****

软工暑期

算法训练报告

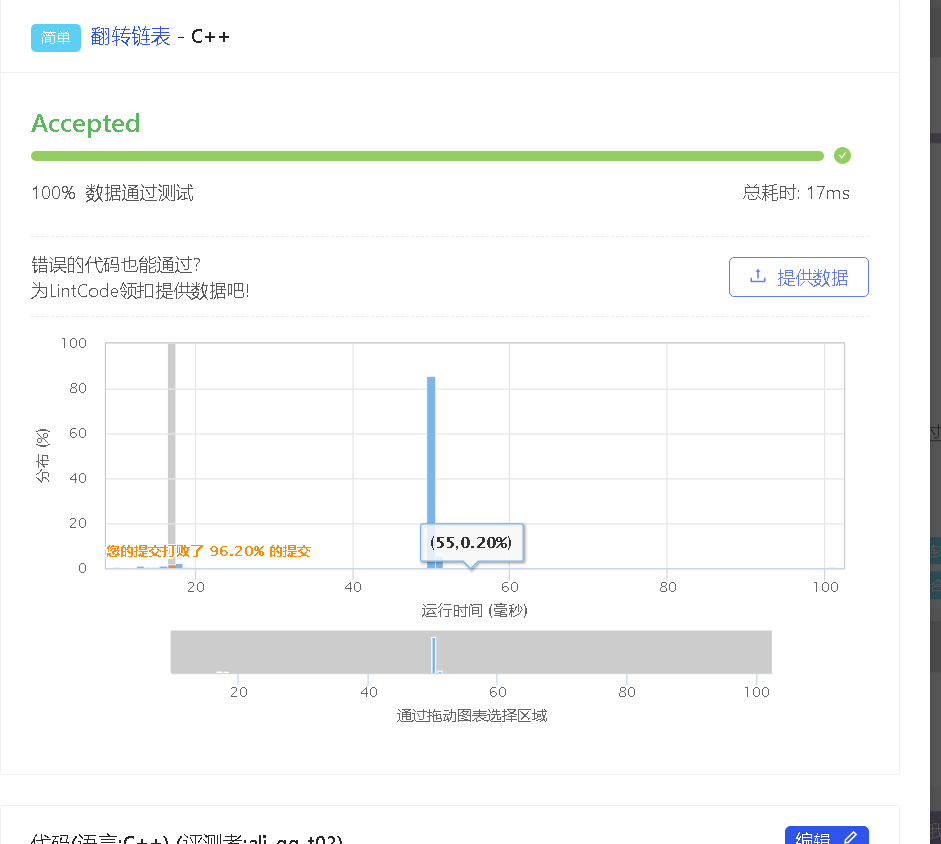
专业班级： 11J192

学 号： 20191000719

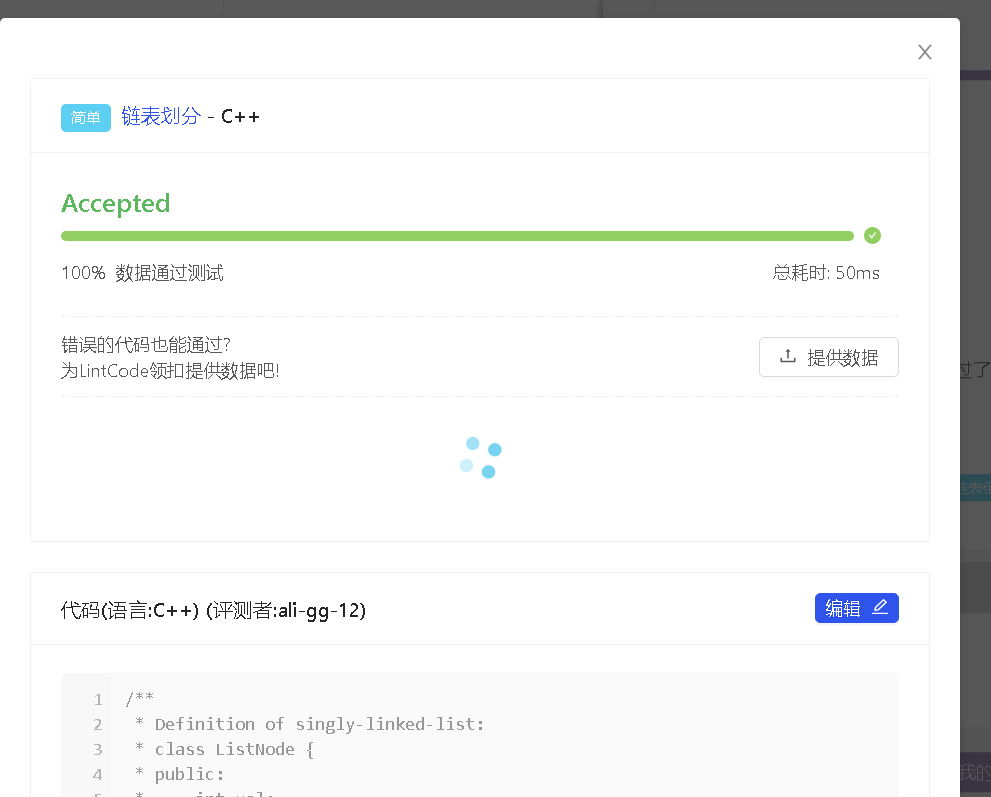
姓 名： 刘世龙

**中国地质大学地理与信息工程学院**

**2020年 7 月**



先设置一个空结点，用它作为中间变量。设置的temp是一个可移动的指针，用它来完成循环后的挪到下一结点。循环第一次，把开始结点变成了尾结点，循环第二次，把temp指向了下一个结点，把下一结点的指针指向了上一结点，完成新的链接，以此类推，直到指向了空，循环结束。完成了链表翻转。



