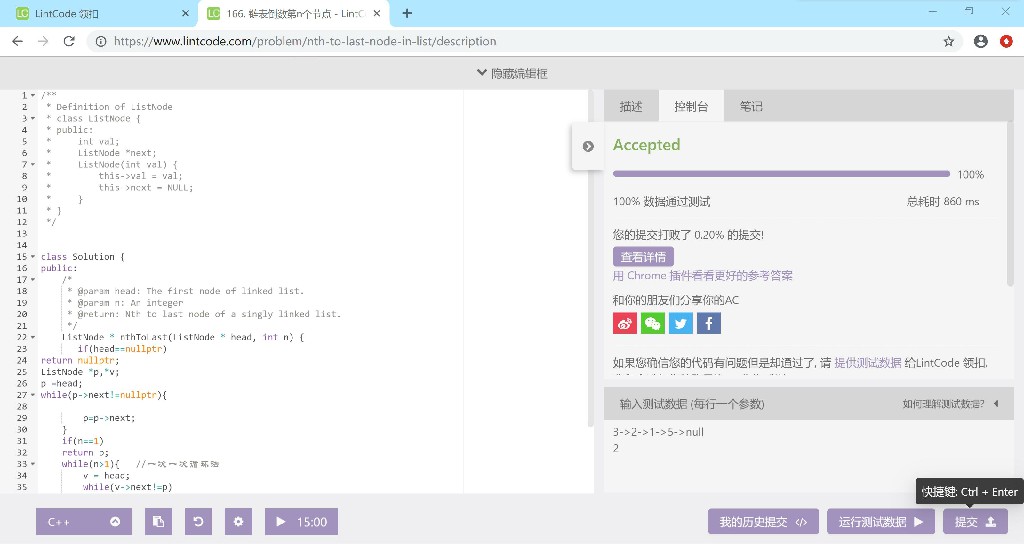
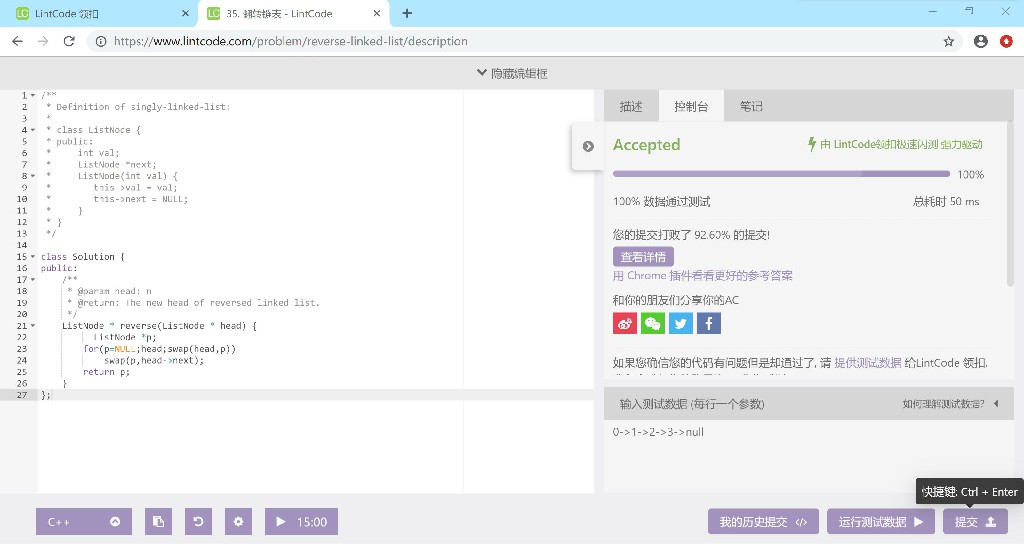
**（删除）倒数第n个节点：创建两个指针，先让第一个指针移动n个节点，然后第一个第二个指针一起移动，当第一个指针走到尾节点的时候第二个指针所指就是倒数第n个节点。再执行删除操作就是删除倒数第n个节点。**

