****

软工暑期算法训练报告

专业班级： 11J192

学 号： 20191001576

姓 名： 李受渊

指导老师： 张剑波

**中国地质大学地理与信息工程学院**

**2020年 8 月**

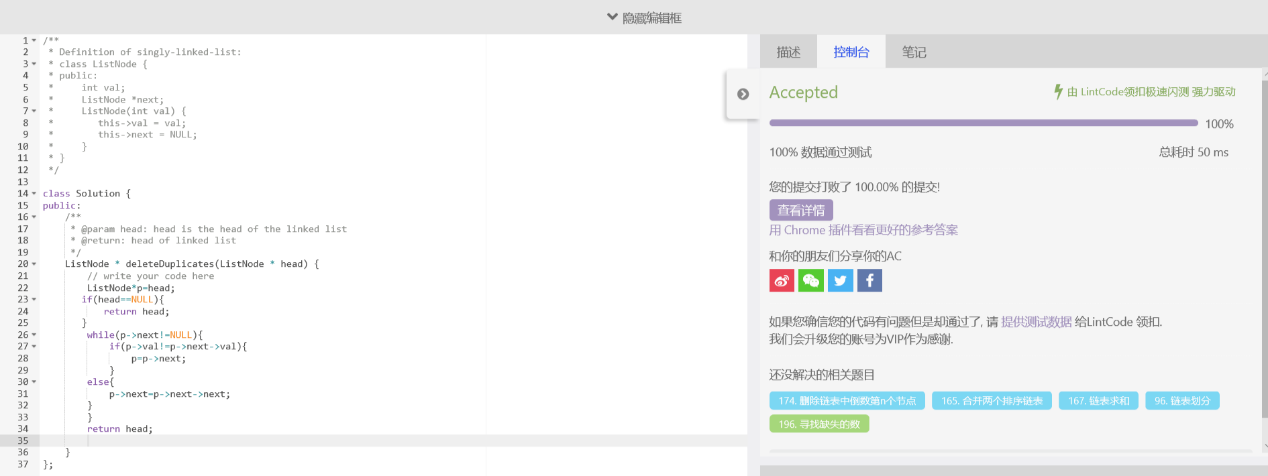
**第一部分：链表**

1. 翻转链表：



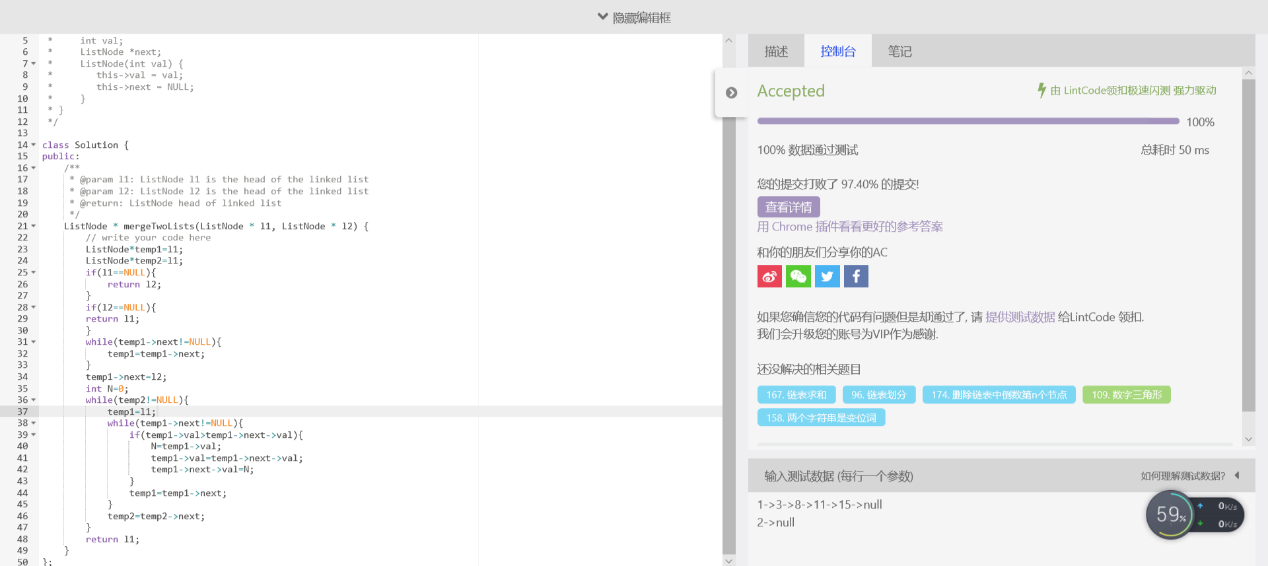
思路：创建temp1先保存了head的地址，将创建的temp2保存ｈｅａｄ－＞ｎｅｘｔ的地址，然后指回原来的地址，就可以实现链表的翻转。

1. 删除排序链表的重复元素：



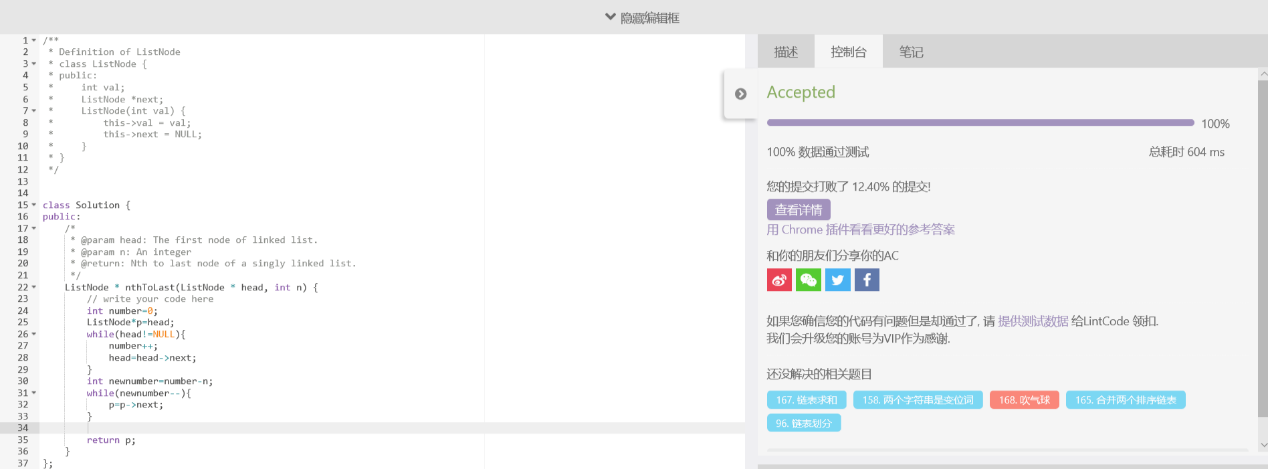
思路：先设立一个指针p指向head，然后去判断链表的每个节点的val值，如果这个值重复，则让p->next=p->next->next;

