总的感觉是自己基础只是比较薄弱，有几道easy的题目之前看过稍微好一些，其他题目需要借鉴答案改来改去才算大概有个样子。（然后看看大家的report觉得自己其实很多边界问题没有考虑清楚）

我是条ddl狗。。。以后一定好好督促自己不迟交。。。

**1**

#206

**Problem: Reverse Linked List**

****

思路：

1因为链表是单向的，如果要反转链表的话我们需要调转节点的方向。

2这题的思路是用两个指针，previous指向前一个节点，current指向当前节点。

将当前（current）节点的下一个节点（next）指向当前（current）节点的前一个节点（previous）

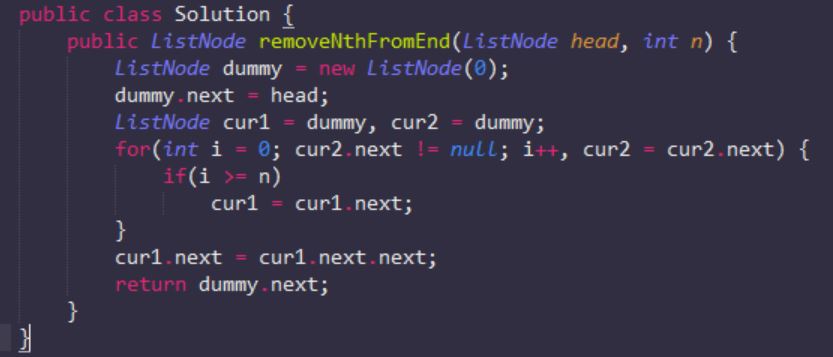
3 让前一个节点（previous）成为当前节点（current），让当前节点成为下一个节点（next）

4利用这两个指针遍历链表，完成翻转

**2**

#19

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

****

思路：

1 设置3个指针。Dummy指针的设置只是为了让dummy.next是head；（让遍历可以从头节点开始）

2另外两个指针cur1，cur2. cur2用来在for循环中计算链表长度，cur 1用来确定距离链表的第n+1个节点

3 删除距离链表第n个节点，只需要在这个节点的前一个节点 (n+1) 的next指针直接指向next.next，（即cur1.next=cur1.next.next）

**3**

#203

**Problem: Remove Linked List Elements**



思路：

和第二题思路基本一致

1 设置两个指针，dummy是为了让dummy.next=head

2 在遍历链表时，cur指针顺着链表方向移动，每次都比较cur.next的节点的数值和设定数值val是否相等

3 如果不相等，移动到cur.next,

如果相等，移动到cur.next.next（删除cur.next指向的节点）

**4**

#83

### Problem: Remove Duplicates from Sorted List

### 

思路：

因为是sorted list，那么出现duplicate的情况只能是前后节点value相同

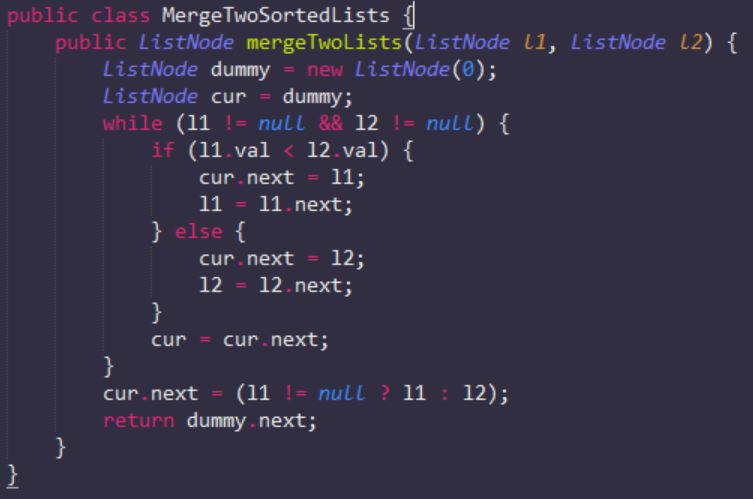
1 使用两个指针分别指向前后两个节点，遍历链表

2 每次比较两个节点数值是否相同，如果相同，则删除，如果不同，按照链表方向向前推移

**5**

#21

**Problem: Merge Two Sorted Lists**



思路：

1 设置dummy用来帮助确定头节点

2 使用current指针帮助确定当前已经merge的链表中的最后一个节点

3 使用l1,l2指针遍历两个链表，判断尚未merge的两个链表的当前节点值哪个更小，如果更小，则将这个节点放入已merge的链表的最后一个（cur=cur.next）

4 检查l1，l2是否有一个merge完另外一个还没merge完的情况

**6**

#160

### Problem: Intersection of Two Linked Lists

### 

思路：

1 常规思路，先用两个指针遍历两个链表，获得两个链表的长度；将长度作差diff，将较长的链表先移动diff长度，这样之后的操作中两个链表就是相同长度的了。之后两个指针同时移动，当节点数值相同(p1.val==p2.val)时，我们就找到了intersection

2这一题示例report中有一个十分巧妙的算法，将list a 遍历完之后接上list b头部遍历，将list b 遍历完之后接上list a头部遍历。这样可以巧妙解决list长度不一的问题。另，如果list a 长度为a，list b长度为b，如果他们只有一个相同node，那么这两个list公共长度为这个node的长度1,遍历次数应该是a+b次

**7**

#2

**Problem: Add Two Numbers**



思路：

1 用dummy帮助确定求和的新链表的第一位

2 利用l1，l2，以及一个预先设为0的carry针对每一位进行计算（不过这题我忘了考虑一些边界情况。比如list长度不一样/最高位进一等等。。。还需要再琢磨）

### 8

#24

**Problem: Swap Nodes in Pairs**

****

思路：这题的思路和2，3，4较为类似

1 使用dummy帮助确定利用头节点

2 利用prev和cur两个指针遍历链表，

3 在完成swap时，先互换当前节点和当前节点下一个节点的值，再将两个指针移动

**9**

#61

### Problem: Rotate List



思路：

1 利用指针遍历链表，然后将尾部节点和顶点相连成环

2 指针移动n-k%n后断开，这之中指针经过的节点就是rotate之后的链表

**10**

#148

**Problem: Sort List**

这题完全不会写，基础知识太薄弱。。。（还需要再学习相关知识点。。。）