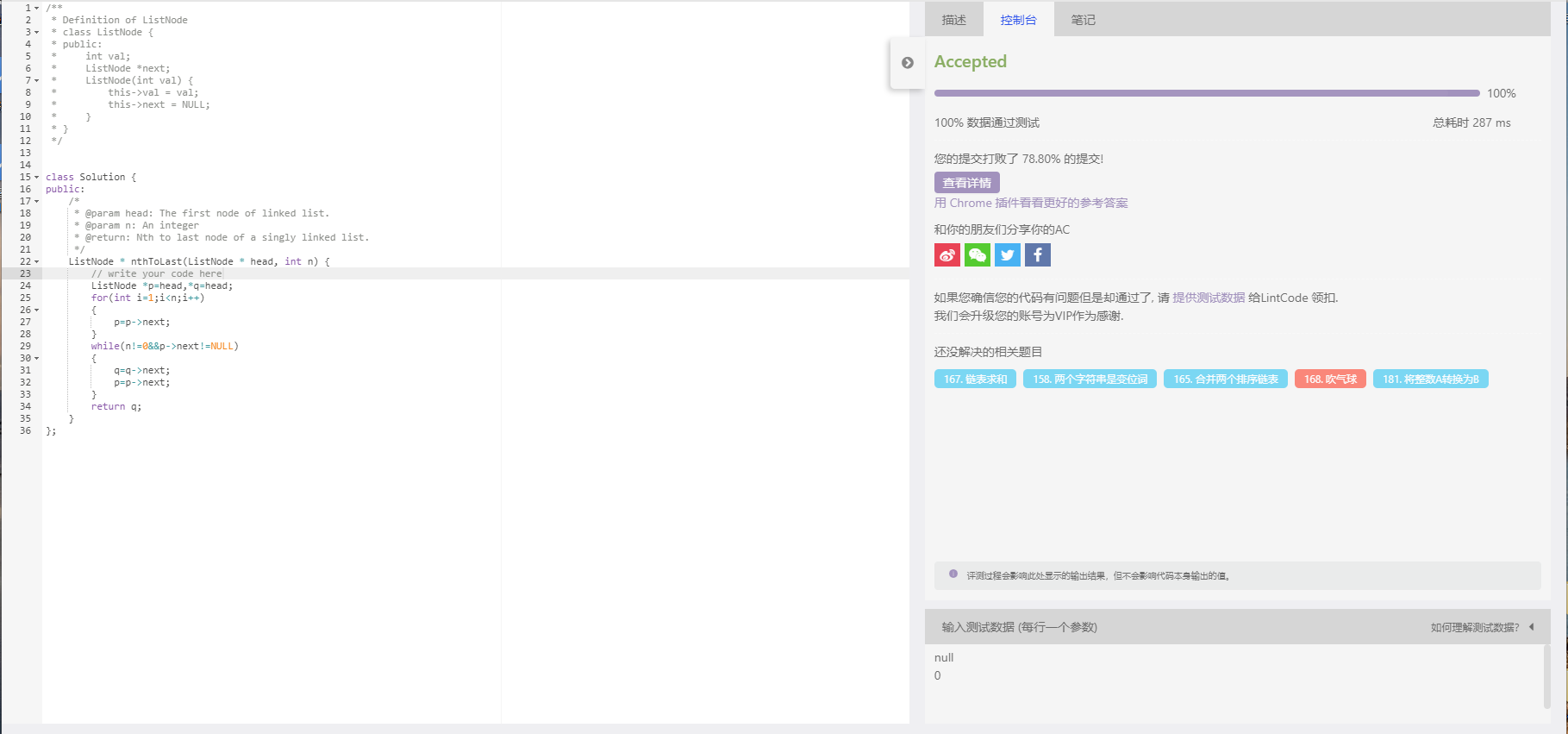
**链表**

1. **链表倒数第n个节点**

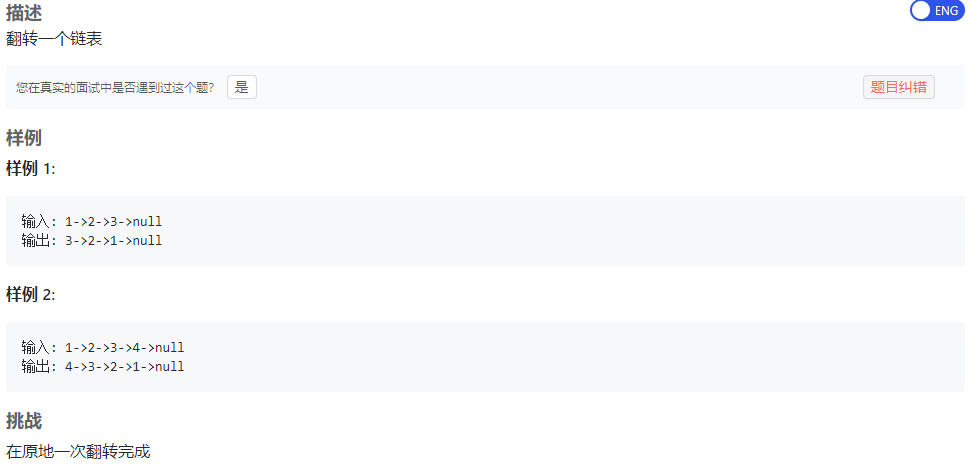


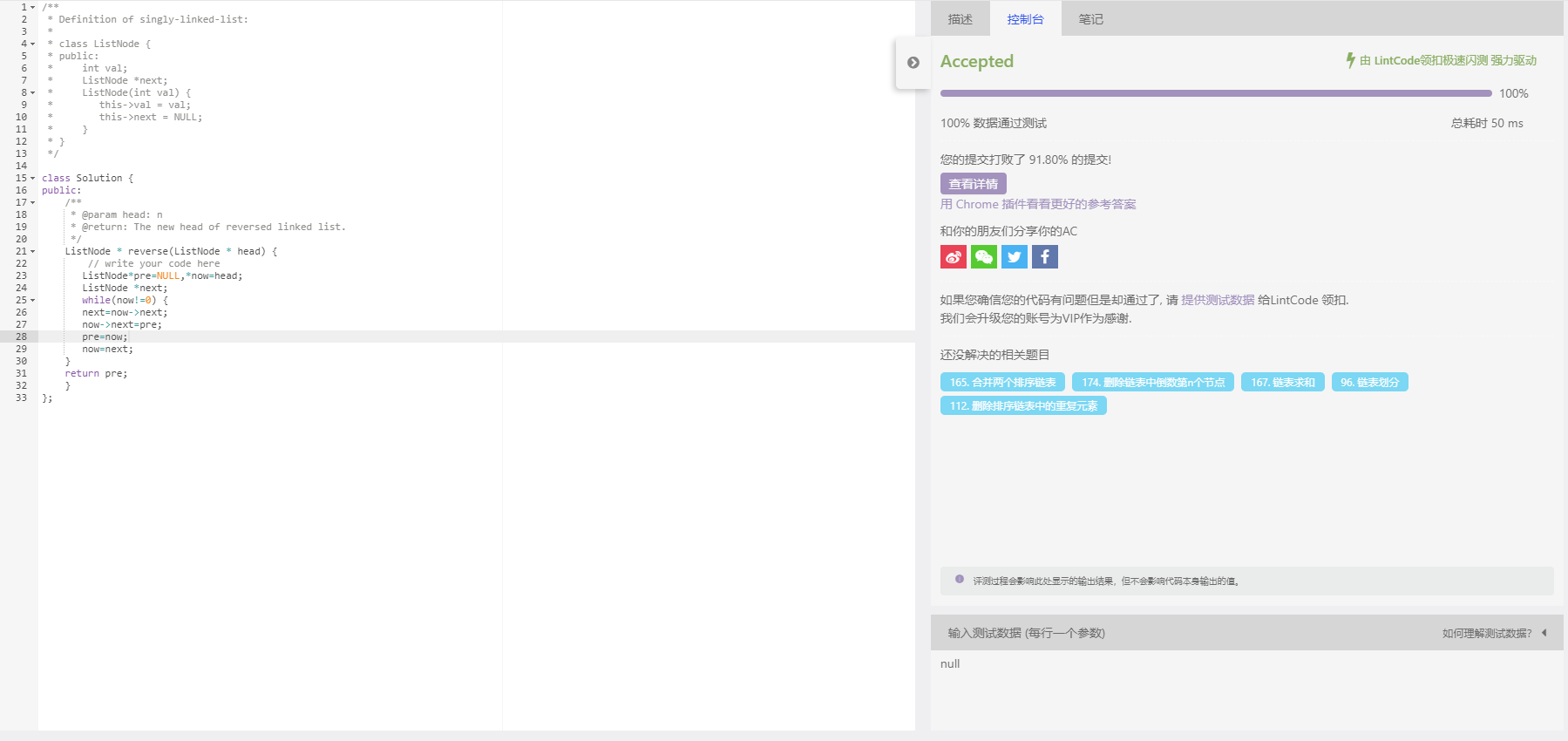


**实现思路：**

利用两个节点，先让第一个节点指向第n个节点，另一个节点指向头结点，然后两节点同时后移，知道第一份节点指向最后一个给节点，此时第二个节点即为所求节点。

**35. 翻转链表**



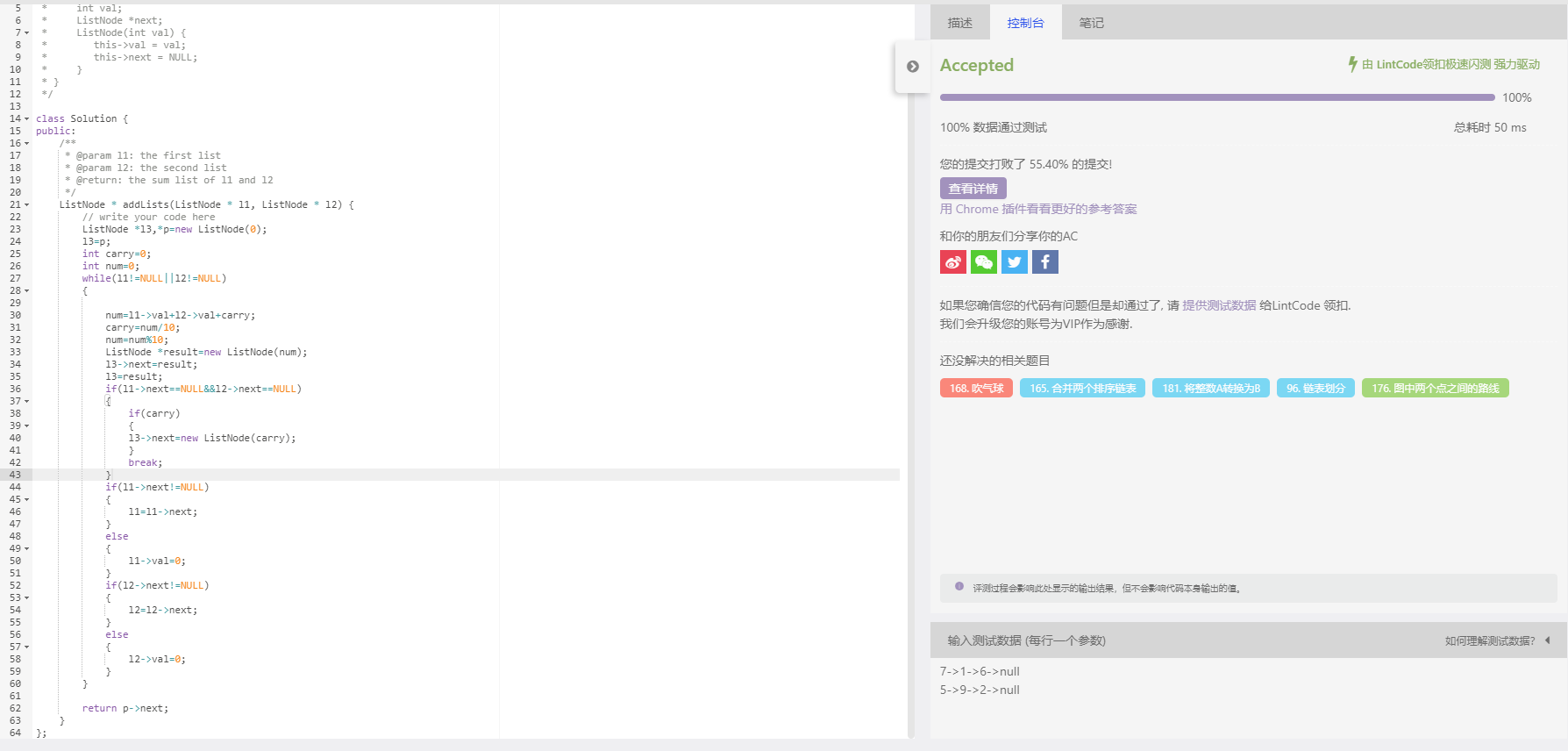


**实现思路：**

用三个指针now、pre、next分别指向当前节点、当前节点的前一个节点及当前节点的下一个节点。将next指向now的下一个节点，将now的next指向pre，再令pre等于now，now等于next，循环直到now为尾结点。