1. 合并两个排序链表



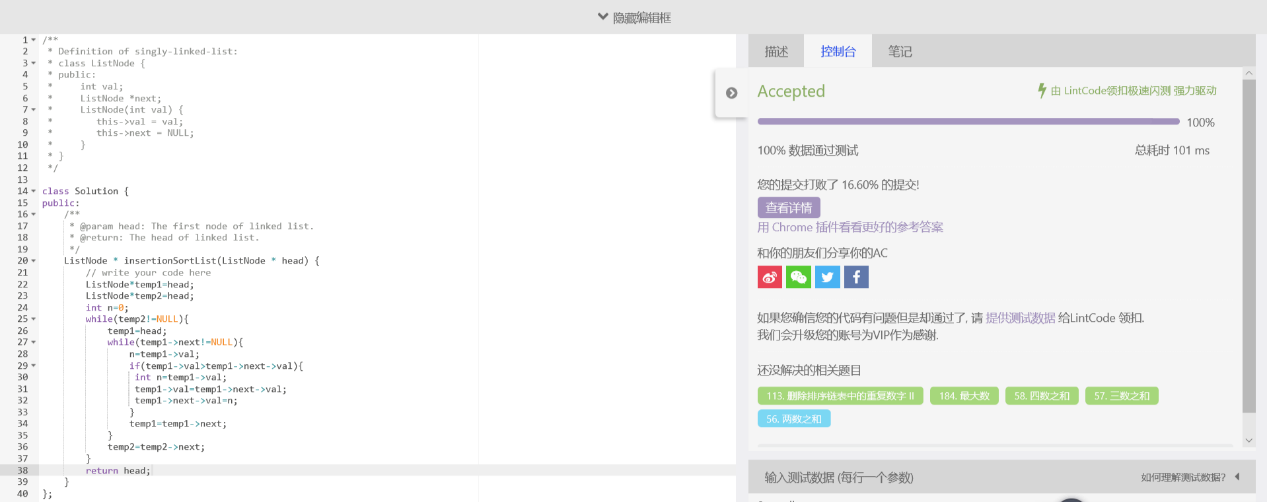
先考虑l1和l2中其中一个为NULL的情况，直接返回另一个，对于l1和l2都不为NULL的情况，先合并两个列表，当temp->next指向为NULL时，让他指向l2即可，接下来便是排序，模仿排序模板即可；

1. 链表的倒数第n个节点



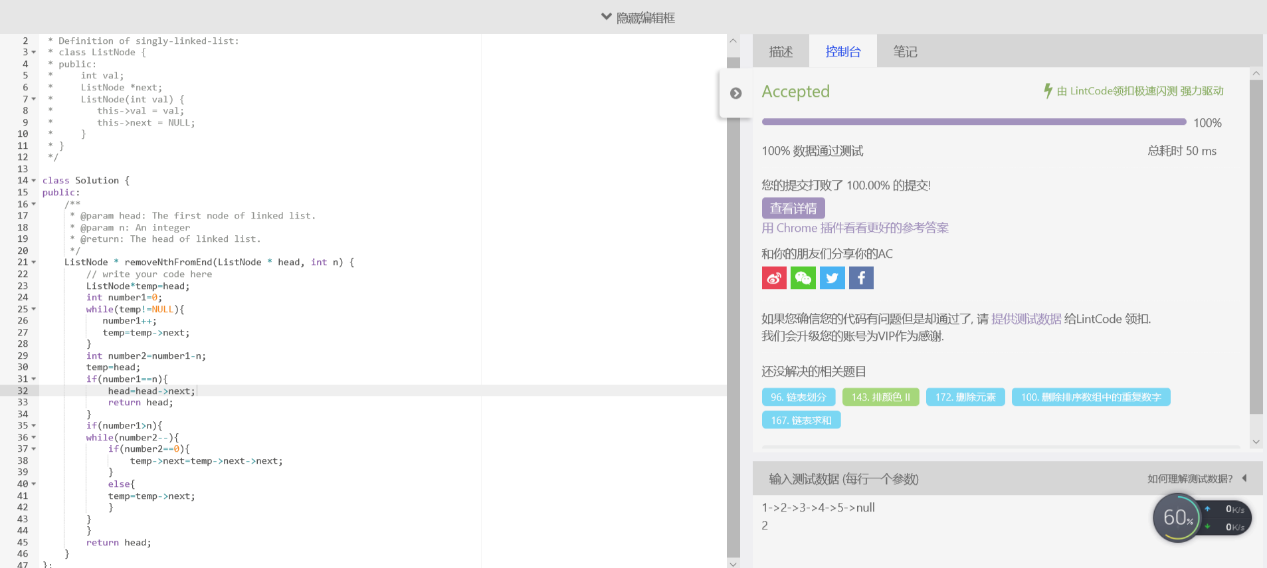
思路：先求出一共有多少个节点，再求出正过来数的位置，再一次遍历求出对应节点的值。