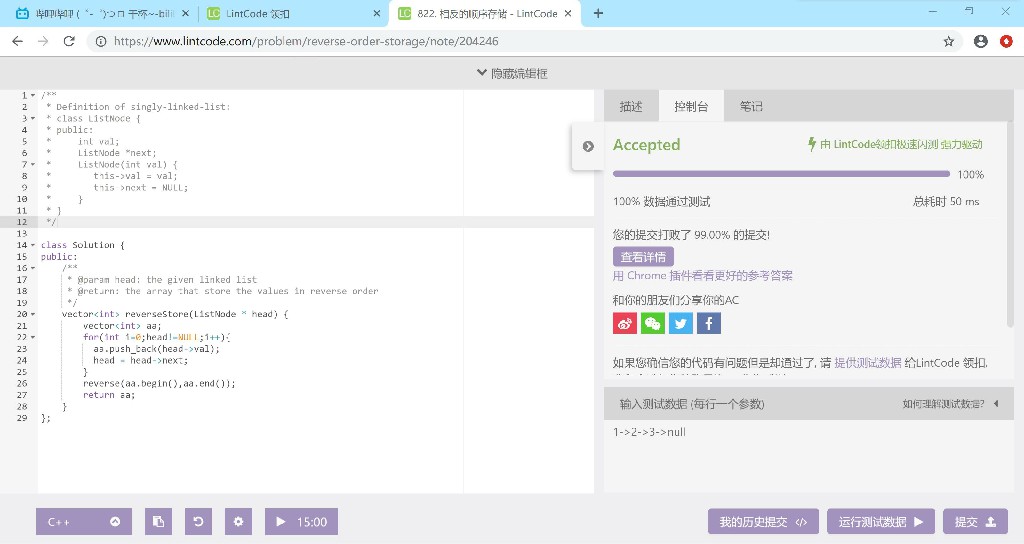




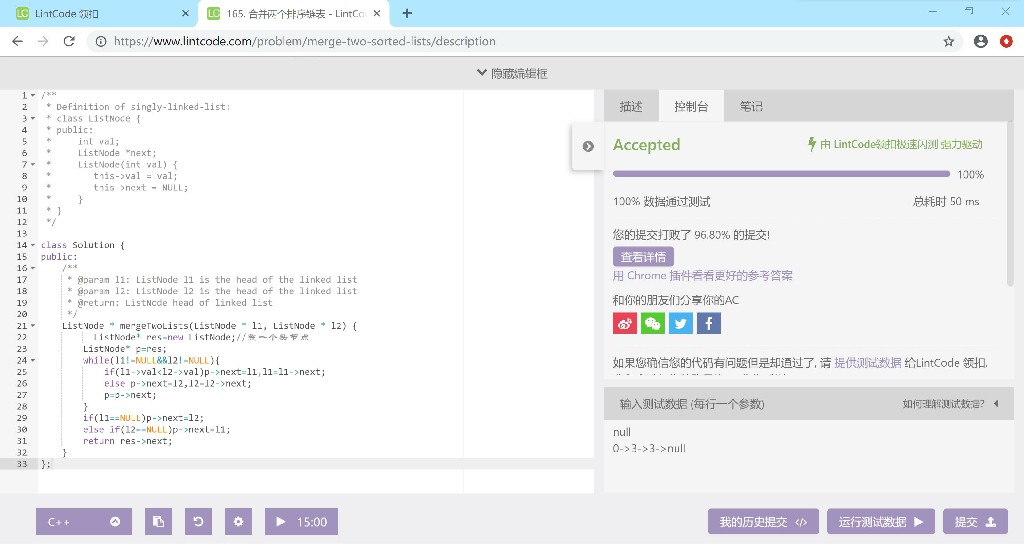
**翻转链表：定义一个指针，依次交换前两个节点，然后指针后移**。



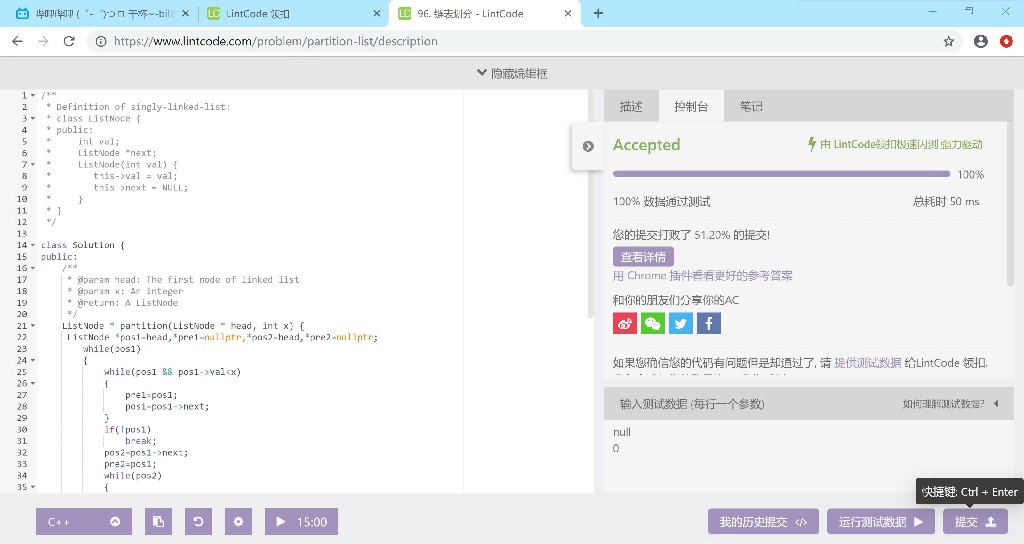
**相反的顺序存储：每次把下一结点的头往列表的首部添加，用vector，push-back输入数据，然后反转。