1. **链表求和**



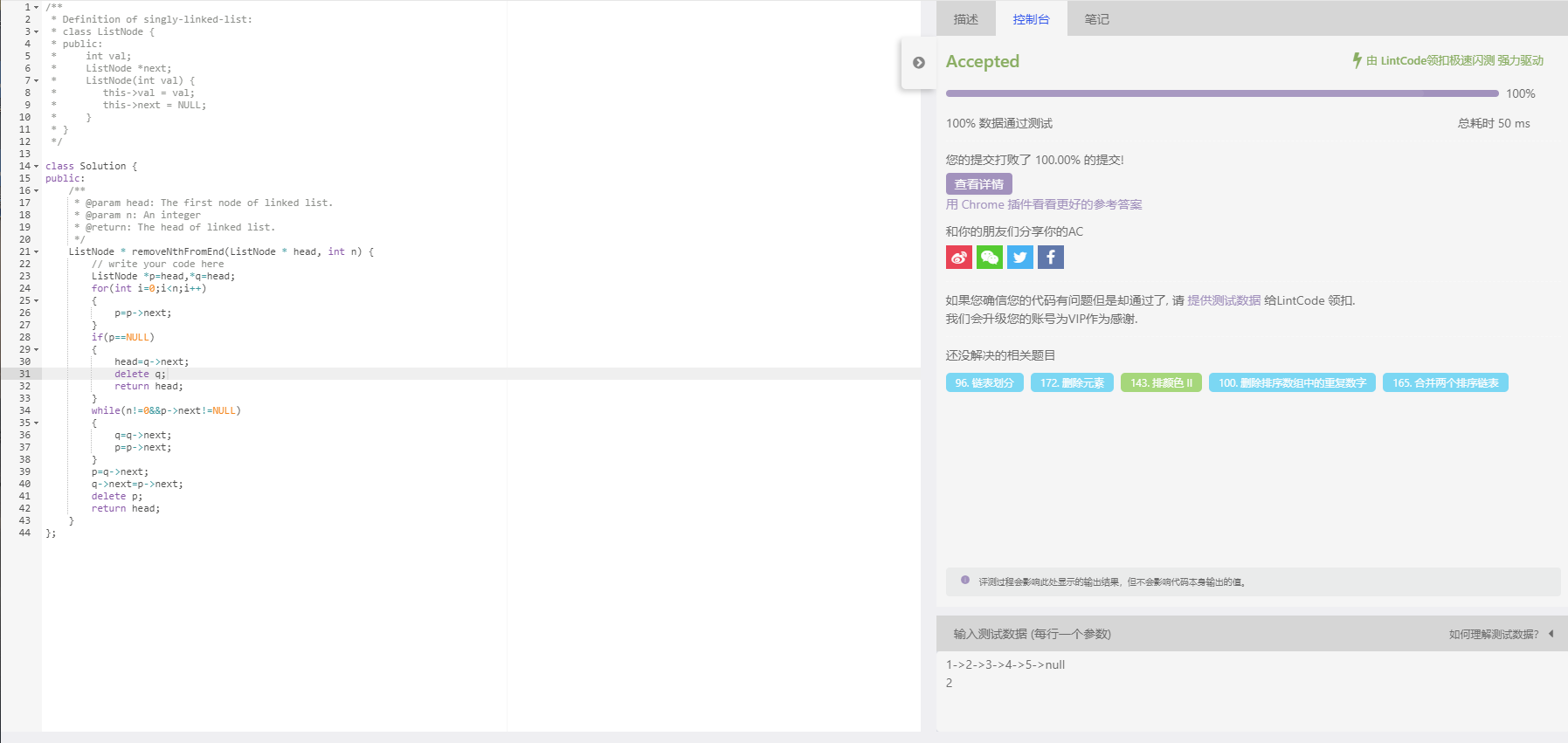


**实现思路：**

两个链表存储的是整数中相反的顺序，而结果也要求存储和的相反的顺序，可以将两个链表中的结点值依次相加，存入新链表，同时判断是否有进位，用于下次相加。如果两个链表的长度不相同，则将短链表尾结点的值设为0，进行相加。

1. **删除链表中倒数第n个节点**

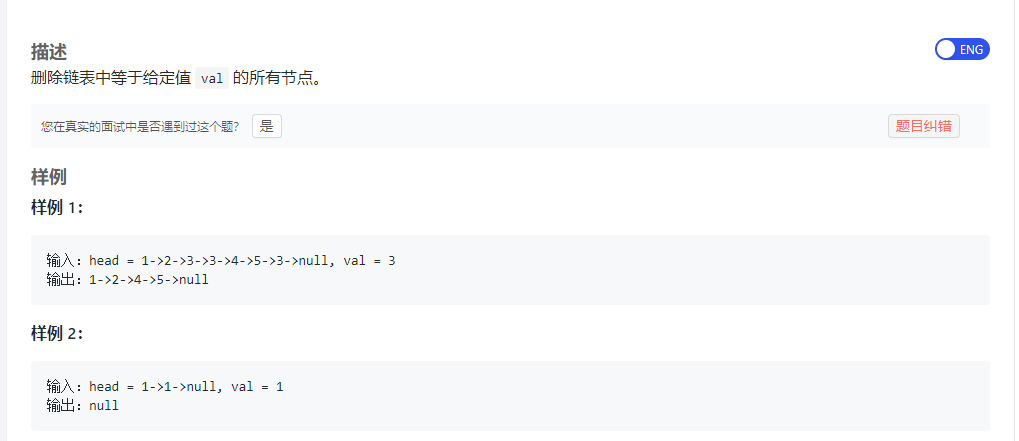


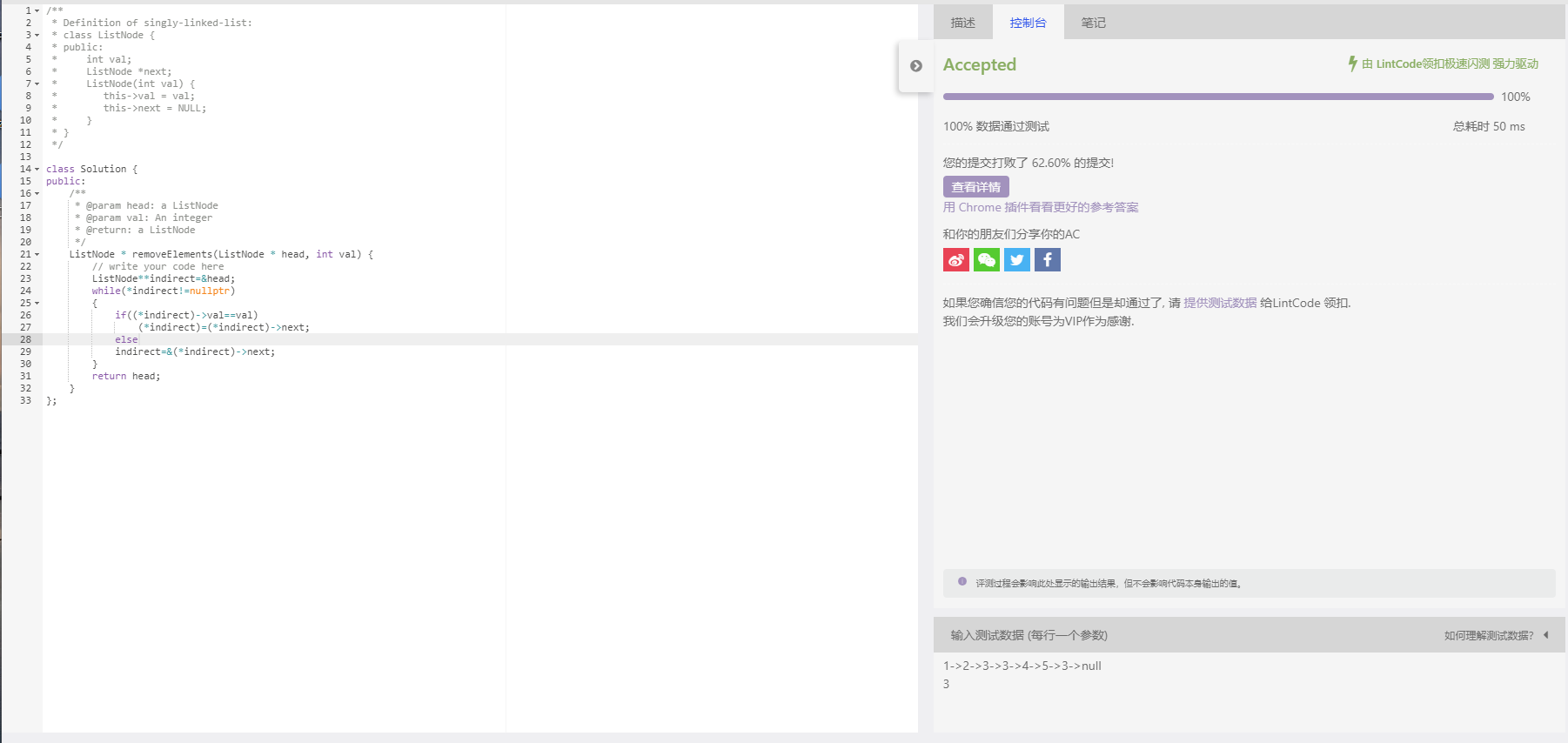


**实现思路：**

利用两个节点，先让快节点指向第n+1个节点，如果快结点此时指向尾结点的下一个结点，则删除头结点，并使头结点指向下一个结点。否则同时移动快慢结点，知道快结点指向尾结点，此时慢结点指向要删除点的前一个结点，做删除操作，返回头结点。