1. 链表插入排序



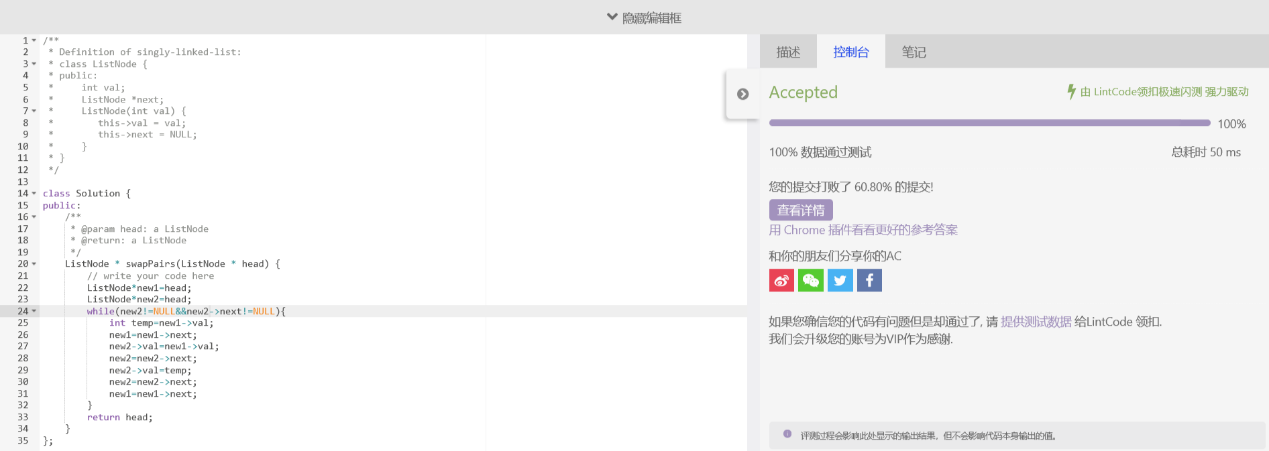
思路：排序只需要建立两个指针，然后模仿排序模板交换val即可。

1. 删除链表倒数第N个节点



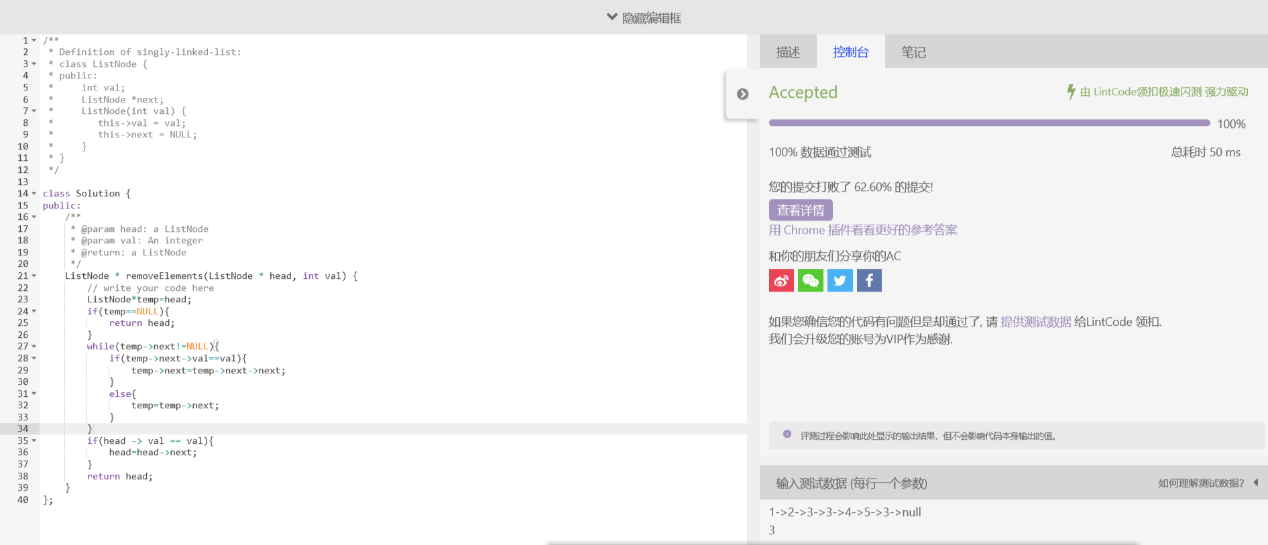
思路：先算出链表的节点数，再算出正数的位置，然后加上判断条件节点数大于等于n，然后到number2位置时删除节点即可；

1. 两两交换链表中的节点



思路：设立两个指针new1和new2指向head；当new2！=NULL且new->!=NULL时进入循环，在循环中设立一个整型的temp来存储刚开始的new1->val，然后让new1=new1->next,然后让把此刻的new1的val赋值给new2，此后再让new2->next,再另new2->val等于temp便完成了两两交换。