**



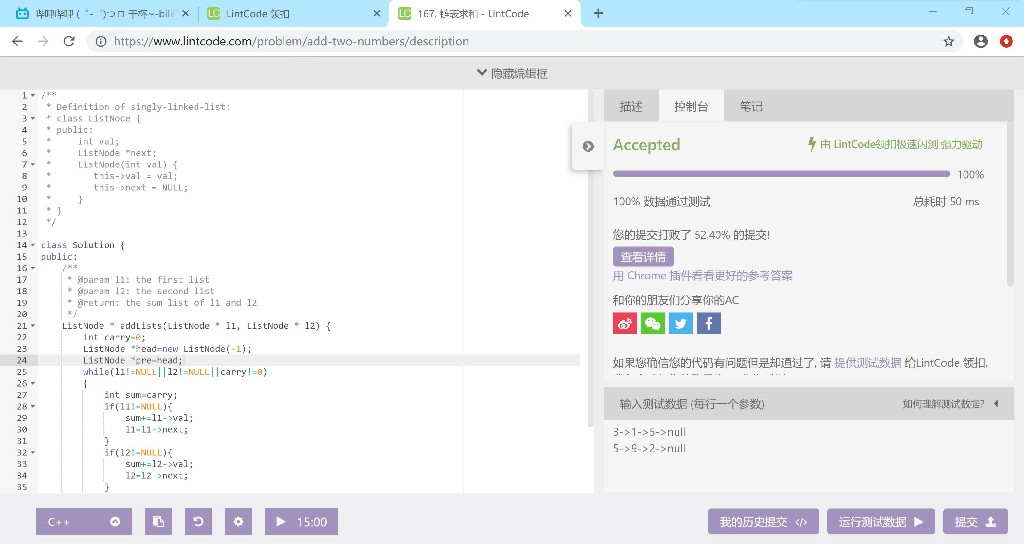
**合并两个排序链表：新建一个链表，当l1和l2不为空的时候，先判断l1和l2的值，将小的value依次加入。如果l1或者l2其中有一个为空，直接返回不为空的那个链表。**



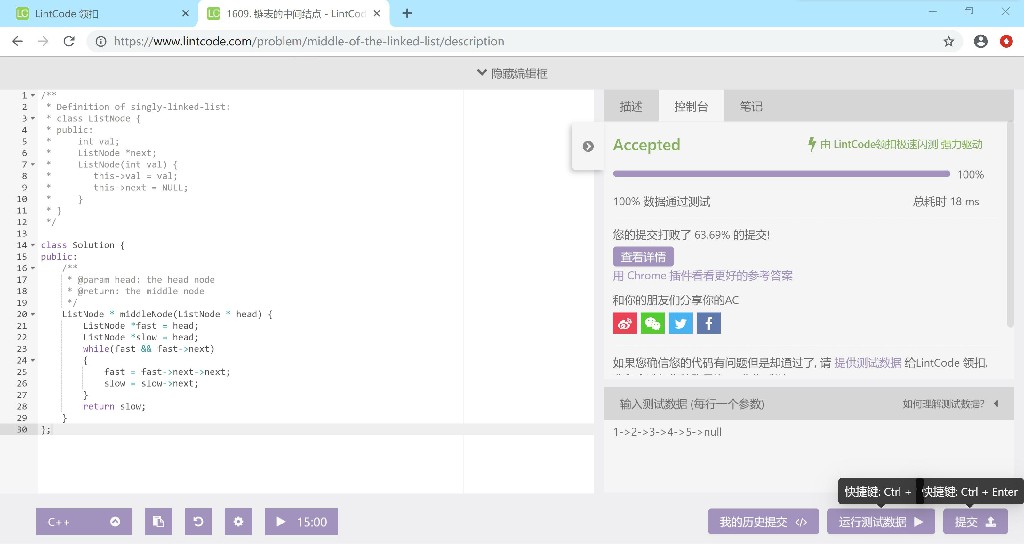
**链表划分：用两个指针连接左右两部分，得到两个链表，一个是小于x，另一个大于等于x，然后将两个链表拼接到一起就行了。**



