

2020-01-18 - Handout – LinkedList Algorithms

Q1. Merge Two Sorted Lists

Link: <https://leetcode.com/problems/merge-two-sorted-lists/>

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

Input: 1->2->4, 1->3->4

Output: 1->1->2->3->4->4

Q2. Add two numbers

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

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 contain a single digit. Add the two numbers and return it as a linked list.

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

Example:

Input: (2 -> 4 -> 3) + (5 -> 6 -> 4)

Output: 7 -> 0 -> 8

Explanation: 342 + 465 = 807.

Q3. Copy List with Random Pointer

Link: <https://leetcode.com/problems/copy-list-with-random-pointer/>

A linked list is given such that each node contains an additional random pointer which could point to any node in the list or null.

Return a deep copy of the list.

The Linked List is represented in the input/output as a list of n nodes. Each node is represented as a pair of [val, random_index] where:

val: an integer representing Node.val

random_index: the index of the node (range from 0 to n-1) where random pointer points to, or null if it does not point to any node.

Input: head = [[7,null],[13,0],[11,4],[10,2],[1,0]]

Output: [[7,null],[13,0],[11,4],[10,2],[1,0]]