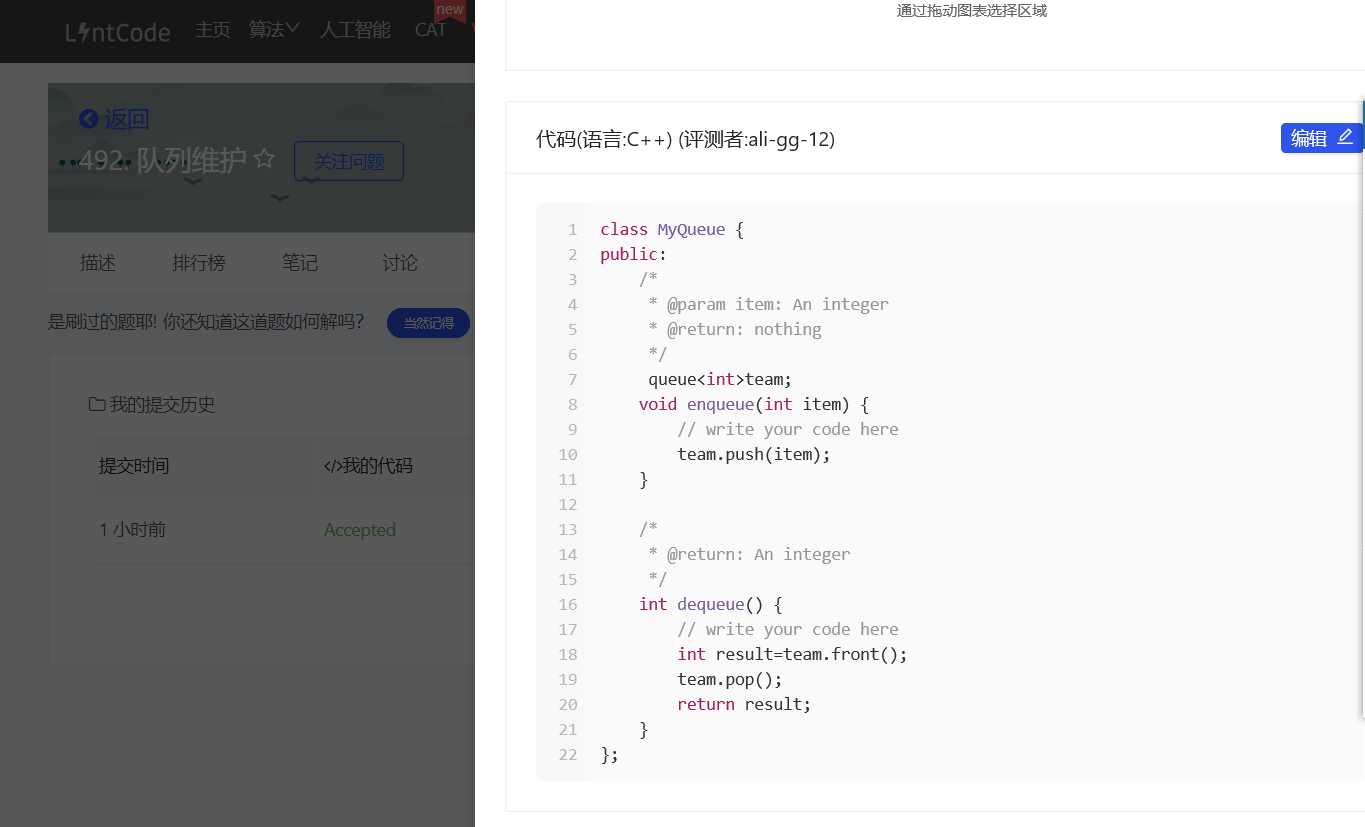
1. 删除链表中的元素



思路：先设立一个指针temp=head；如果temp=NULL则直接返回head；

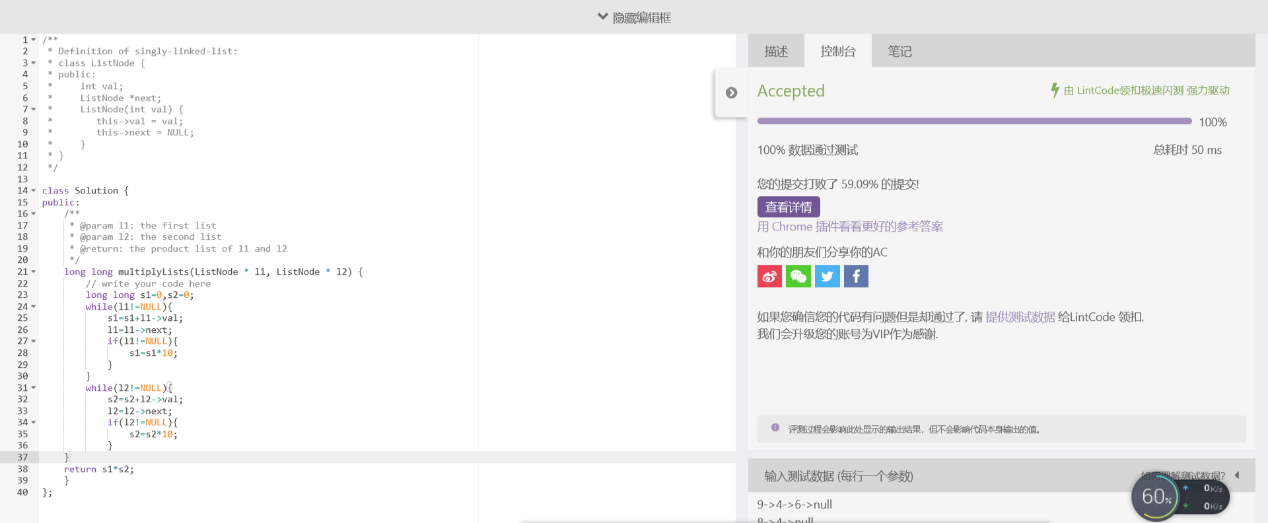
如果temp->next不为NULL则进入循环判断，如果temp->next的val和要删除的值相等，则让temp->next=temp->next->next即可，最后再判断头的val是否与要删除的数相等，如果相等就让head=head->next；

1. 队伍维护



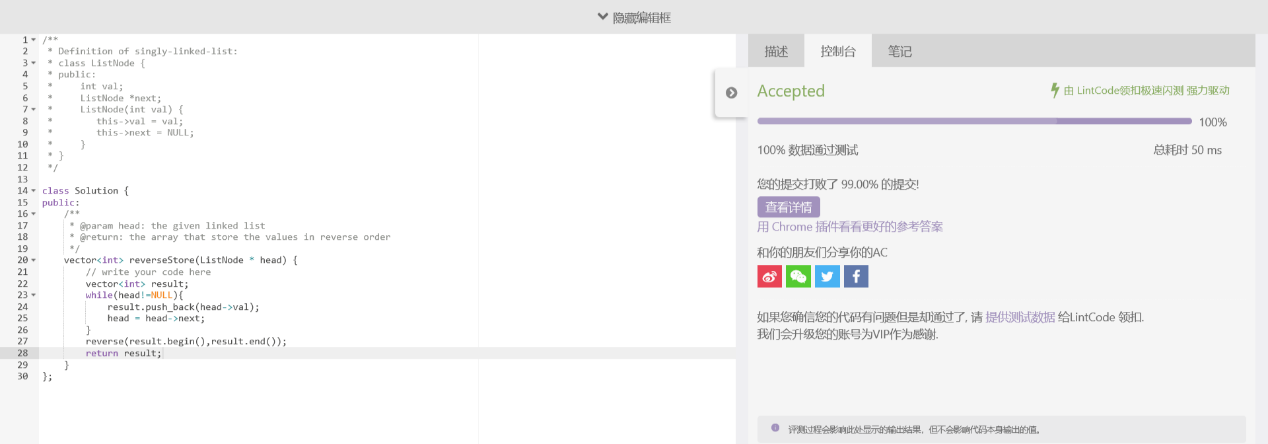
思路：创建一个queue的team，然后再enqueue函数中把目标值item给push进去；再dequeue中，用front（）把team中的第一个元素给整型的result，然后把第一个元素pop掉，最后返回result即可。

1. 两数相乘：



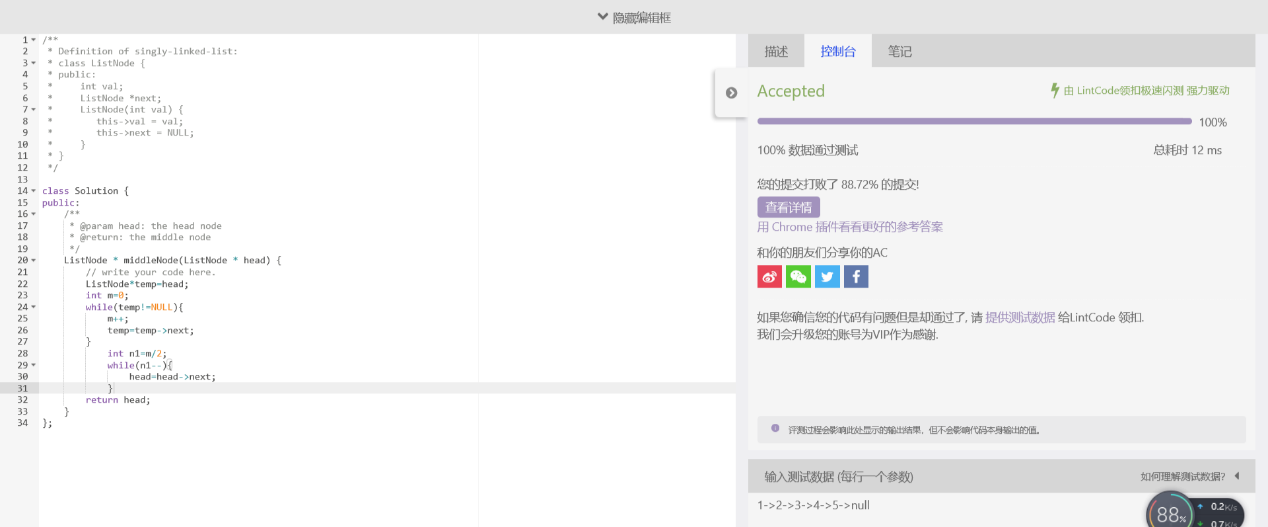
思路：设立两个long long类型的s1，和s2分别记录两个链表（l1和l2）算出来的结果，然后设立循环当l1！=NULL时进入循环，让s1=s1+l1->val,再让s1=s1\*10便可以求出l1表示的数，l2的循环也是如此，最后返回s1\*s2即可；