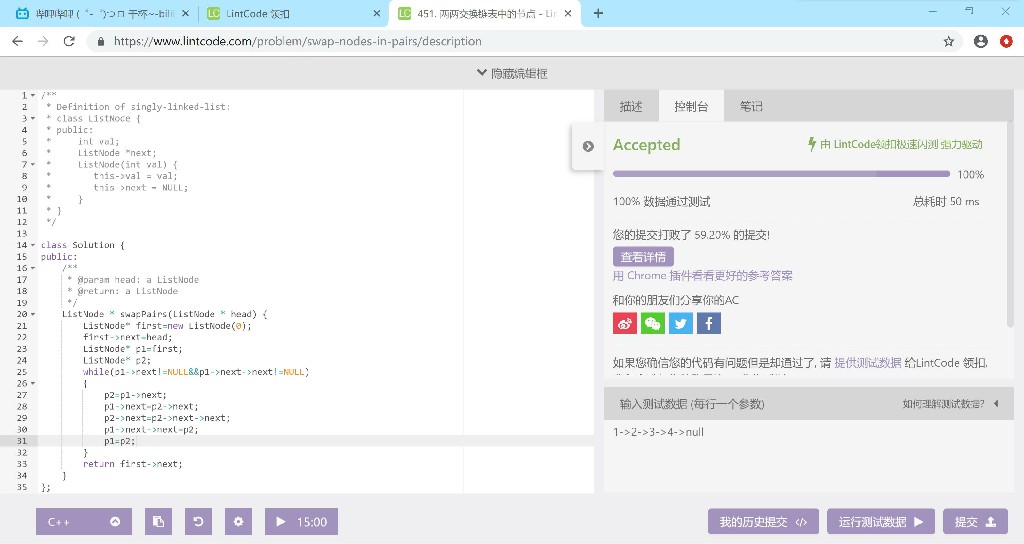
**链表求和：新建一个pre指向head，后面head不动，只有pre移动，然后返回head->next。主要需要考虑进位问题以及节点数不存在的时候新建节点的问题，进位数就是两个数的和/10，节点数则是两个数的和%10。**



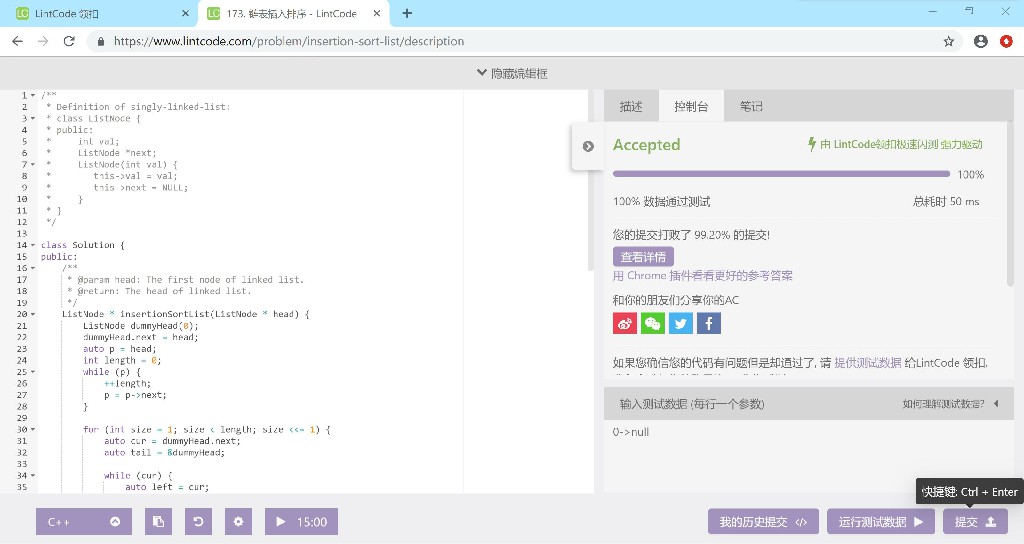
**链表中间节点：定义两个指针，快指针每次后移两个节点，慢指针每次后移一个节点，当快指针为NULL或快指针->next为NULL时，慢指针就是链表中间的节点，然后返回慢指针。**



