverseList;
};
```

✓ Accepted

Editorial

+ Solution

Runtime

Details

57 ms

Beats 82.62% of users with TypeScript

Memory

Details

44.96 MB

Beats 51.08% of users with TypeScript

This solution fails  $O(1)$  space complexity so now

**Solution in Optimize way**

Shubham Agrahari

```

34
35 function reverseList(head: ListNode | null): ListNode | null {
36     if(!head || !head.next){
37         return head;
38     }
39     let current :ListNode = head ;
40     let prev:ListNode | null = null ;
41     while(current){
42         let nextNode :ListNode = current.next;
43         current.next = prev;
44         prev = current;
45         current = nextNode;
46     }
47     return prev;
48
49
50 };

```

✓ Accepted

Editorial

+ Solution

Runtime

Details

**56** ms

Beats 85.48% of users with TypeScript

Memory

Details

**44.01** MB

Beats 98.25% of users with TypeScript