先找到比给定的val大和小的结点，找到之后，比给定值小的划分到left的后面，比给定值大的结点划分到right的后面，再把两个链表还有给定的结点合在一起。



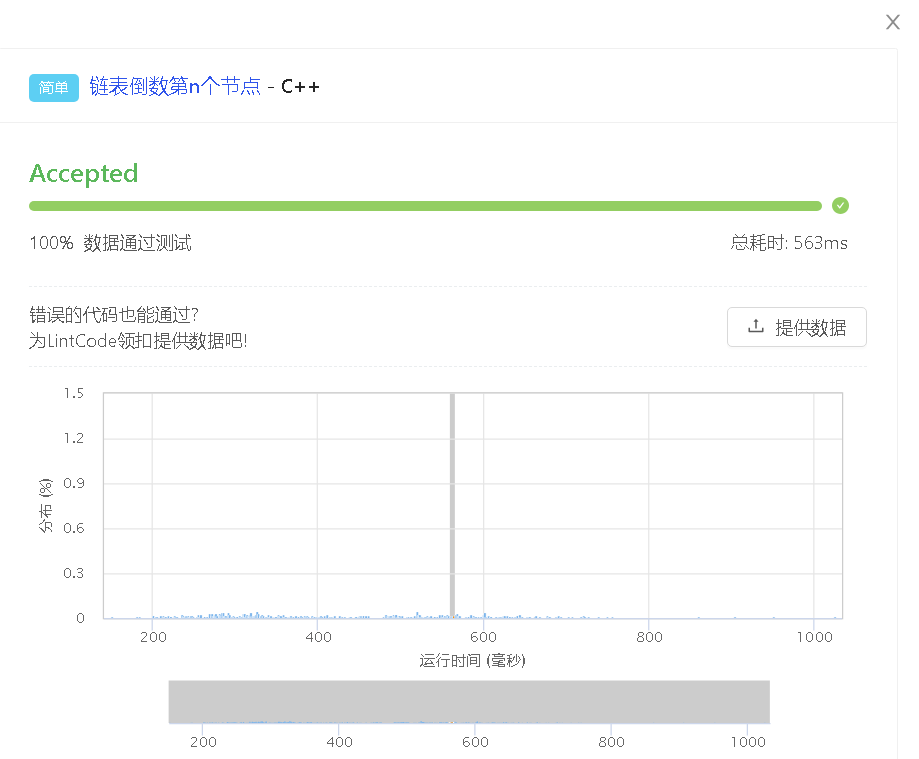
二重循环遍历每一种情况

外层循环控制每个节点的处理

内层循环是从这个节点的下一个节点开始的遍历判断，如果相同则删除