1. **删除链表中的元素**



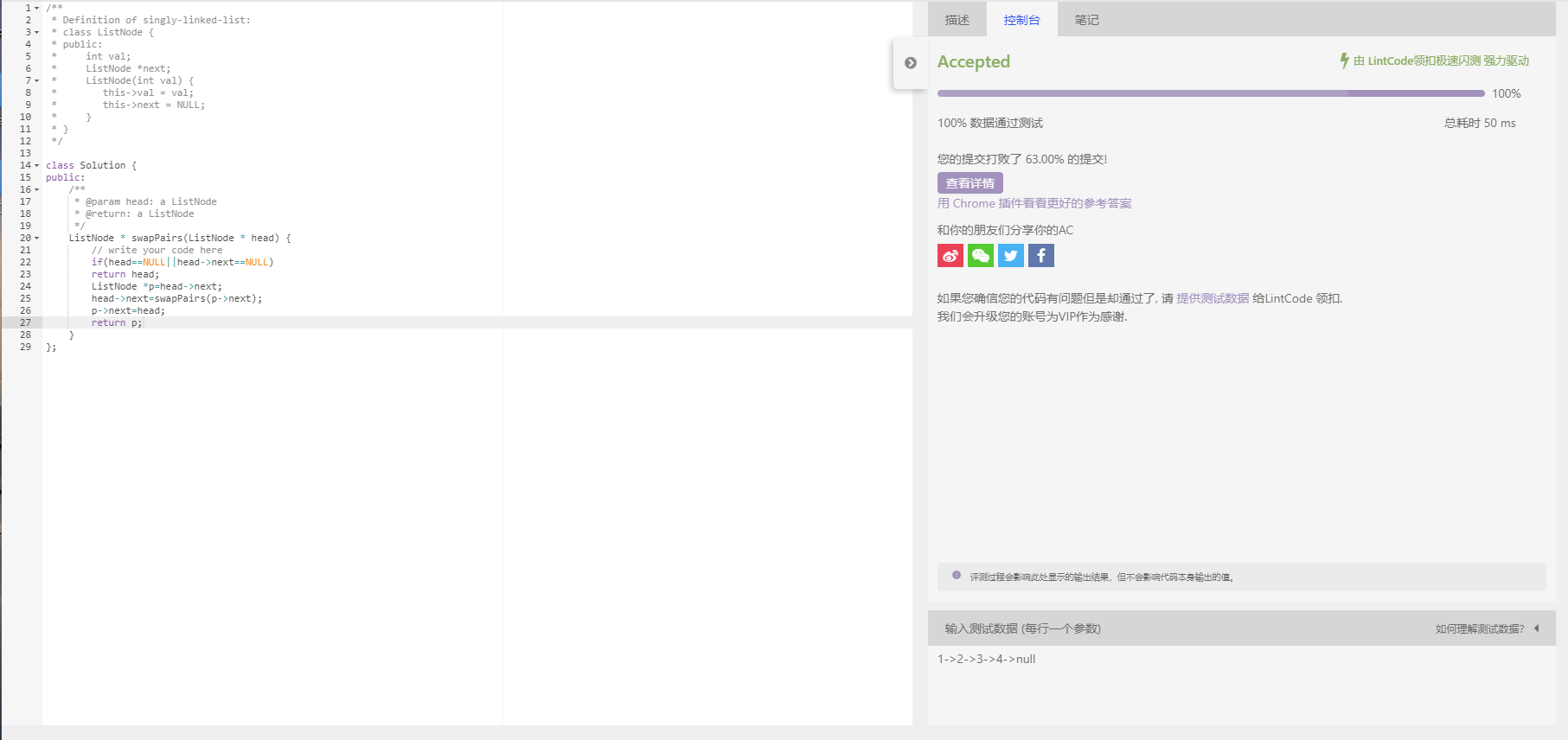


**实现思路：**

利用一个二级指针指向当前结点的地址，即前一个结点的next指针，通过遍历删除链表中等于给定值 val 的所有节点。

1. **两两交换链表中的节点**



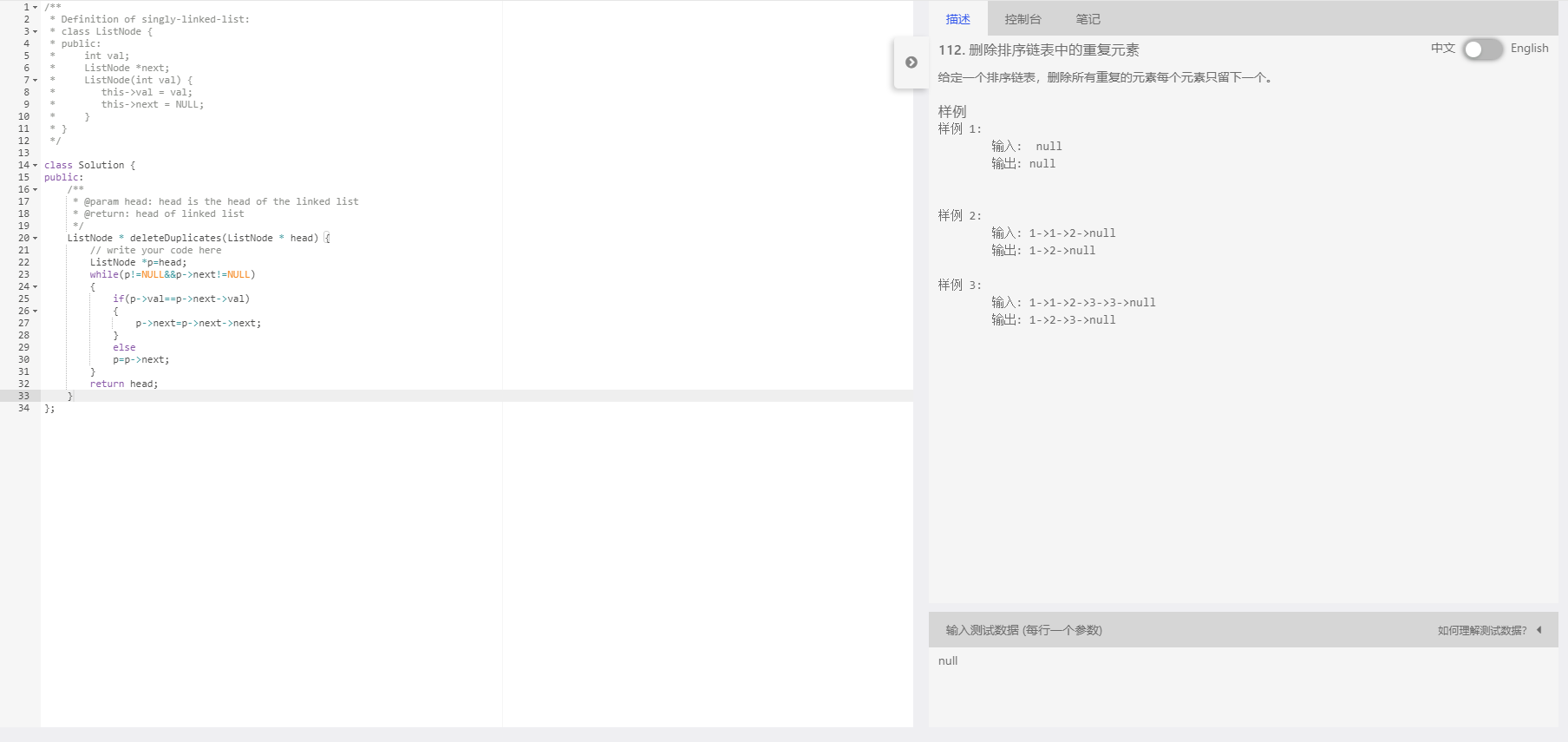


**实现思路：**

利用递归，将只剩空结点或单节点的情况作为递归退出条件，将当前偶数结点的第二个结点保存，并使第二个结点的next指向其后结点的递归结果，达到两两交换的目的。

1. **删除排序链表中的重复元素**



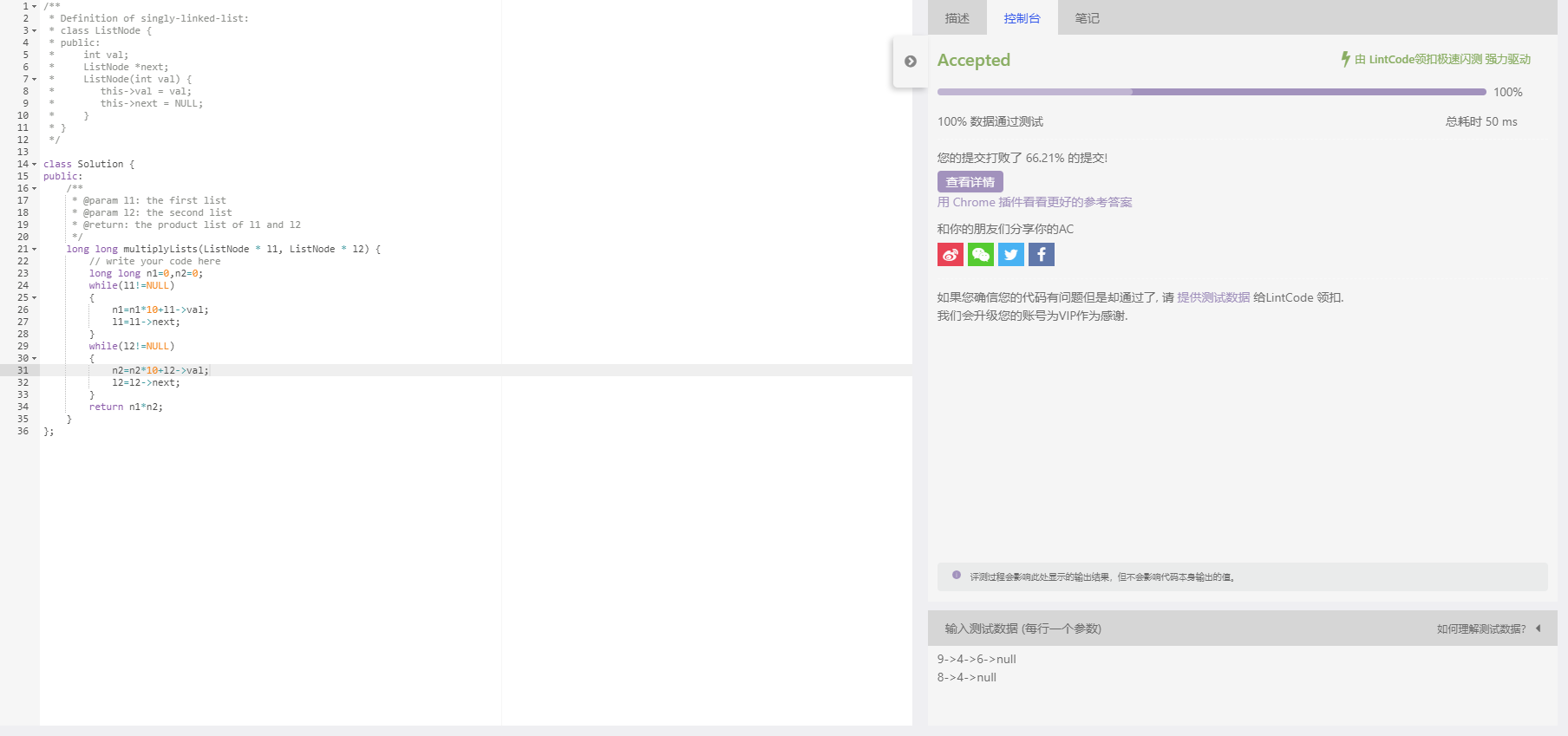