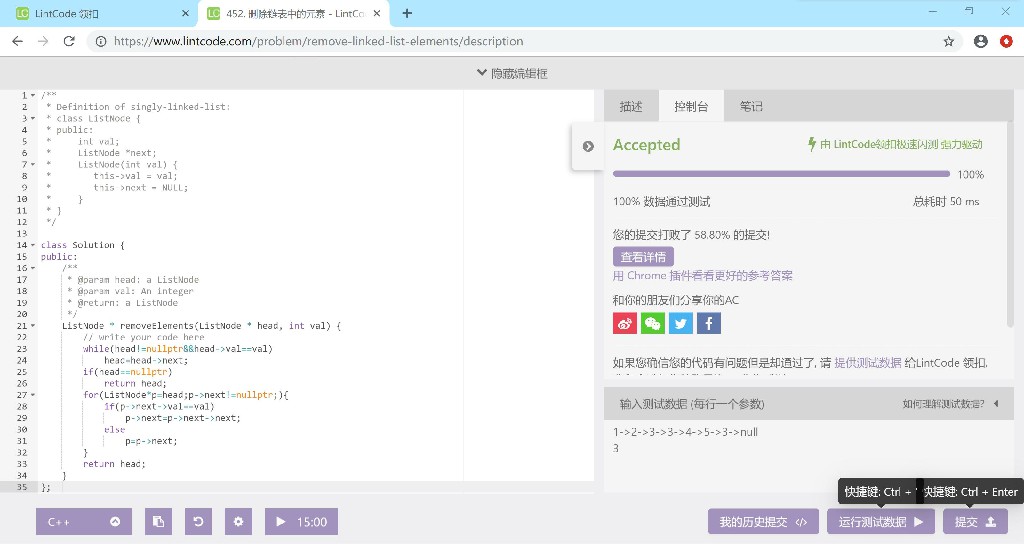
**两两交换节点：先用first节点固定头节点，然后定义两个节点，一个是需要交换的节点前面的节点，然后将第二个节点与它的下一个节点交换。**



**链表排序：用递归的话会超时，先两个两个的排序，然后再四个四个的排序，直到完成全部排序。**



**移除元素：不断后移节点来判断节点是否和要删除的值相同，相同的话就向后移动一个节点实现删除操作。**



**删除重复元素：遍历链表，如果发现相同的就通过 pi->next = pi->next->next实现删除操作。**



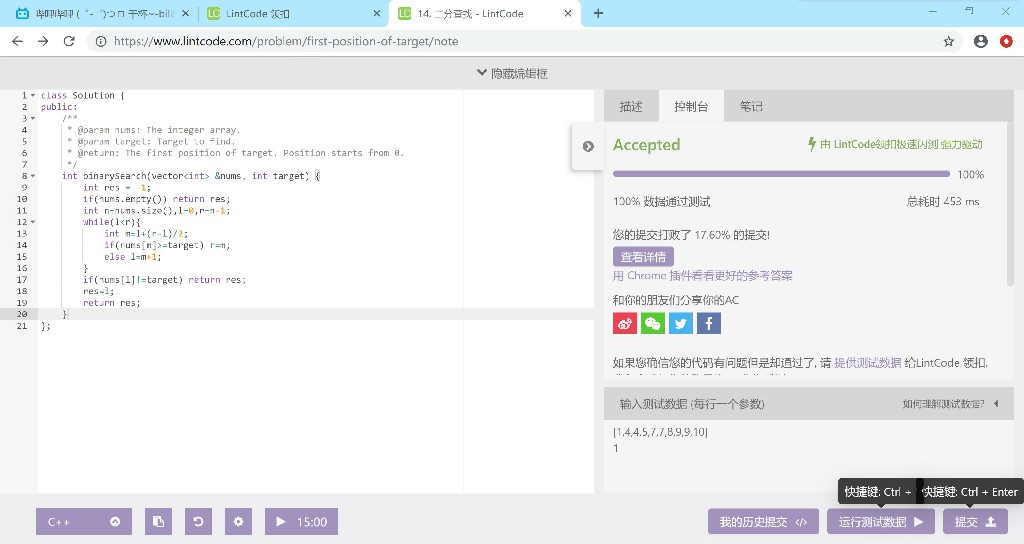
**不同路径2：网上找的视频讲解，使用动态规划，主要就是所以从坐标 (0, 0)(0,0) 到坐标 (i, j)(i,j) 的路径总数的值只取决于从坐标 (0, 0)(0,0) 到坐标 (i - 1, j)(i−1,j) 的路径总数和从坐标 (0, 0)(0,0) 到坐标 (i, j - 1)(i,j−1) 的路径总数，当ij处是障碍的时候，那么路径就是零。**