1. 相反的顺序储存



思路：不确定数组长度，用vector，配合push-back输入数据，最后用reverse反转就可以；

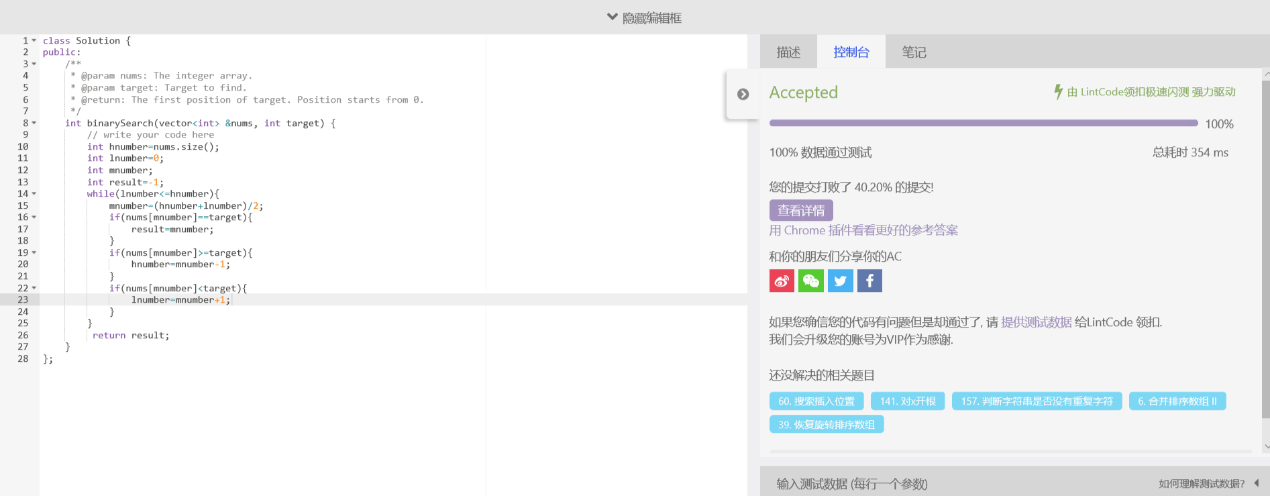
1. 链表的中间节点



思路：先遍历求出节点个数，然后求出中间节点，再一次遍历head让head到中间节点位置，返回head即可。

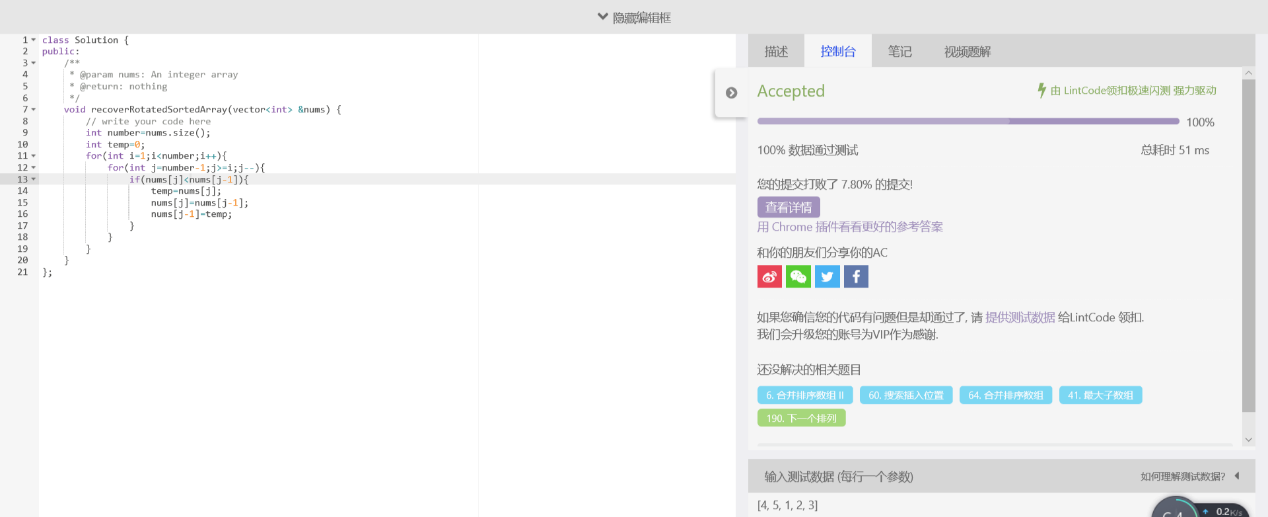
**第二部分：数组**

1. 二分查找：



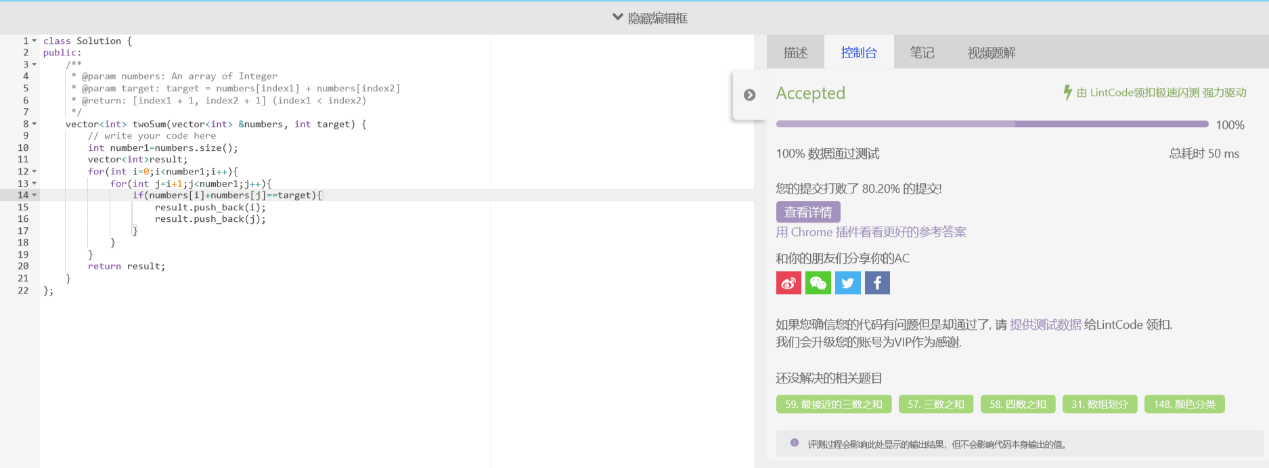
思路：设立三个整型数保存最高，最低和中间位；然后分情况判断，如果直接中间位等于目标值，直接返回，如果没有就判断目标值在中间位的前面还是后面，然后逐渐缩短区间直到找到。