**实现思路：**

从头结点开始遍历链表，当前结点的值与下一个结点的值相等时删除下一个结点，否则当前结点后移，直到当前结点为空或者下一个结点为空。

1. **两数相乘**



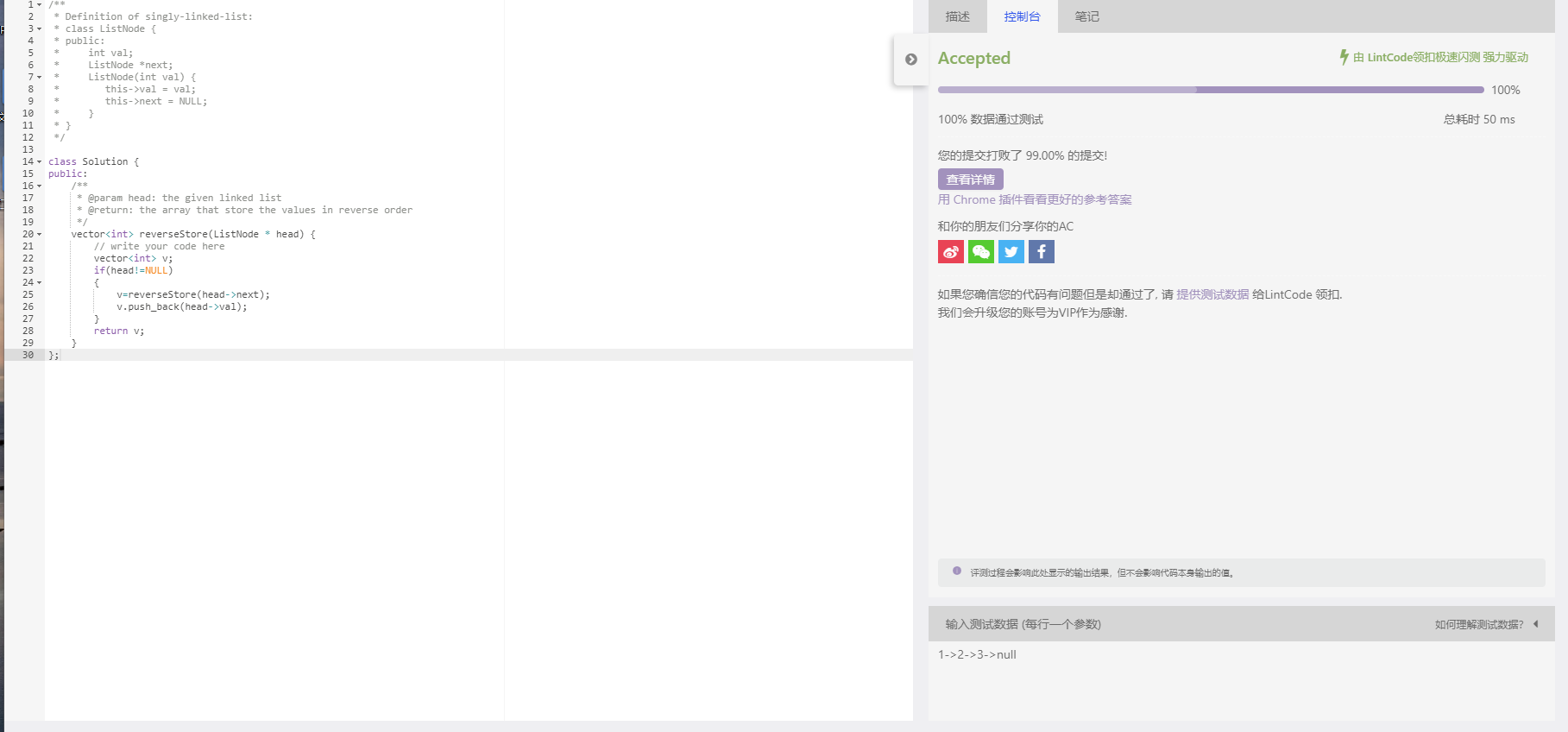


**实现思路：**

分别遍历两个链表，从第一个结点的值开始，每次乘10后与下一个结点值相加，得到两个整数，相乘得到结果。

1. **相反的顺序存储**



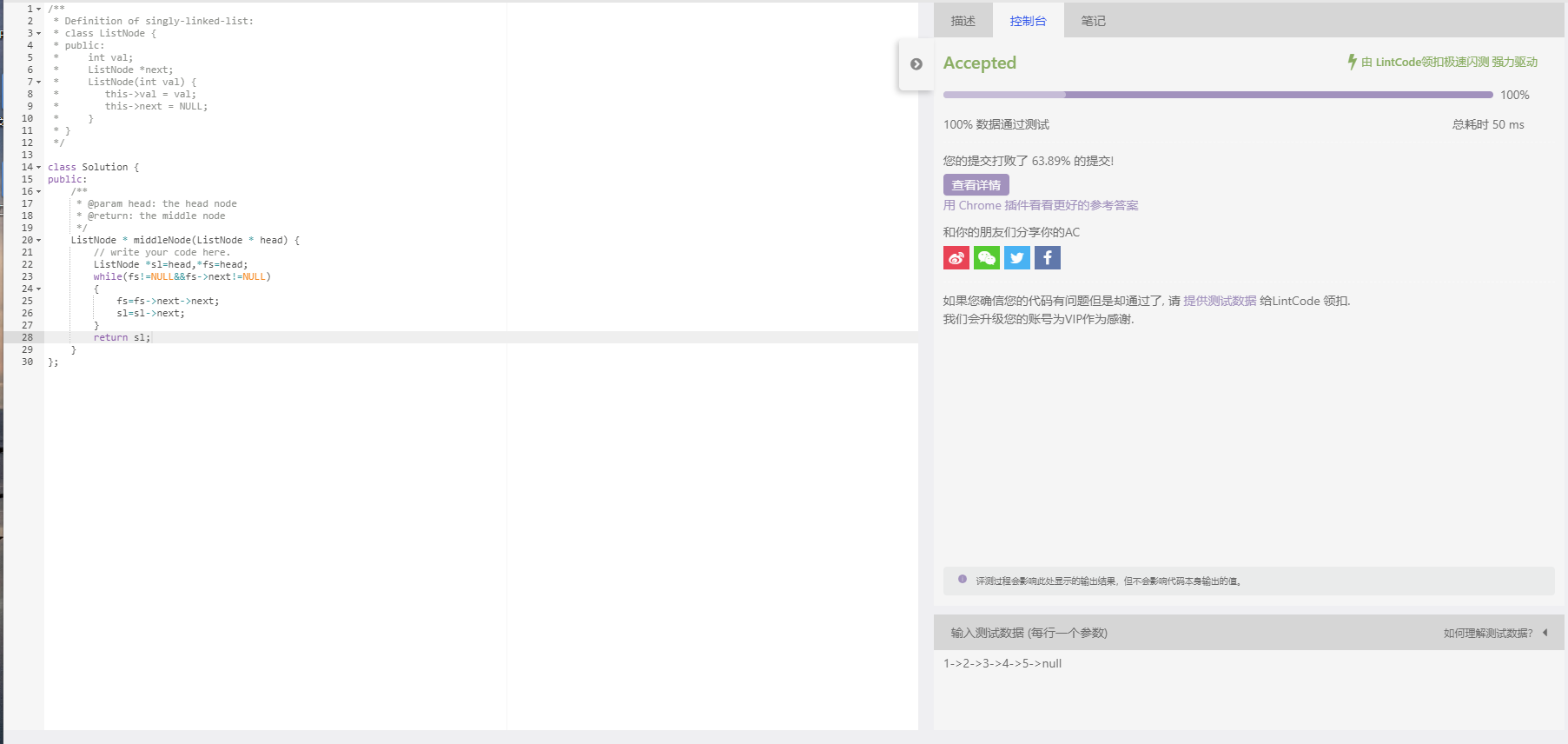


**实现思路：**

利用递归，当到达最后的空结点时逐步返回，从而将链表的值以倒序存储到容器中。

1. **链表的中间结点**



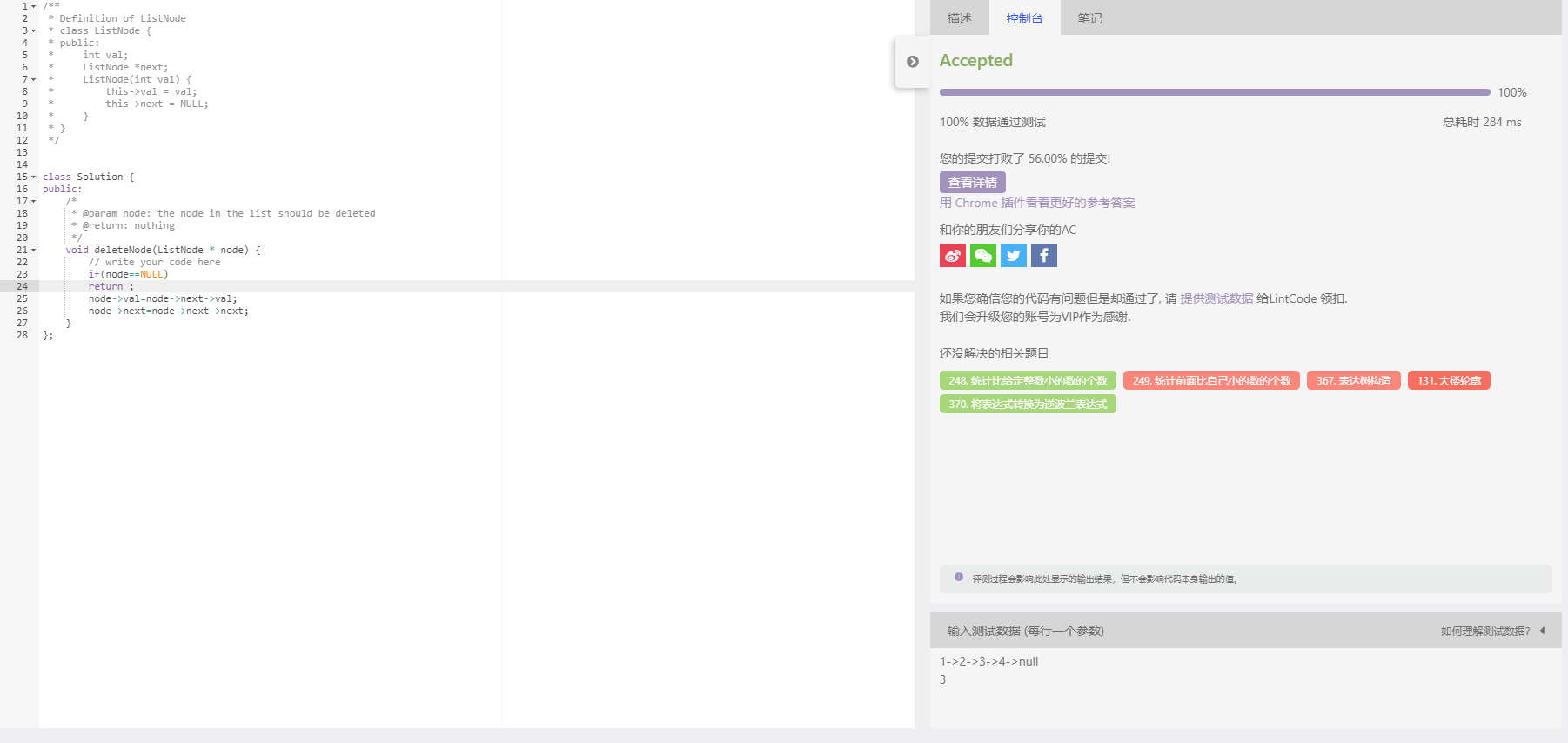