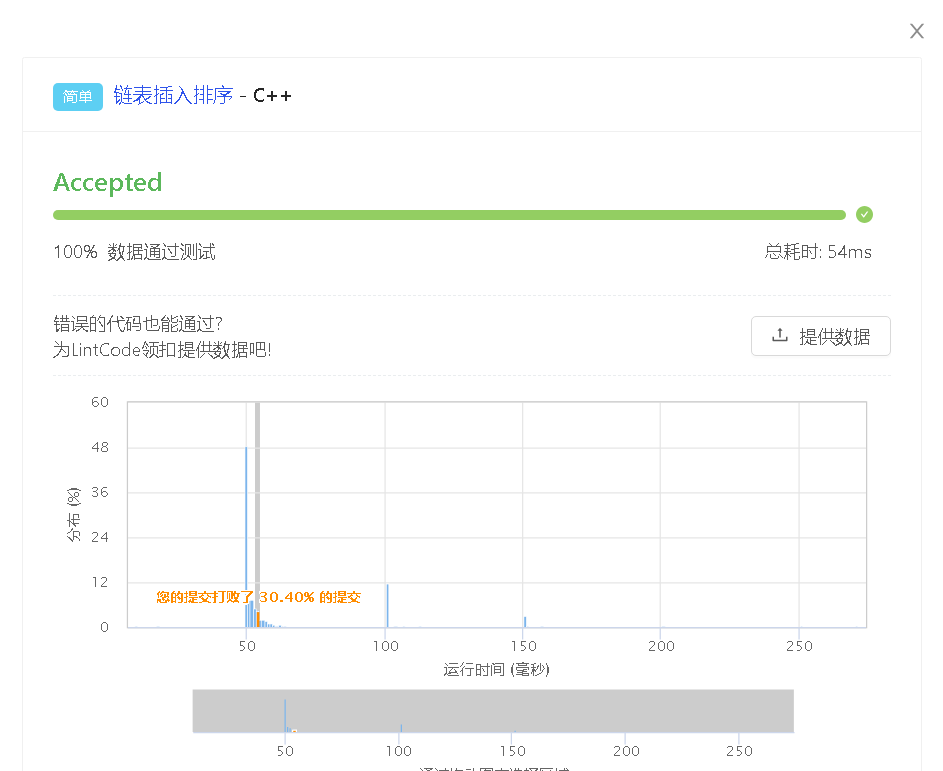
两个链表都为有序，则每次比较两个链表头元素，取小的加入合并链表的新一个节点。



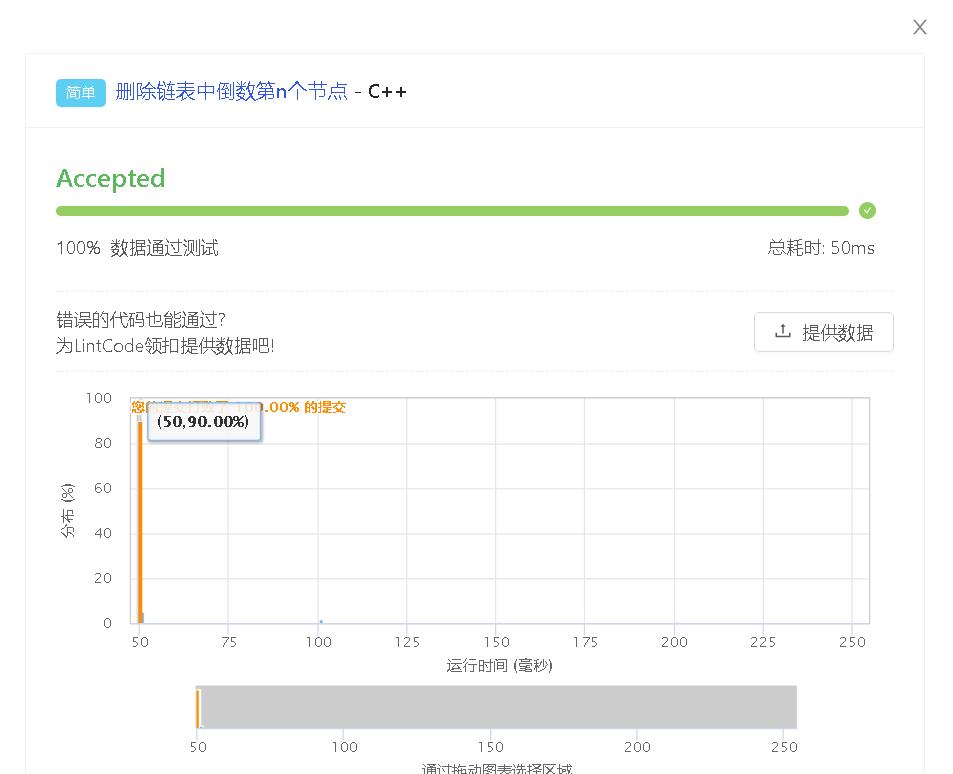
创建两个指向head的指针p q,让p遍历，p先开始移动，p走到第n-1个节点是，之后p q 一起往后移动，这时候当p指向最后一个节点的时候，q就指向了倒数第n个节点的位置，这时候返回q就可以了