1. 恢复旋转排序数组



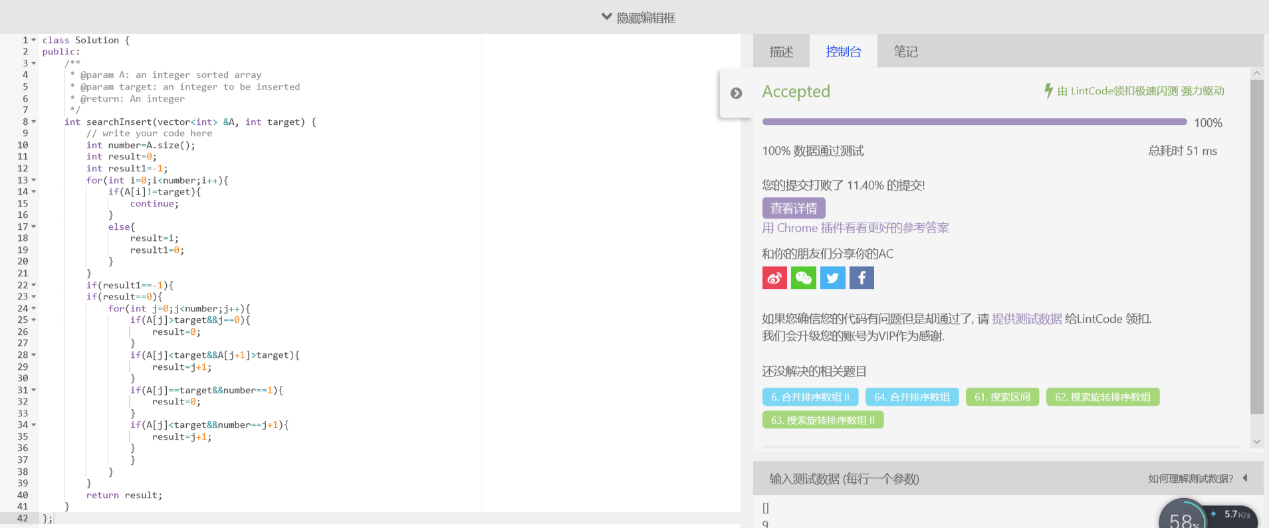
思路：先用size（）算出元素个数，然后再用排序算法排序即可；

1. 两数之和



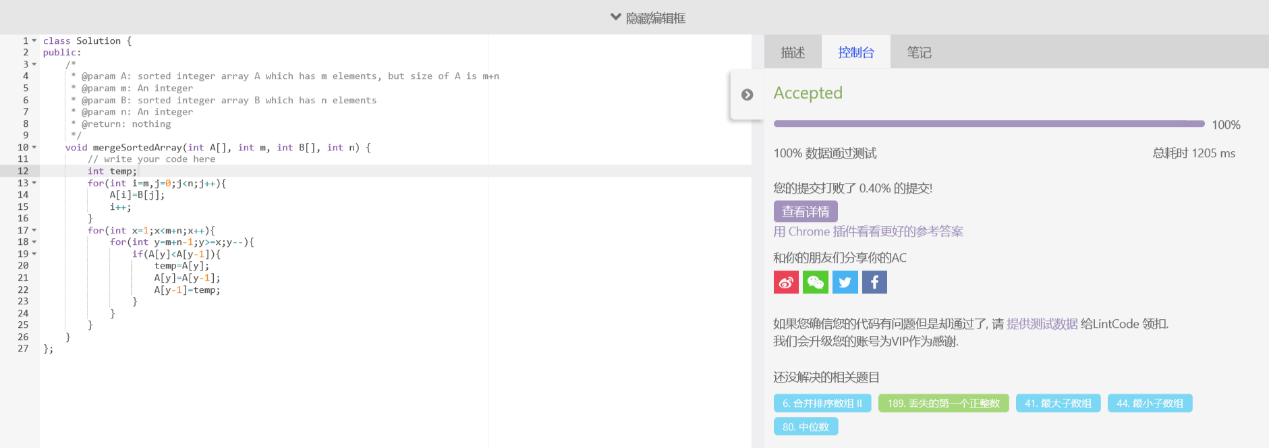
思路：先设立一个vector型的result，然后循环判断给的数组中哪两个值相加等于给定值，然后把这两个值的对应下标push\_back（）给result，最后返回result即可；

1. 搜索插入位置



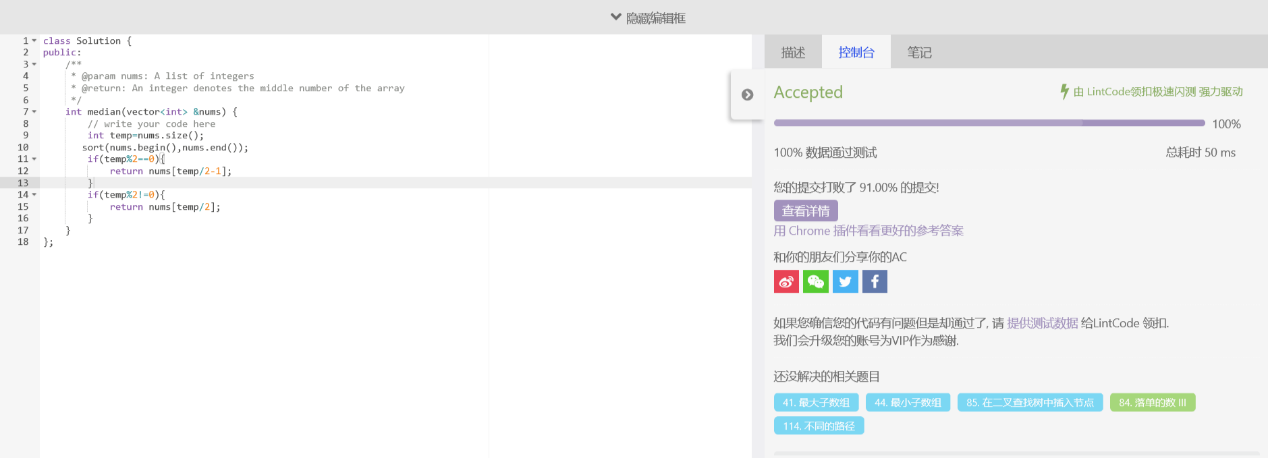
思路：分两种情况进行，设一个result1=-1区分两种情况。第一种是能在数组中找到他target，则结果result为此时下标，并且把result1赋值为0，让他跳过第二种情况；  
如果属于第二种情况，则考虑不同的情形  
1.数组第一个元素就比target大，则result=0；  
2.在数组中找到当前比target大，下一个比target小的位置，result设为当前位置+1；  
3.遍历整个数组，全比target小，则result为j+1，即number；

