第一周的10个题目都是从Leetcode OJ 上直接找的，6个easy，4个medium。#后面是题目在leetcode上的编号。另，Intersection of Two Linked List 一题，正好是在Github上的报告模板中提到的一题，大家可以参考。

**1**

#206

**Problem: Reverse Linked List**

**Description:**

Reverse a singly linked list.

**2**

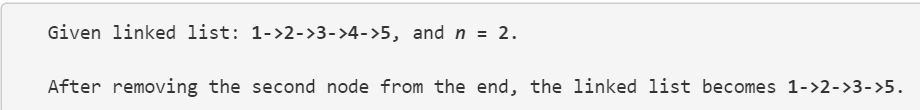
#19

**Problem: Remove Nth Node from End of List**

**Description:**

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

For example,



**Note:**  
Given *n* will always be valid.  
Try to do this in one pass.

**3**

#203

**Problem: Remove Linked List Elements**

**Description:**

Remove all elements from a linked list of integers that have value ***val***.

**Example**  
***Given:*** 1 --> 2 --> 6 --> 3 --> 4 --> 5 --> 6, ***val*** = 6  
***Return:*** 1 --> 2 --> 3 --> 4 --> 5

**4**

#83

### Problem: Remove Duplicates from Sorted List

**Description:**

Given a sorted linked list, delete all duplicates such that each element appear only *once*.

For example,  
Given 1->1->2, return 1->2.  
Given 1->1->2->3->3, return 1->2->3.

**5**

#21

**Problem: Merge Two Sorted Lists**

**Description:**

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

**6**

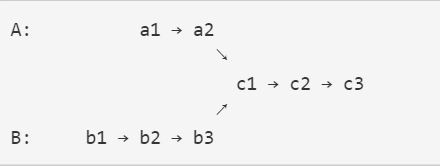
#160

### Problem: Intersection of Two Linked Lists

**Description:**

Write a program to find the node at which the intersection of two singly linked lists begins.

For example, the following two linked lists:



Begin to intersect at node c1.

**Notes:**

* If the two linked lists have no intersection at all, return null.
* The linked lists must retain their original structure after the function returns.
* You may assume there are no cycles anywhere in the entire linked structure.
* Your code should preferably run in O(n) time and use only O(1) memory.

**7**

#2

**Problem: Add Two Numbers**

**Description:** You are given two linked lists representing two non-negative numbers. 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.

**Input:** (2 -> 4 -> 3) + (5 -> 6 -> 4)  
**Output:** 7 -> 0 -> 8

### 8

#24

**Problem: Swap Nodes in Pairs**

**Description:**

Given a linked list, swap every two adjacent nodes and return its head.

For example,  
Given 1->2->3->4, you should return the list as 2->1->4->3.

Your algorithm should use only constant space. You may **not** modify the values in the list, only nodes itself can be changed.

**9**

#61

### Problem: Rotate List

### Description:

### Given a list, rotate the list to the right by *k* places, where *k* is non-negative.

For example:  
Given 1->2->3->4->5->NULL and *k* = 2,  
Return 4->5->1->2->3->NULL.

**10**

#148

**Problem: Sort List**

**Description:**

Sort a linked list in *O*(*n* log *n*) time using constant space complexity.