**实现思路：**

利用两个快慢指针，快指针的步数一直是慢指针的2倍，因此当快指针走到尾结点或者空结点时，慢结点为链表中间结点。

1. **在O(1)时间复杂度删除链表节点**





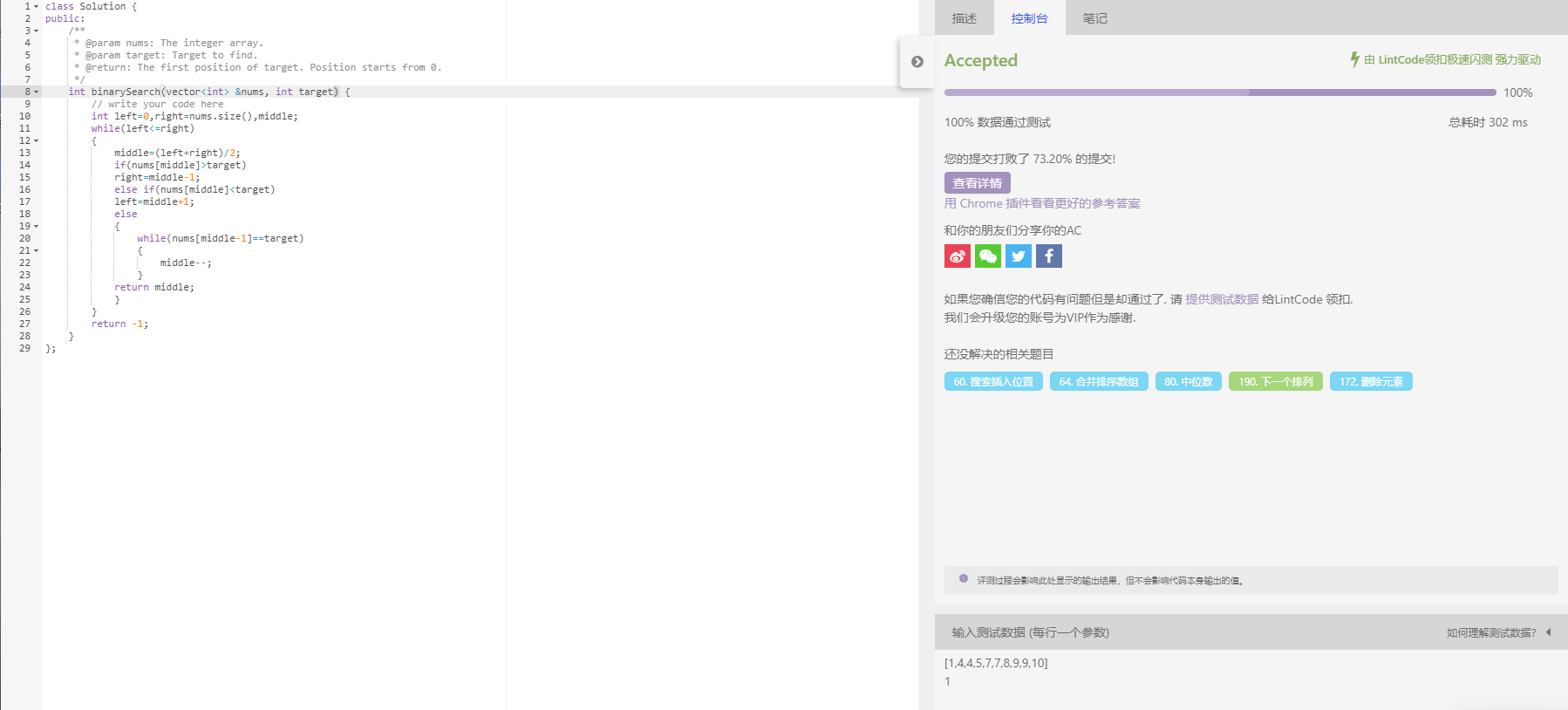
**实现思路：**

当链表为空时，直接返回。否则将下一个结点的值赋给当前结点，当前结点的next指向下一结点的next。

**数组**

1. **二分查找**



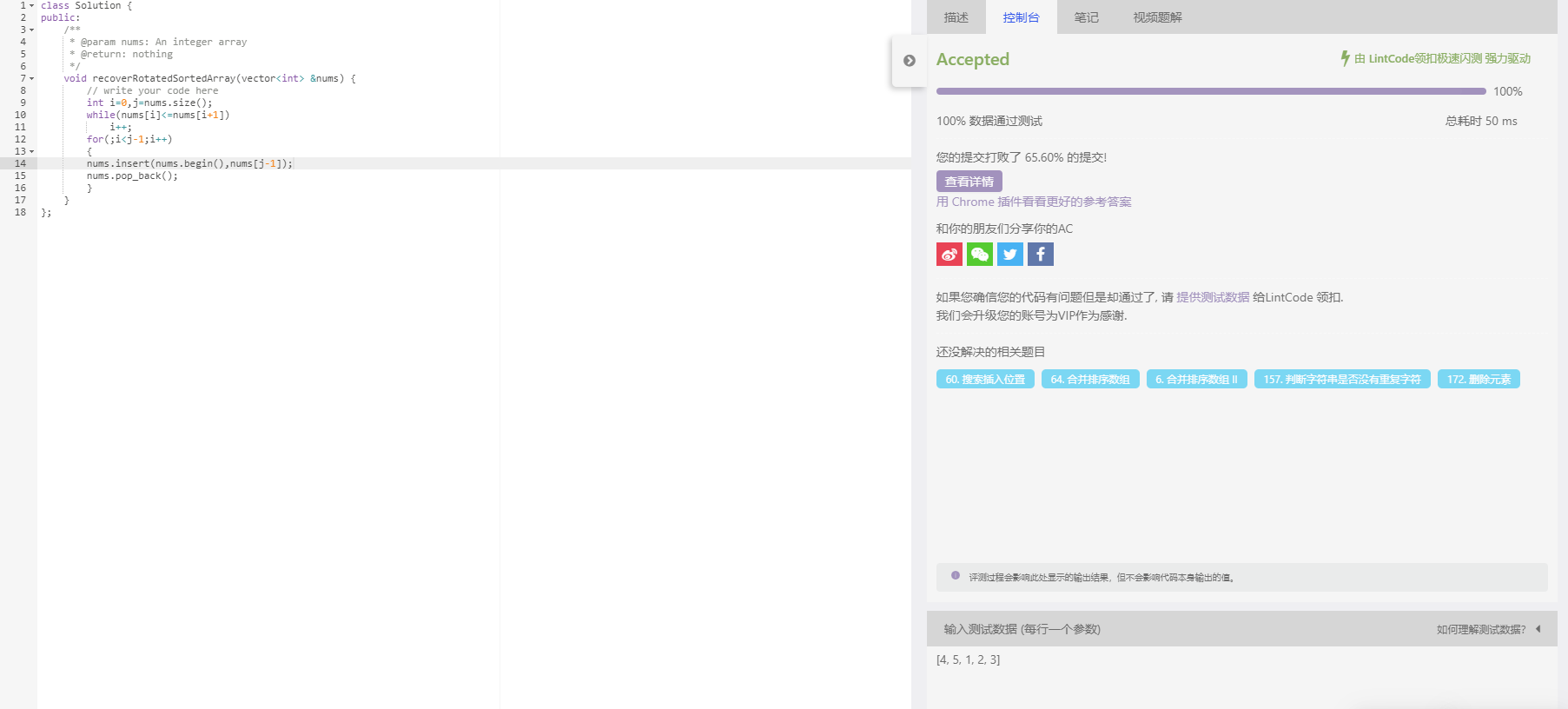


**实现思路：**

利用二分查找，当中点值与给定数相等时，判断是否是该值第一次出现的位置，是则返回该处下标，否则向前移。

1. **恢复旋转排序数组**



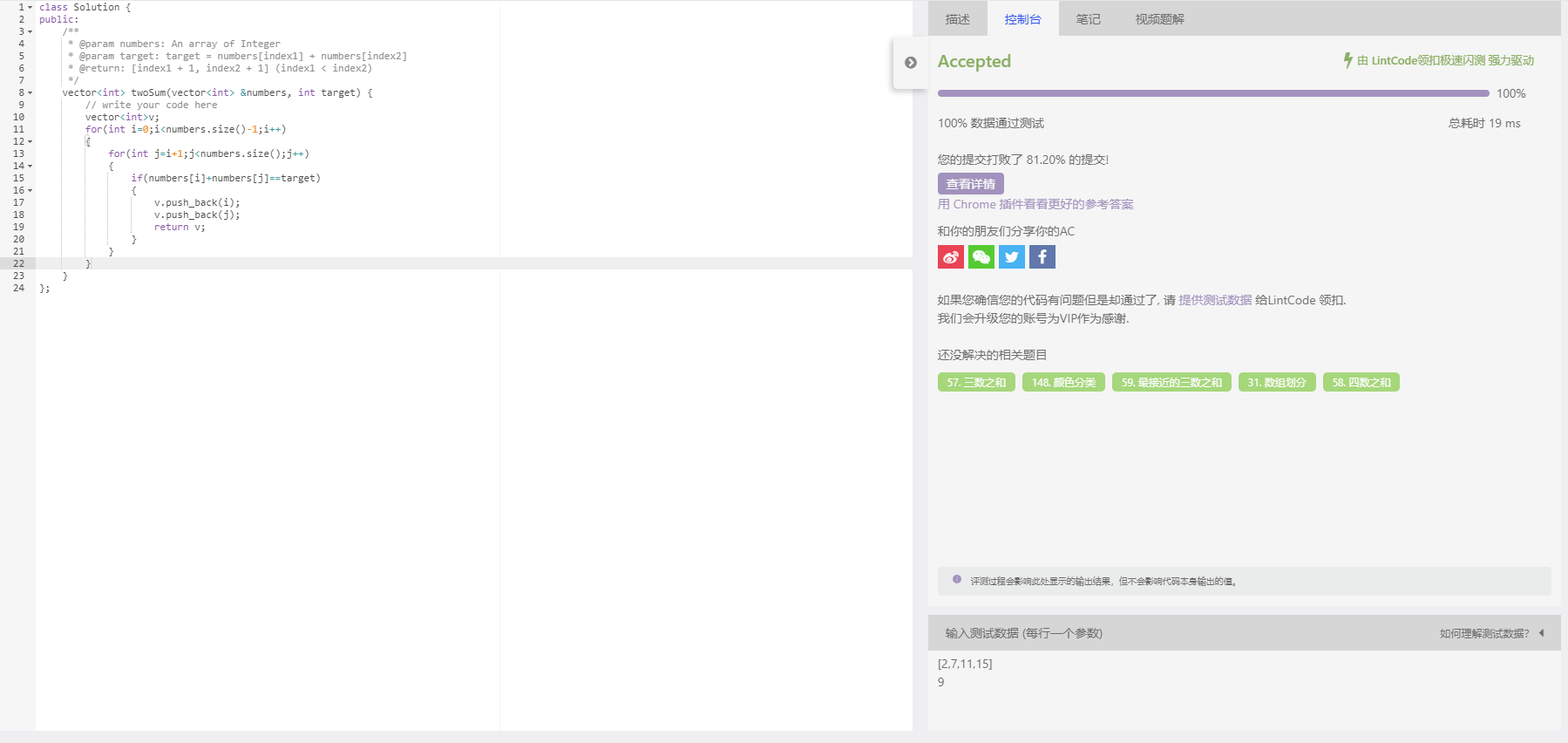