1. 合并排序数组



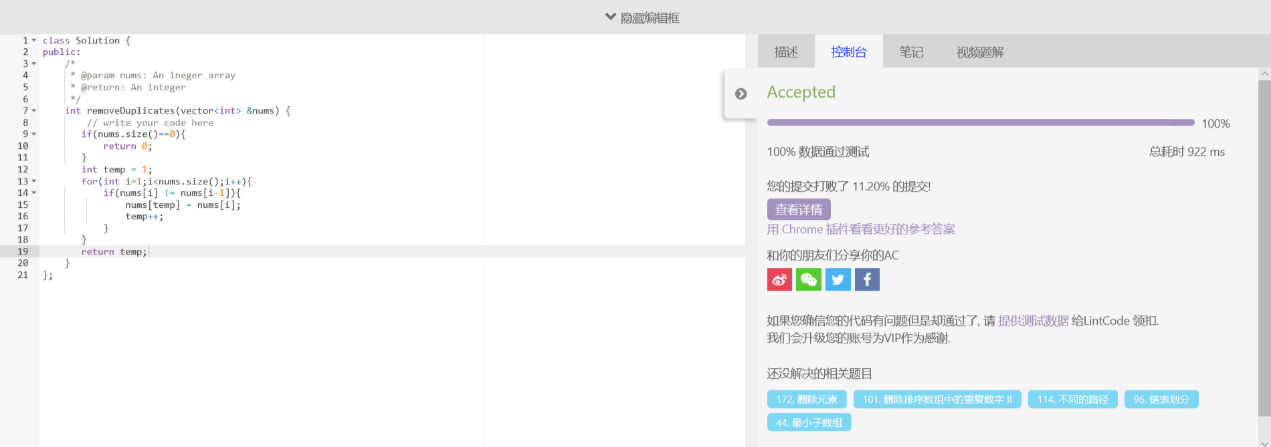
思路：先合并，后排序即可，其实排序可以用sort（A，A+m+n）直接排序，只不过我想练练我对排序的手感就直接写了；

1. 中位数



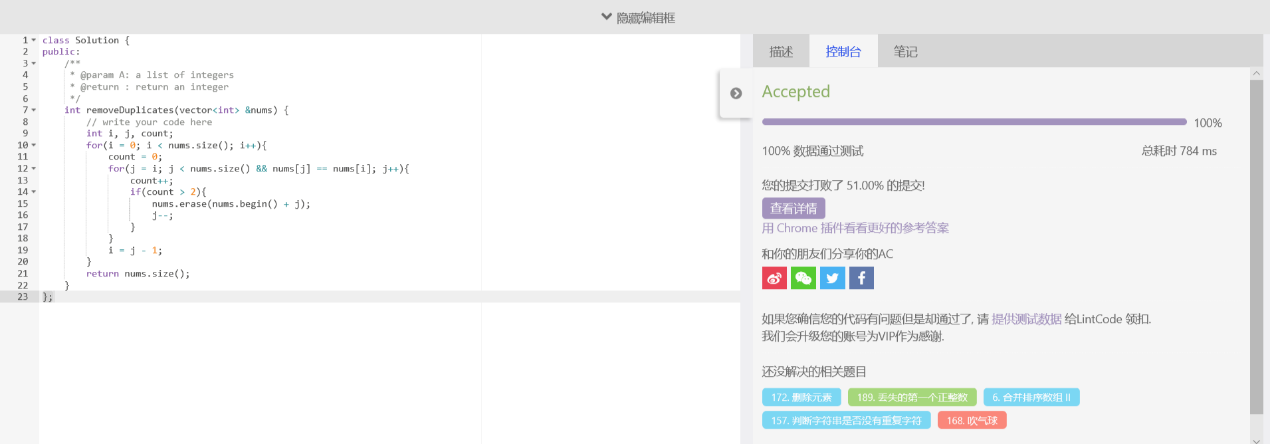
思路：调用sort（）进行排序，再判断是奇数还是偶数return即可

1. 删除排序数组中的重复数字1



思路：删除就是要把下一个数和重复的一个数给替换，用temp来记录新数组长度，然后返回新数组长度即可；

1. 删除排序数组中的重复数字2



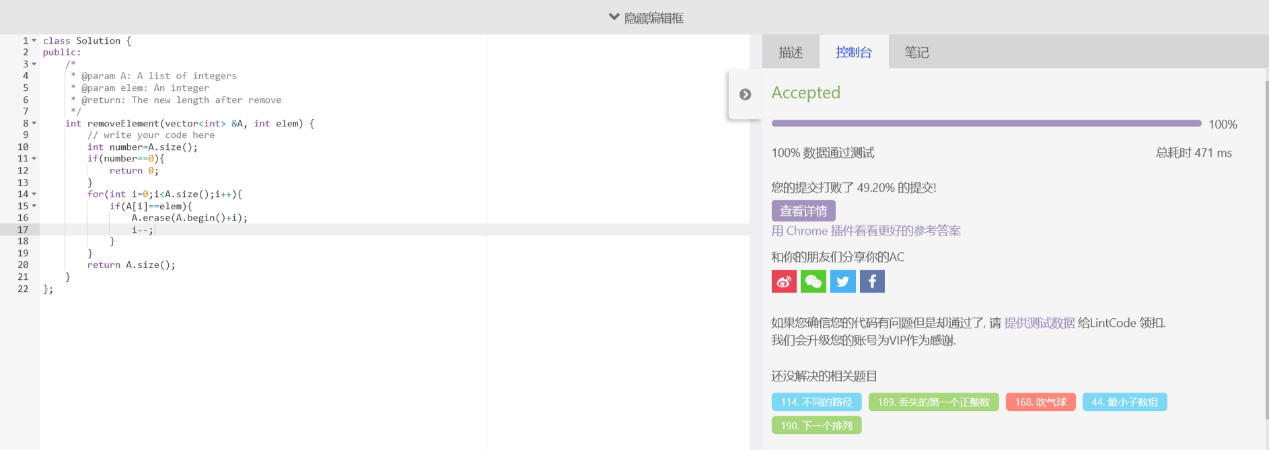
思路：设一个count来计数看是否出现2次以上，然后进入循环，如果出现2次以上就用erase（）删除掉，最后返回新数组长度即可；