遍历两个链表知道每个位的值，两个值加上前一位进位值（0或者1）模10就是该位的值，除以10就是向高位的进位值（0或者1）。由于两个链表的结点可以长度不同，所以要处理一旦为null，该位的值就要变成0。



创建一个链表，从head链表中依次向链表中按插入排序规则添加节点。每次节点都需要遍历到最后一个小于head节点的值位置，这样head节点插入进来就保持了排序



1.先用一个节点dummy保存head节点的前一节点，最后返回dummy.next

2.用一个循环让head节点右移n个节点

3.用一和dummy同指向的节点preDelete，和head节点同步移动，直到head移动到null(尾节点.next)

4.用preDelete.next=preDelete.next.next删除该节点



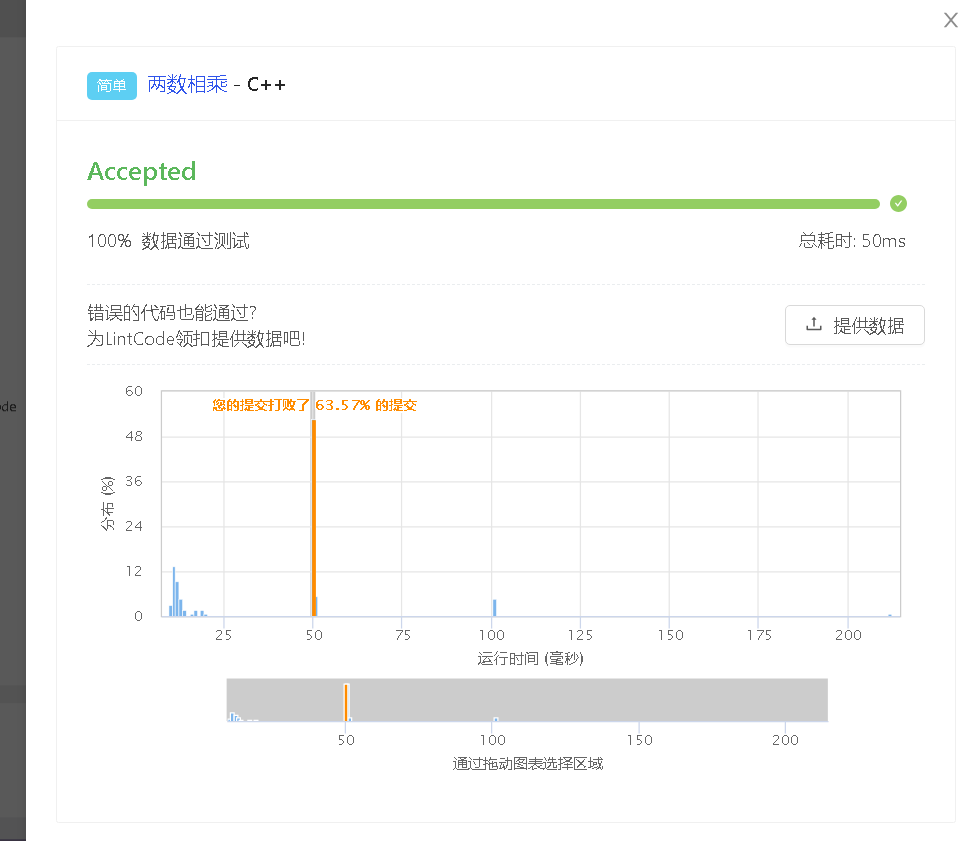
利用双队列，将根结点加入a队列，b队列则存放a队列的后的节点，生成第一层的链表，再将b队列的孩子放入a队列，b队列依次出队得到第二层链表，以此类推，得到最后一层。