**实现思路：**

首先找到数组中最大元素出现的位置，然后将数组尾端的元素依次插入到数组始端，并移除末尾元素，循环直到数组顺序。

1. **两数之和**



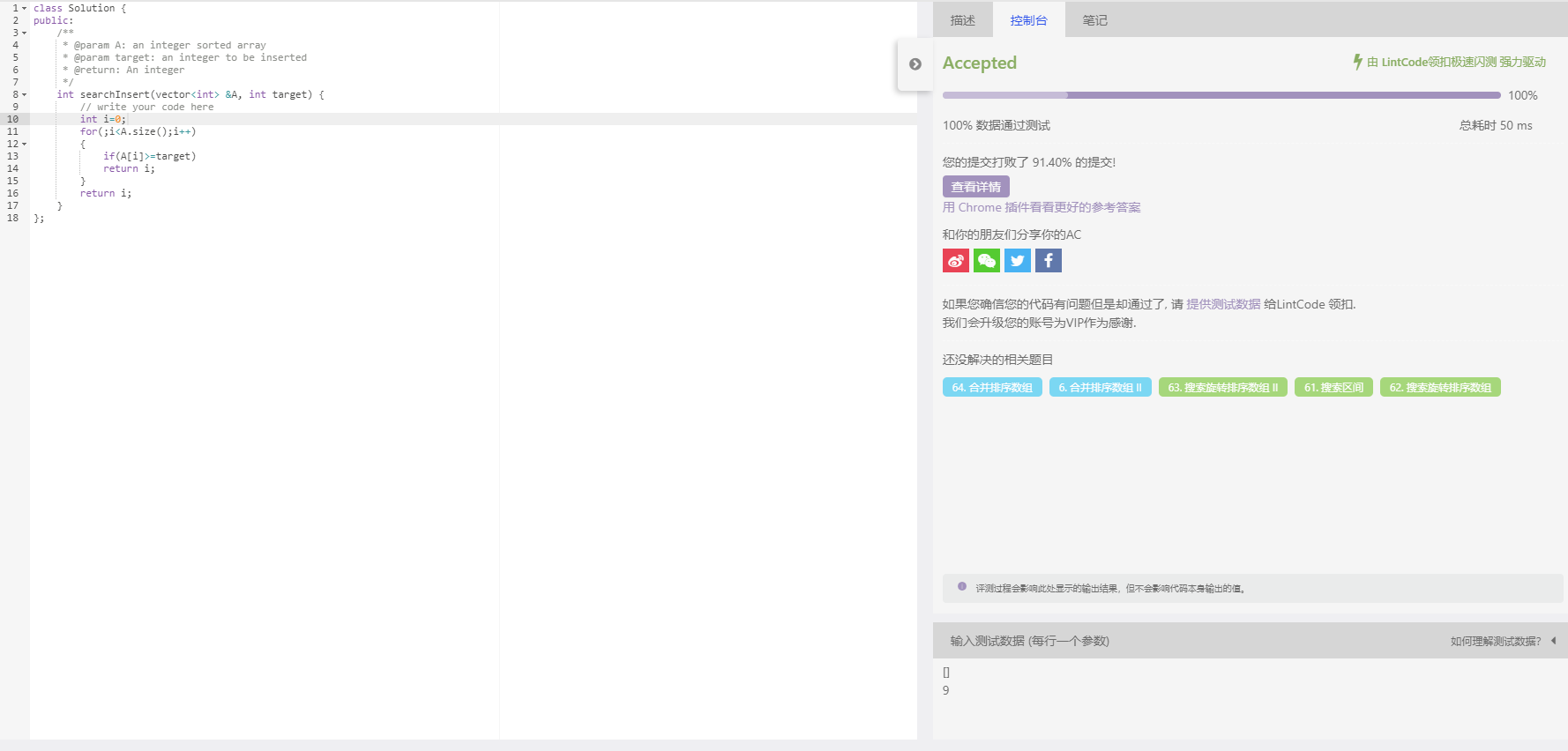


**实现思路：**

利用双层循环判断数组前后两数相加是否等于给定数，找到则记录两个数的下标，返回。

**60. 搜索插入位置**



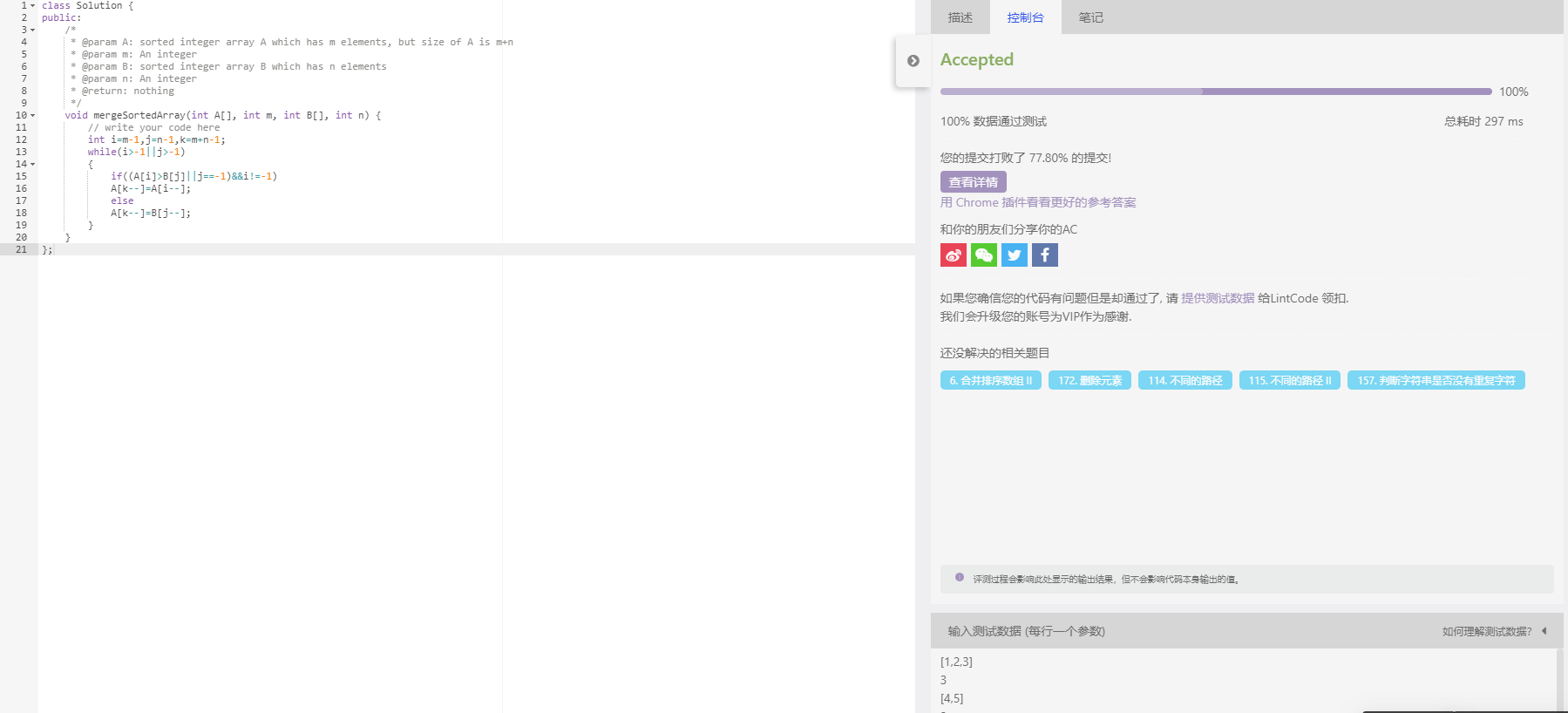


**实现思路：**

遍历数组，当前值大于给定数时，返回当前值的下标。给定值大于数组所有值时，返回数组大小为下标。

1. **合并排序数组**



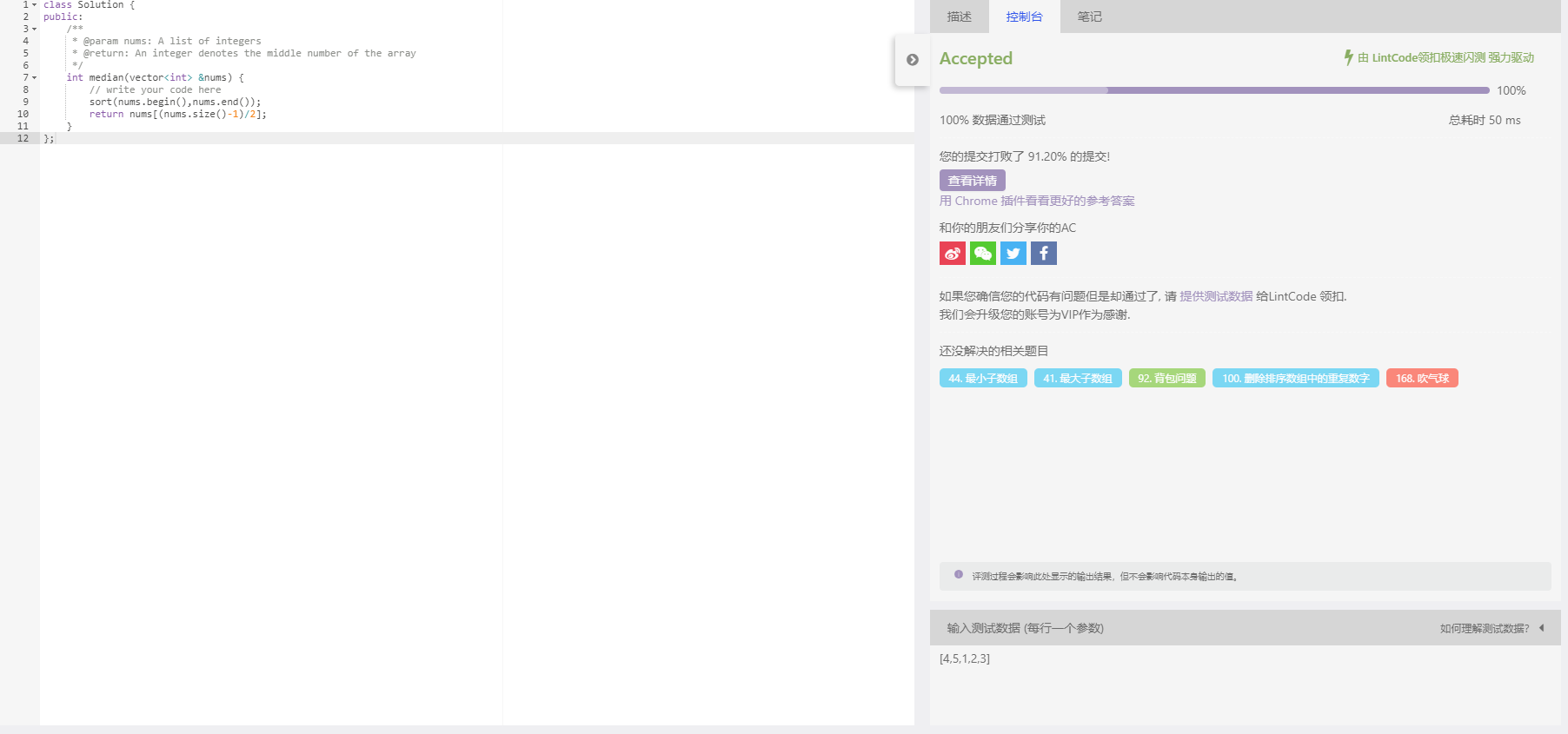