**第三大的数：依次处理数组中的每一个数。处理每一个数时，将其依次与最大、次大、第三大……的数进行比较，如果它大于对应的数，就先从最后一个数开始腾位置，最后把新的数放到腾出来的位置里。**



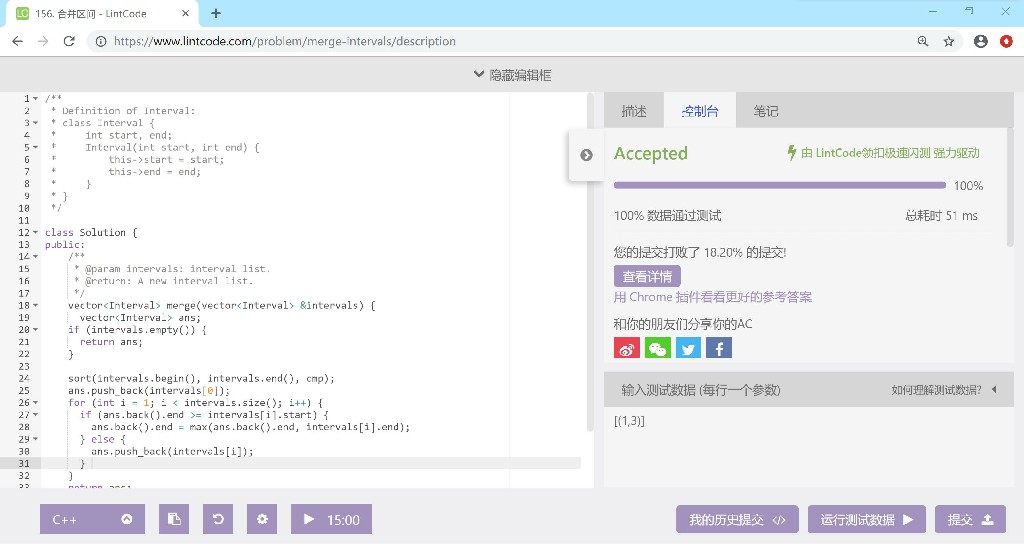
**二分查找：关键在于缩小查询范围，每次缩减一半，直到范围为1个元素，结束查找，需要用while循环，寻找第一个出现的元素，在每次循环时都判断中间元素是否和目标元素相同，相同时就给记下来，然后再进行下次循环。**



