**2. Add Two Numbers**

<https://leetcode.com/problems/add-two-numbers/>

1. **Listen**

**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.

You may assume the two numbers do not contain any leading zero, except the number 0 itself.

**Input**:

**l1** is the head of a **linked list** representing a number

**l2** is the head of a **linked list** representing a number

**Goal**:

Add the two numbers in l1 and l2

**Return**:

return the **sum** of l1 and l2 as a singly linked list

1. **Example**

**Constraints:**

* The number of nodes in each linked list is in the range [1, 100].
* It is guaranteed that the list represents a number that does not have leading zeros (except for the number 0 itself).
* Digits are stored in reverse order in the list.
* Each node contains a single digit and cannot be negative (0-9 value). 0 <= Node.val <= 9
* Lists cannot be empty (head !- null).

**Test Cases:**

* l1 is longer than l2
* l1 is longer than l2 and has a carry
* l2 is longer than l1
* l2 is longer than l1 and has a carry
* l1 and l2 are same size
* l1 and l2 are same size but have a carry

**Edge Cases:**

* One number is 0 and the other isn’t
* Both numbers are 0
* Perform many carries

**Questions & Assumptions:**

**Example 1:** Same Length with No Carry

Think about solving this like you are solving a basic addition problem.

(1)

342

+ 465

---------

807

Notice how we have to account for carries.

Diagram

Description automatically generated

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

**Output:** [7,0,8]

**Explanation:** 342 + 465 = 807.

A piece of paper with writing

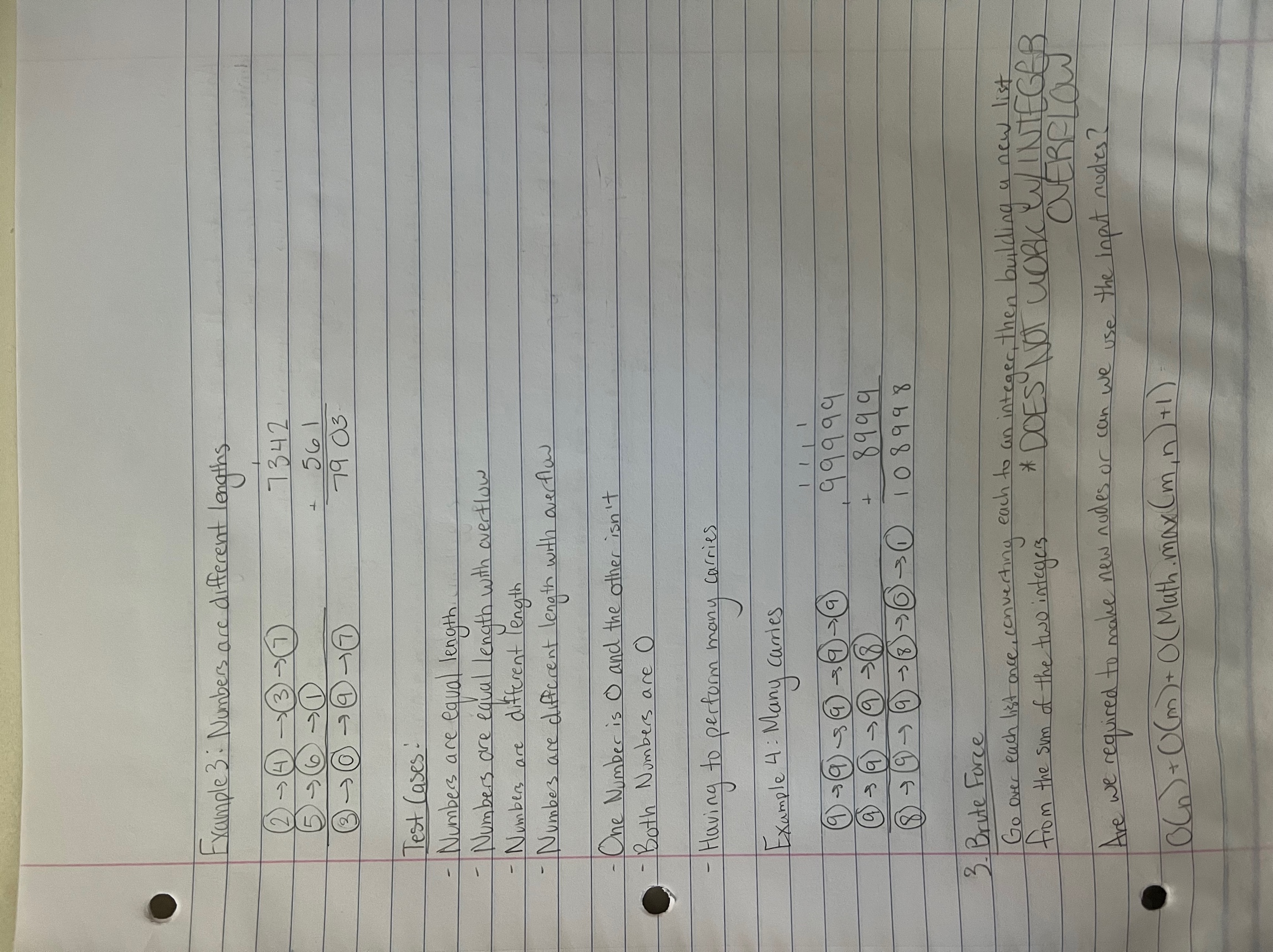
Description automatically generated with medium confidence

**A picture containing timeline

Description automatically generated**

We must account for carries that may carry across the longer number to the very end of the list as well.

**Example 4: Many Carries**

****

1. **Brute Force**

**Solution 1:**

Go over each list once, converting each to an integer, then build a new list from the sum of these two integers.

This would run in O(n) + O(m) + O(Math.max(m, n) + 1) runtime.

This solution, however, does not work because of **integer overflow**!

There can be up to 100 digits in a number which would look something like 1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000.

Realize that a signed integer’s value can only go up to 232 which is equivalent to 2,147,483,647 to -2,147,483,647.

Solution 2:

Add the numbers together in one pass,