**实现思路：**

反序比较两个数组，当A数组值大于B数组值时，将A数组的值移到尾端，否则将B数组值移到A数组尾端，直到访问两数组所有元素。

**80. 中位数**



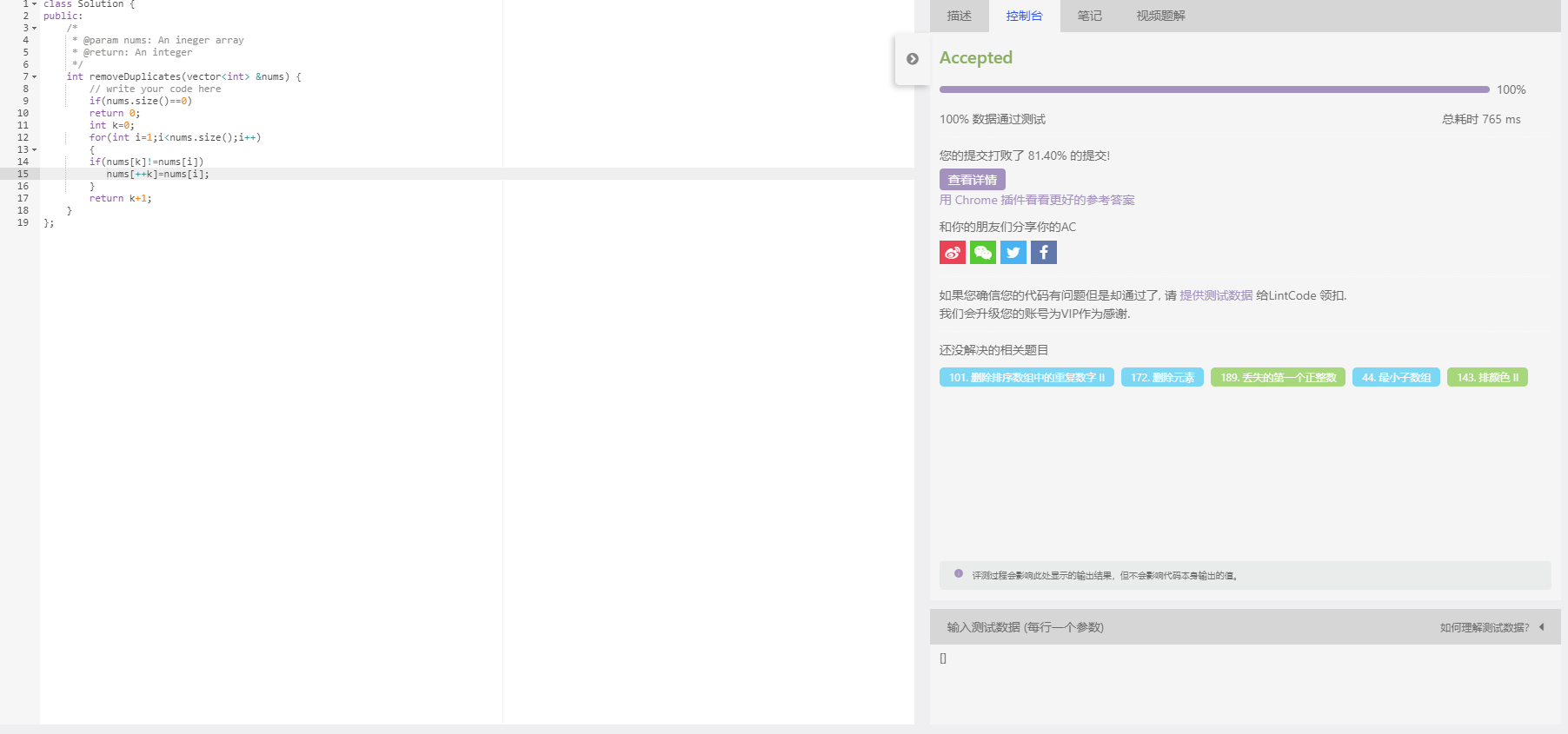


**实现思路：**

先用sort函数对数组进行排序，然后无论数组的个数为偶数还是奇数，都是返回数组下标为(nums.size()-1)/2的数。

1. **删除排序数组中的重复数字**



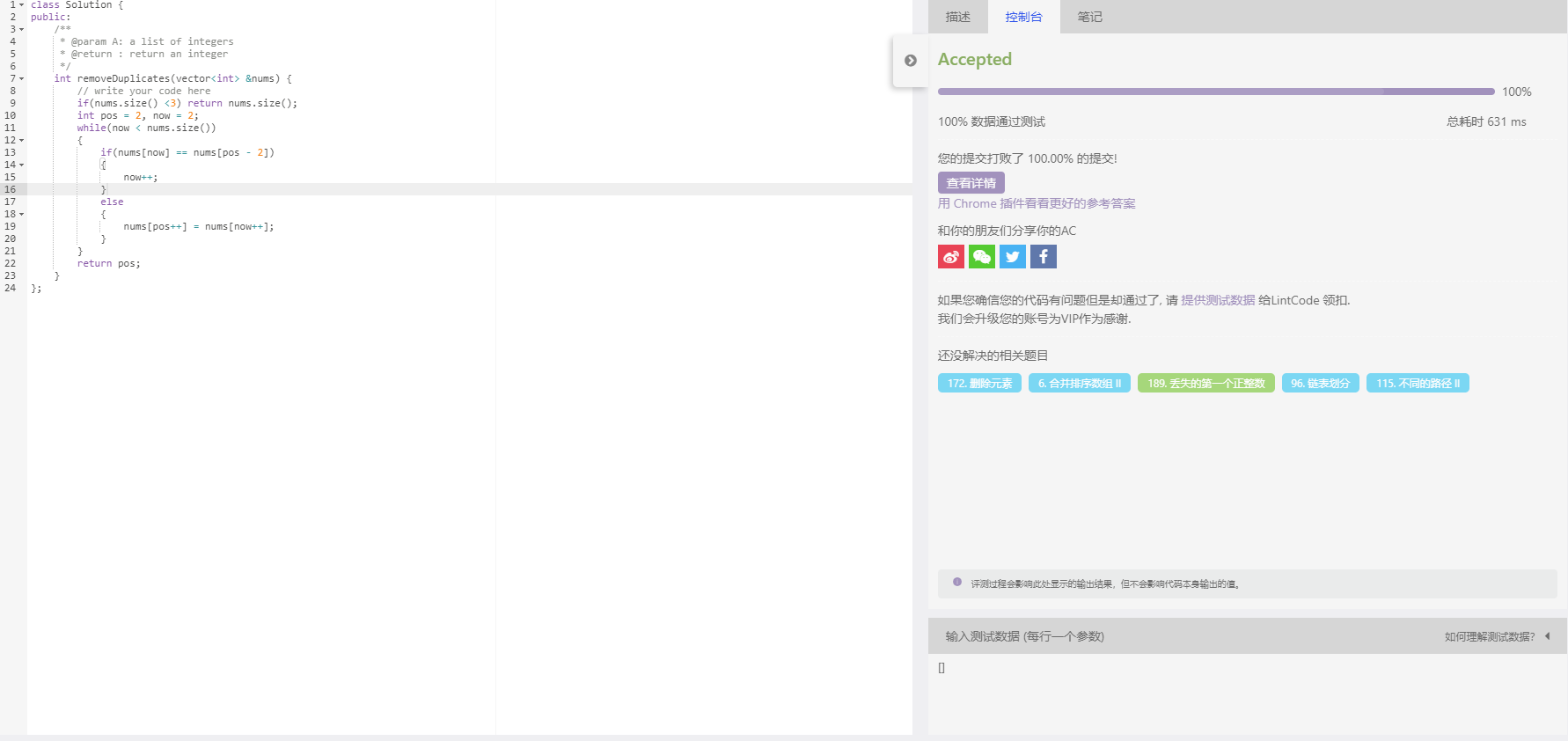


**实现思路：**

利用快慢指针原理，跳过相等元素，将相等元素中的第一个放到最前面，返回数组长度。

1. **删除排序数组中的重复数字 II**



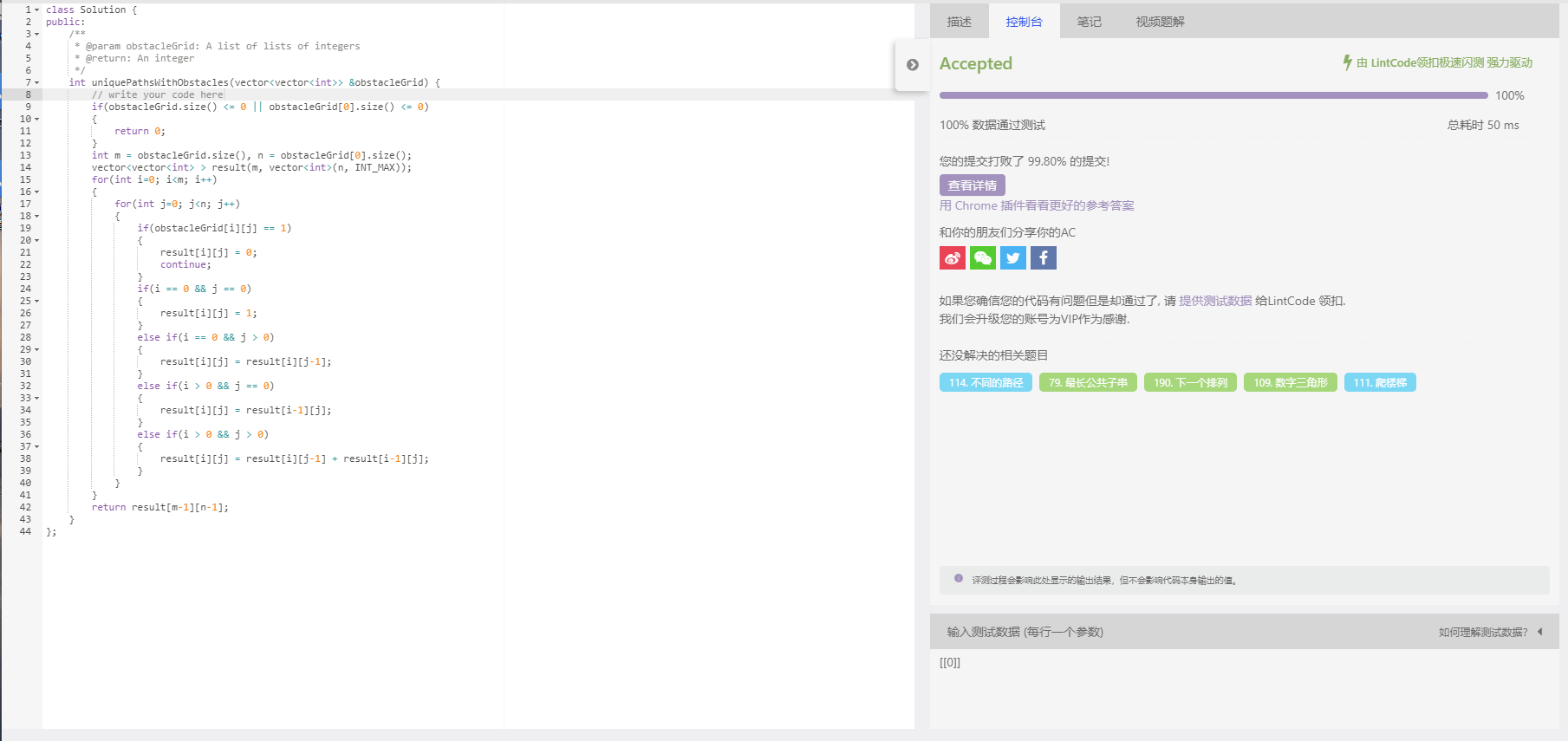


