1. 删除链表中的元素

解决：按顺序判断，如果当前位置值和下一个位置的值相等，就用下下个位置的值替换下一个位置。遇到相等的值时，当前位置指针不变，判断是否还会出现更多的重复值。

1. 链表求和

解决：对每次节点相加的值取余，保存取余之后的值，通过除以10来保存进位的值，每次存入值的时候就新建节点来保存数据。

1. 链表划分

解决：创建两个链表 分别保存小于x的结点和大于等于x的结点，连接两个链表

1. 在排序链表中插入一个节点

解决：构建一个当前排好序的链表，然后一个变量不断指向链表中的下一个节点。用变量c表示当前要插入的节点，每次循环找到c节点在当前排好序的链表中对应的位置，然后插入进去，然后指向原链表中下一个节点，继续进行插入过程，直到原链表的所有节点都完成。

1. 删除链表的倒数第N个节点

解决：先让 l1 向右移动 n - 1 次，若此时 l1->next 为空，说明我们要删除第一个元素。直接返回 head->next 即可。若 l1->next 不为空，l1 再向右移动一次，即共移动 n 次。

1. 链表插入排序

解决：用插入排序进行排序。注意空链表

1. 删除排序链表中的重复元素

解决：不断判断当前节点的下一个节点的val是否与当前节点的val相同，相同的话当前节点的next指针指向下一节点的下一个节点，否则当前节点后移一个节点。

1. 两两交换链表中的节点

解决：两两交换，后一个指向前一个，前一个指向 后面交换之后返回的头节点的值。

节点为null，或节点只有一个，则直接返回。

1. 合并两个排序链表

解决：将链表l1和l2合并后生成的新链表头节点设置为哑节点p，然后将l1和l2的节点比较大小，按照从小到大的顺序依次添加到哑节点p之后，如果l1的值val小于等于l2的val值，则将l1的val值追加到p之后，否则将l2的val值追加到p之和，从而构成新链表，最后合并链表尾部，如果链表l1不为空，则p继续指向l1，否则p指向链表l2剩余部分。

1. 链表倒数第n个节点

解决：快指针先走n步，然后快慢一起走，直到快指针走到最后

1. 翻转链表

解决：循环翻转链表，每次循环翻转一个结点。判断node是否是最后一个结点，如果是最后一个节点，则reverseHead指向node（确定翻转链表表头节点），然后node指向left（翻转链表），退出循环；如果不是最后一个节点，则node指向left（翻转链表），移动left和node指针。

1. 将二叉树按照层级转化为链表

解决：二叉树的层次遍历，每层的数据用链表存储

1. 合并区间

解决：先按照坐标轴，将区间排列。以区间里元素start升序排列。将重合的区域组合成新的区域

1. 中位数

解决：对于一组有有限个数的数据来说，它们的中位数是这样的一种数：这群数据里的一半的数据比它大，而另外一半数据比它小。 计算有限个数的数据的中位数的方法是：把所有的同类数据按照大小的顺序排列。 如果数据的个数是奇数，则中间那个数据就是这群数据的中位数。如果数据的个数是偶数，则中间那2个数据的算术平均值就是这群数据的中位数。

1. 搜索插入位置

解决：逐个比较数组元素与target值大小，若小于，则下一个，直到不小于，返回此时的索引值即可，下一个值是大于它还是等于他，target都要插入这里

1. 恢复旋转排序数组

解决：找到最小的那个数，然后把前面的数全部按顺序加到后面，最后再将前面的数删除

1. 删除排序数组中的重复数字

解决：c记录数组中不同数字的个数，每次出现不同数字时，增加c，

并且用这个值取作为下标覆盖掉原数组中该下标的值