1. 合并区间



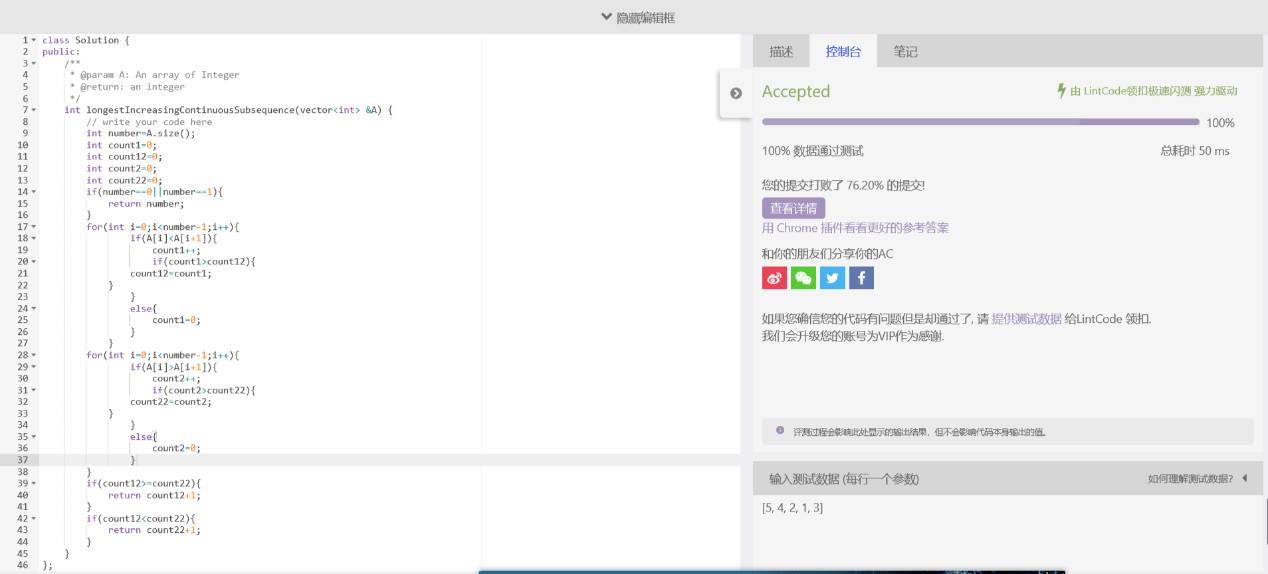
思路：如果数组为空，则直接返回，否则先按start，用sort（）排序，之后进入循环，如果一个大区间把下一个小区间全包了，就用erase（）把小区间给删了，然后元素数-1，如果一个区间和另一区间有重叠但不全包，就把end赋值，然后把后面那个区间删了即可；

1. 删除元素



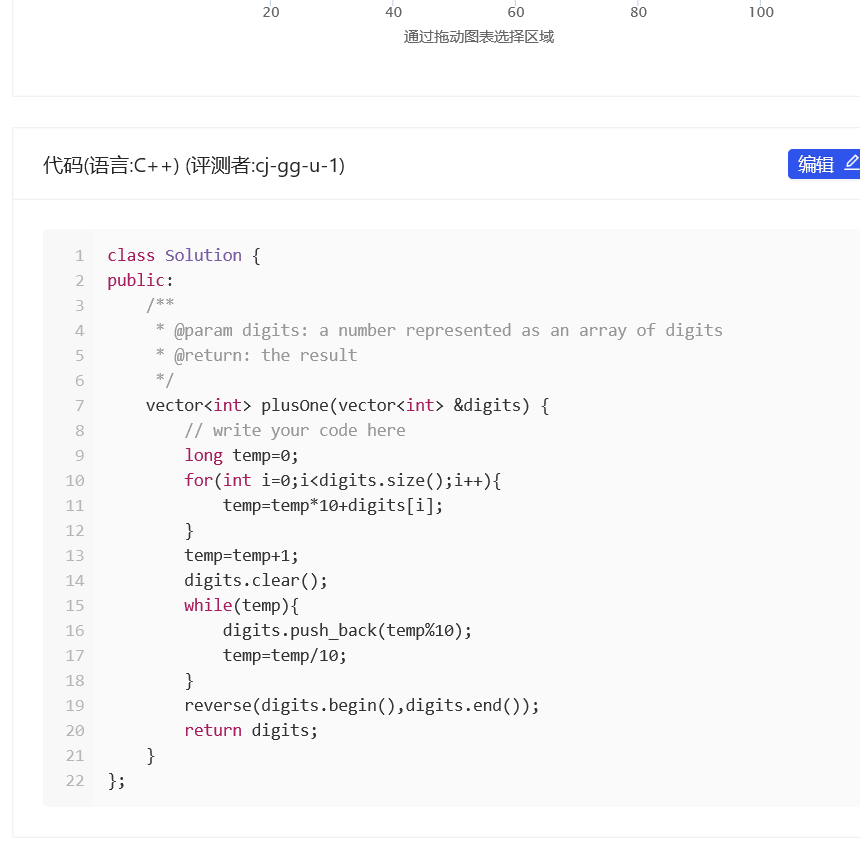
思路：先求出数组元素个数，如果为0直接返回0；如果不是则开始循环，当元素值等于给定值，用erase（）删除，然后返回新数组长度即可；

1. 最长上升连续子序列



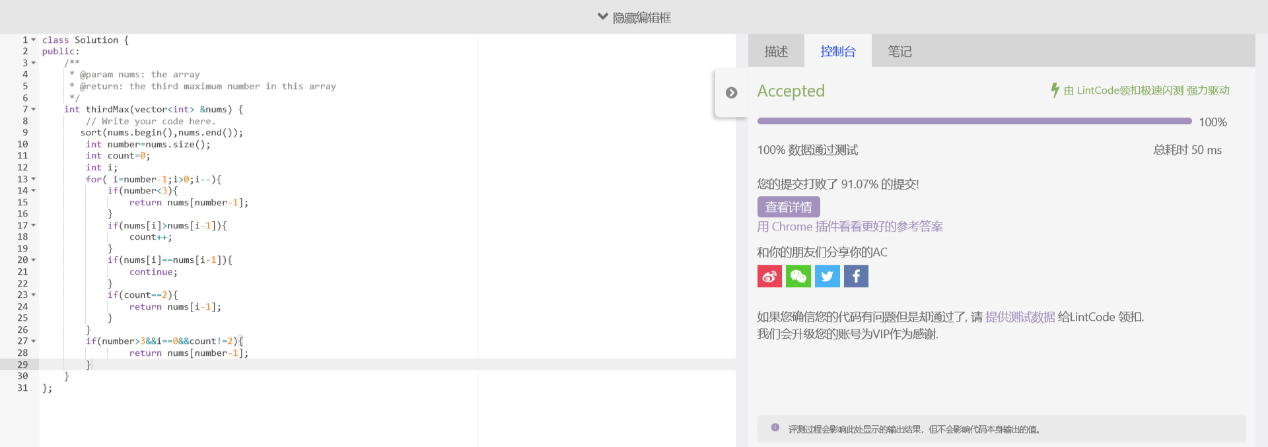
思路：抓住关键词，最长，上升，连续；  
先考虑特殊情况数组元素为0或者1；直接返回数组元素个数；  
然后按升序和降序分别这样的子数列中所含的元素个数；  
最后判断是升序的多还是降序的多，返回多的即可；

1. 加一



思路：先把数组表示的数用循环求出来，然后让这个数+1，最后再进入循环把每个位都返回到数组中，再用reverse（）反转即可。

1. 第三大的数



思路：先对所求的队伍用sort（）排序，之后按从最末尾开始，找第三大的数，count是用来记录第三大的数是否出现，当count=2时出现，并且还要考虑数组元素有没有到达3个的情况，如果没有，则直接返回最大的数，还有一种情况就是遍历完了，数组长度大于3，但是没出现第三大的数，也返回最大的数。

14，15这两道一道是那个不同的路径2，我看了看讨论好像涉及到了一些动态规划，另一道是最近的城市那道题，我如果用很多的遍历去一个一个找最近的城市的话估计太雍长了，所以这两道我等到后面学了数据结构后再解决。