**实现思路：**

数组长度小于3时直接返回数组长度，否则从下标为2的数开始循环，判断当前值与其前两位的值是否相等，相等则后移，否则将其后的值前移，返回条件终止时的位置为数组长度。

1. **不同的路径 II**





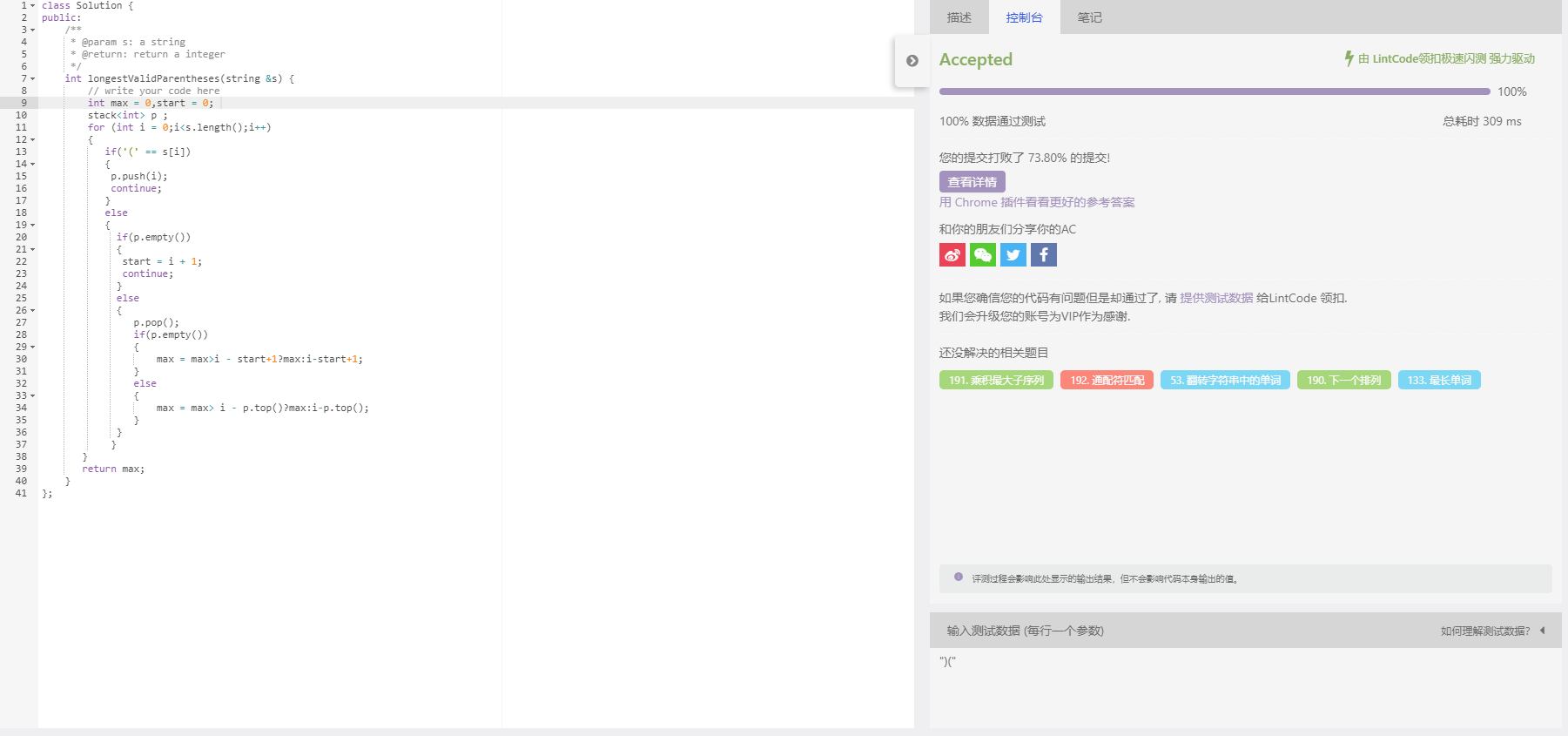
**实现思路：**

首先判断网格的大小是否有效，其次创建一个新二维容器，遍历网格数组，若遇到障碍物则障碍物所在的行与列都不能走，进行标记为0，否则计算达到当前位置的有效路径数，逐步累加，最后返回。

**栈**

**193. 最长有效括号**





**实现思路：**

利用栈的后进先出对括号进行最近匹配的原则，若为左括号，则入栈，否则为右括号判断栈是否为空，为空则改变起始位置，不为空则移除栈顶元素，再判断栈是否为空，选择合法子串的长度，循环直到遍历字符串。