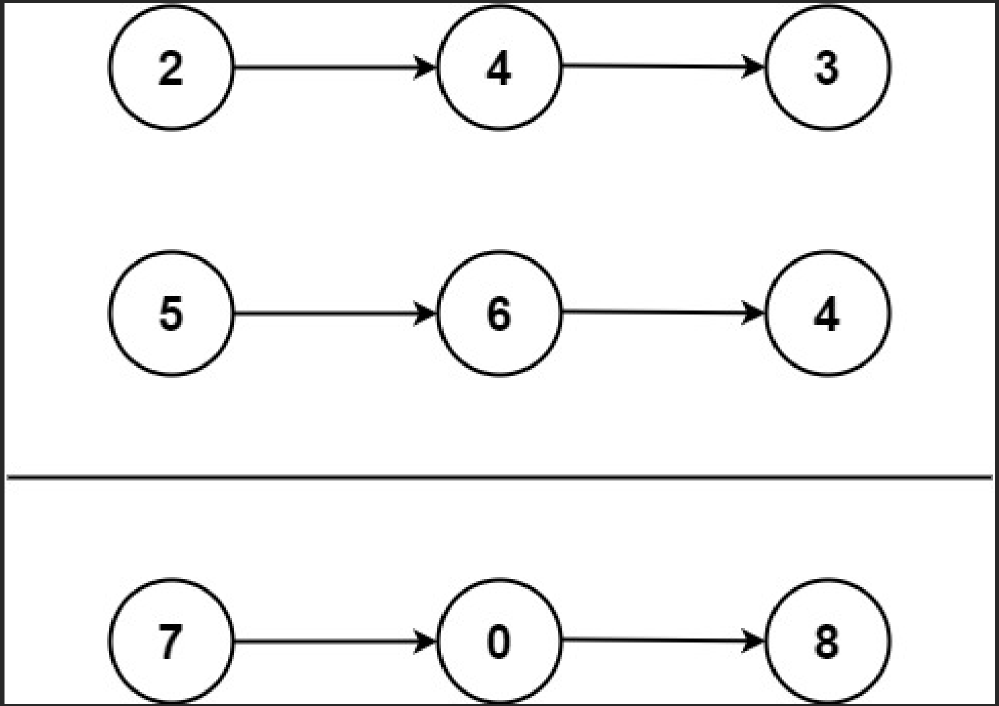
Add two number in Linked List

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.

