

## Interview Questions from Linked List

### 1. Remove Nth Node From End of List

#### Problem Statement:

Given the head of a linked list, remove the nth node from the end of the list and return its head.

#### Input Description:

- head: The head of a singly linked list.
- n: An integer representing the position from the end of the list.

#### Output Description:

- The head of the modified linked list.

#### Constraints:

- The number of nodes in the list is sz.
- $1 \leq sz \leq 30$
- $0 \leq \text{Node.val} \leq 100$
- $1 \leq n \leq sz$

#### Example 1:

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

Output: [1,2,3,5]

Explanation: The second node from the end is 4, so we remove it.

#### Example 2:

Input: head = [1], n = 1

Output: []

Explanation: The first node from the end is 1, so we remove it.

#### Example 3:

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

Output: [1]

Explanation: The second node from the end is 2, so we remove it.

### 2. Reverse Linked List

#### Problem Statement:

Reverse a singly linked list.

#### Input Description:

- head: The head of a singly linked list.

Output Description:

- The head of the reversed linked list.

Constraints:

- The number of nodes in the list is sz.
- $1 \leq sz \leq 5000$
- $-5000 \leq \text{Node.val} \leq 5000$

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: []

### 3. Merge Two Sorted Lists

Problem Statement:

Merge two sorted linked lists and return it as a sorted list. The list should be made by splicing together the nodes of the first two lists.

Input Description:

- list1: The head of the first sorted linked list.
- list2: The head of the second sorted linked list.

Output Description:

- The head of the merged sorted linked list.

Constraints:

- The number of nodes in both lists is sz.
- $0 \leq sz \leq 50$
- $-100 \leq \text{Node.val} \leq 100$
- Both list1 and list2 are sorted in non-decreasing order.

Example 1:

Input: list1 = [1,2,4], list2 = [1,3,4]

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

Example 2:

Input: list1 = [], list2 = []

Output: []

Example 3:

Input: list1 = [], list2 = [0]

Output: [0]

#### 4. Linked List Cycle

Problem Statement:

Given head, the head of a linked list, determine if the linked list has a cycle in it.

Input Description:

- head: The head of a singly linked list.

Output Description:

- true if there is a cycle in the linked list, otherwise false.

Constraints:

- The number of nodes in the list is sz.

-  $0 \leq sz \leq 10^4$

-  $-10^5 \leq \text{Node.val} \leq 10^5$

Example 1:

Input: head = [3,2,0,-4], pos = 1

Output: true

Explanation: There is a cycle in the linked list, where the tail connects to the 1st node (0-indexed).

Example 2:

Input: head = [1,2], pos = 0

Output: true

Explanation: There is a cycle in the linked list, where the tail connects to the 0th node.

Example 3:

Input: head = [1], pos = -1

Output: false

Explanation: There is no cycle in the linked list.

## 5. Add Two Numbers

### Problem Statement:

You are given two non-empty linked lists representing two non-negative integers. The digits are stored in reverse order, and each of their nodes contains a single digit. Add the two numbers and return the sum as a linked list.

### Input Description:

- l1: The head of the first linked list.
- l2: The head of the second linked list.

### Output Description:

- The head of the linked list representing the sum of the two numbers.

### Constraints:

- The number of nodes in each linked list is sz.
- $1 \leq sz \leq 100$
- $0 \leq \text{Node.val} \leq 9$
- It is guaranteed that the list represents a number that does not have leading zeros.

### Example 1:

Input: l1 = [2,4,3], l2 = [5,6,4]

Output: [7,0,8]

Explanation:  $342 + 465 = 807$ .

### Example 2:

Input: l1 = [0], l2 = [0]

Output: [0]

### Example 3:

Input: l1 = [9,9,9,9,9,9,9], l2 = [9,9,9,9]

Output: [8,9,9,9,0,0,0,1]