**合并排序数组：定义新数组的长度大小为m+n，比较原来两个数组的大小然后从后往前填入新数组中。**



**合并区间：判断前一个区间和后一个区间的关系，包含的话第一个区间不变，然后第二个区间后移；交叉改变第一个区间的端点值，第二个区间后移；不相交第一个和第二个区间都向后移动一个区间。**



**恢复旋转数组：先判断该数组是否已经排好序，如果没有排好序，那么这个数组肯定有一个数会比前一个数小，找到那个数，然后将这个数前面的数翻转，再将这个数之后的数翻转，最后将整个数组翻转，最后就是按照升序排列的数组。**



**加一：只要是某一位加1之后小于10，直接返回最后一个数加一。**

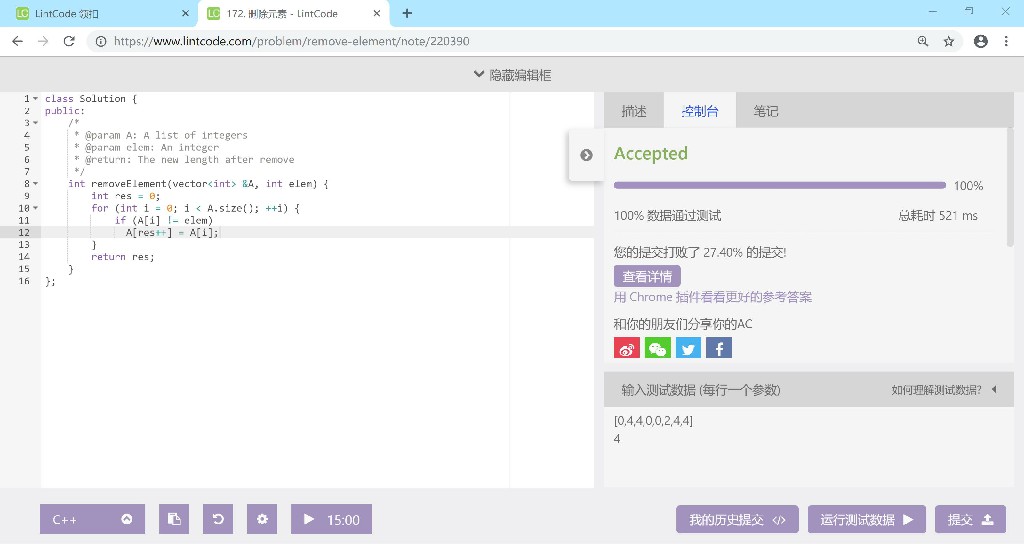
**如果加到最后一位都没有返回，那么说明要进1，只需要把首位设为1，其他位都是0就行了。**