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]
```

$$\begin{tikzcd} 2 \arrow[r] & 4 \arrow[r] & 7 \end{tikzcd}$$

$$\begin{tikzcd} 3 \arrow[r] & 5 \arrow[r] & 6 \end{tikzcd}$$

$$\begin{array}{r} 7\ 4\ 2 \\ +\ 6\ 5\ 2 \\ \hline 1\ 3\ 9\ 4 \end{array}$$

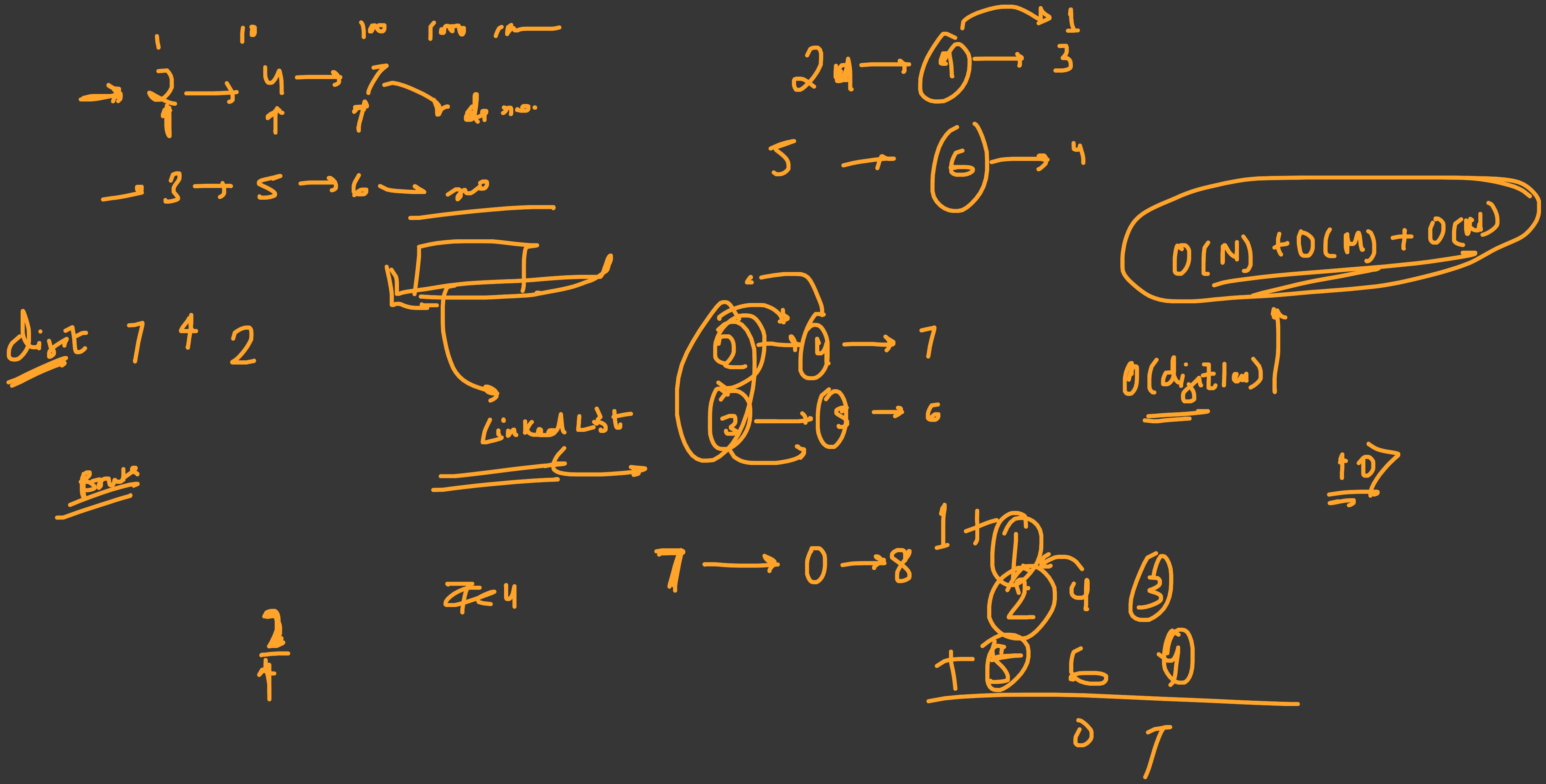
$\longleftrightarrow$

Output $4 \longrightarrow 9 \longrightarrow 3 \longrightarrow 1$

Constraint

$1 \longrightarrow 100 \longrightarrow$ Linked List range

$0 \longrightarrow 9 \longrightarrow$ Node val.



1 10 100 1000
2 → 4 → 7
3 → 5 → 6
digit 7 4 2
Linked List
2 4 → 3
5 → 6 → 4
7 → 0 → 8
1 + 1
2 4 3
+ 5 6 4
0 7
O(N) + O(M) + O(N)
O(digitlen)

• Idea → We maintain carry variable which store if any carry is there.

reverse / reverse

l_1 l_1 l_1

$$\begin{tikzcd} 2 \arrow[r] & 4 \arrow[r] & 7 \arrow[r] & X \end{tikzcd}$$

$$\begin{tikzcd} 3 \arrow[r] & 6 \arrow[r] & \boxed{1} \arrow[r] & 9 \arrow[r] & X \end{tikzcd}$$

l_2 l_2 l_2 ↑

sum 210 carry ~~1~~ 0

10 % 10 = 1

$\frac{10}{10} = 0$

$$\begin{tikzcd} 0 \arrow[r] & 5 \arrow[r] & 0 \arrow[r] & 9 \arrow[r] & 9 \arrow[r] & X \end{tikzcd}$$

5099

9905

26 742

$$\begin{array}{r} 9163 \\ \hline 9965 \end{array}$$

Pseudocode-:

```
Node* dummy = new Node(0), *temp = dummy;
  int carry = 0;
  while((l1 != NULL || l2 != NULL) || carry){
     int sum = 0;
     if(l1 != NULL){
        sum += l1 ->data;
        l1 = l1->next;
     }
     if(l2 != NULL){
        sum += l2->data;
        l2 = l2->next;
     }
     sum += carry;
     carry = sum/10;
     Node *sumNode = new Node(sum%10);
     temp ->next = sumNode;
     temp = temp->next;
  }
  Node *ans = dummy->next;
  delete dummy;
  return ans;
```

$\rightarrow$ S.C = O(length of ans = N)

T.C. = ~~O(N)~~

O(max(N, M))

S.C. = O(N)