**两数之和：二重循环寻找相加为目标数的两个数，然后判断大小对res赋值，最后返回res。**

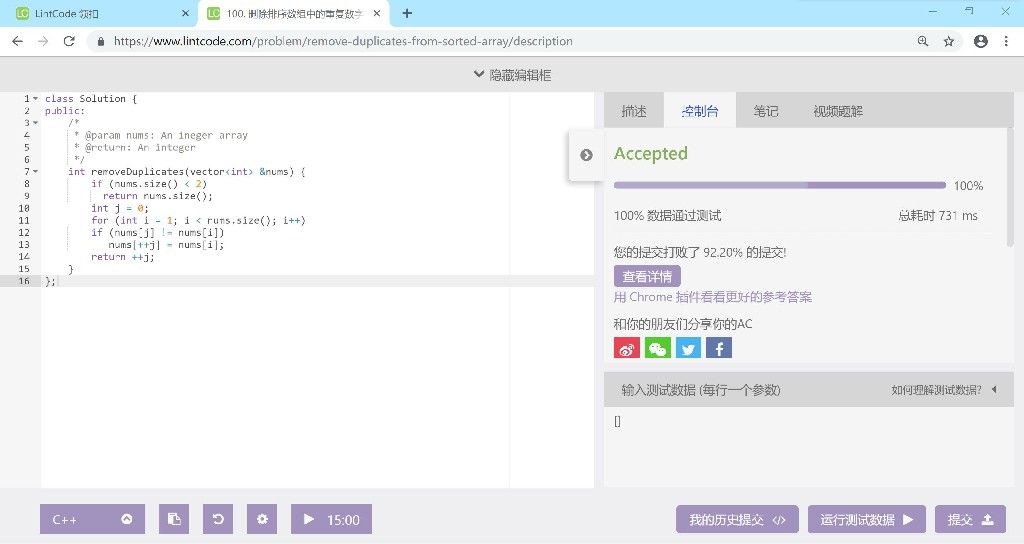


**删除元素：定义一个技术变量，然后遍历原数组，如果当前的数和目标不同，就把当前的数覆盖计数变量的位置，并将计数变量加1。**



**删除重复数字1：先判断数组的长度，如果长度小于2，那么不可能有重复的数**

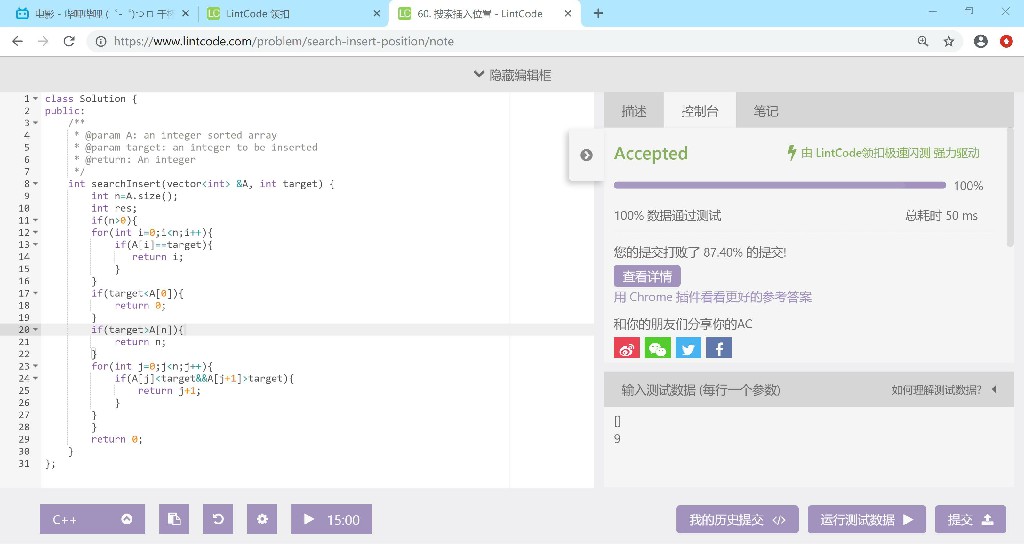
**字，直接返回原数组的长度，否则就判断相邻两个数是否相同，不同的话j加一实现计数，最后加上第一个数返回j+1.**



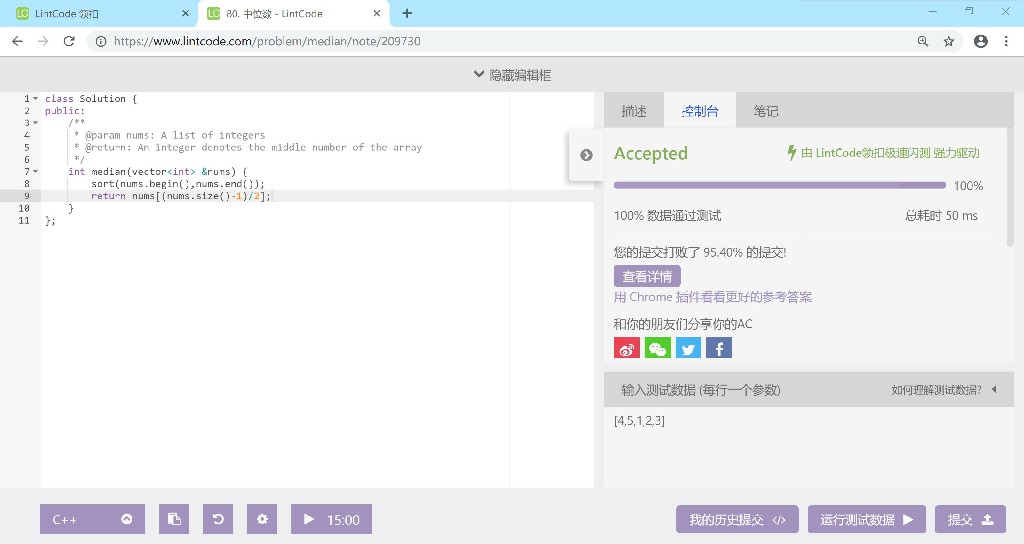
**删除重复数字2：判断数组长度是否大于2，不大于2直接返回原数组的长度，如果新元素加入后不会和前两个元素构成3个相同的元素，就将新元素加入到新数组中，并更新长度。**



**搜索插入位置：循环判断数组是否有目标，有的话就返回i，如果比数组最小的数小，则返回0，比最大的大，就返回数组最大值n，如果介于两个数之间，则返回j+1.**



**中位数：开始使用的是先用冒泡排序法实现数组的排序，然后判断奇偶再进行返回，这样虽然大多数的数据都能通过，但是当给出的数据非常庞大的时候会超时，然后参考了网上的快排方法。**



**最短无序连续子数组：先复制一个数组，然后排序，然后分别从头和尾向中间进行靠拢，如果相等的话就++和--，不相等的话就break；**



**最长上升子序列：一个上升序列，一个下降序列，遍历数组，如果遇到有不是上升或者下降的序列，获取最大值然后tmp重置。最后返回序列最大值。**