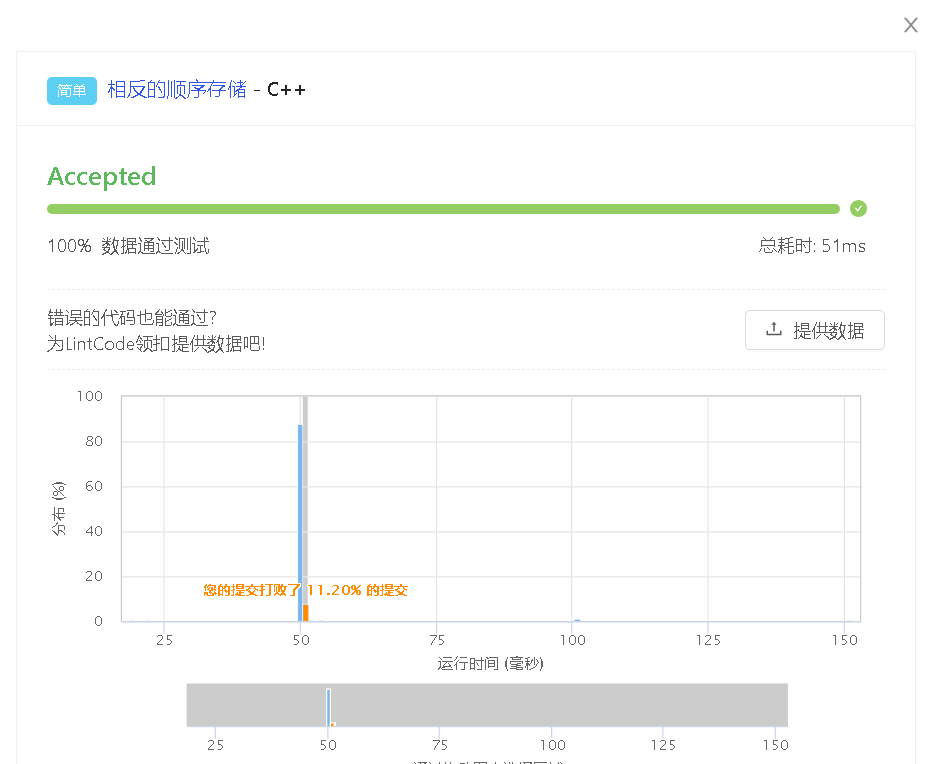
1.判断head是否为空

2.要删除的节点在头部

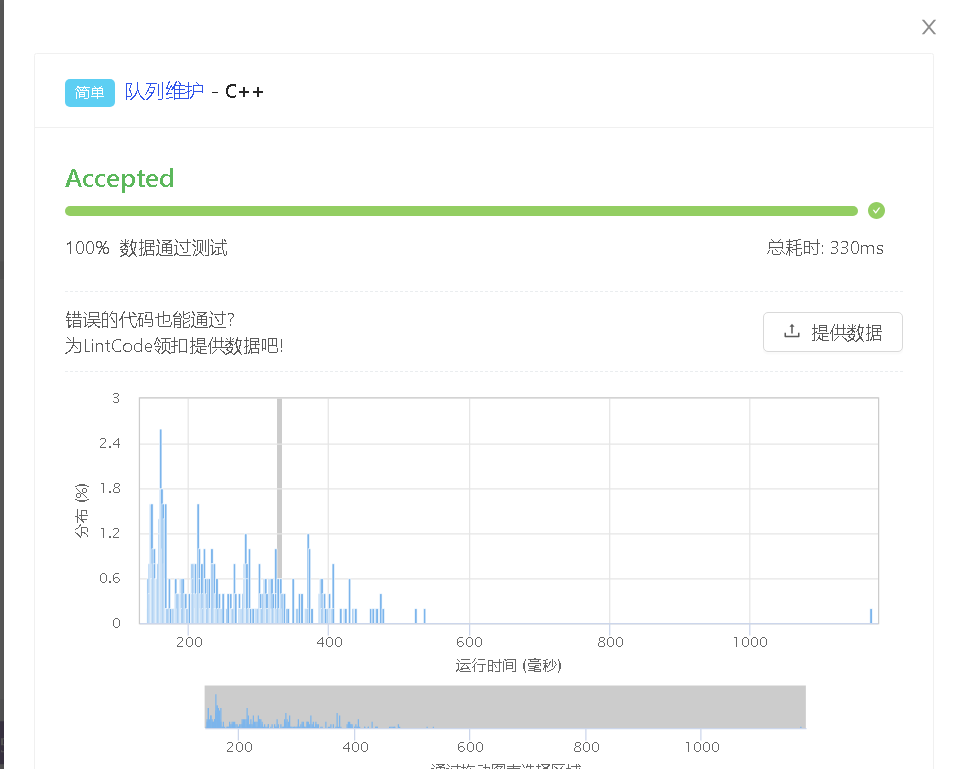
3.要删除的节点在中间、尾部，遍历链表删除



给出两个链表形式表示的数字,写一个函数得到这两个链表相乘乘积。



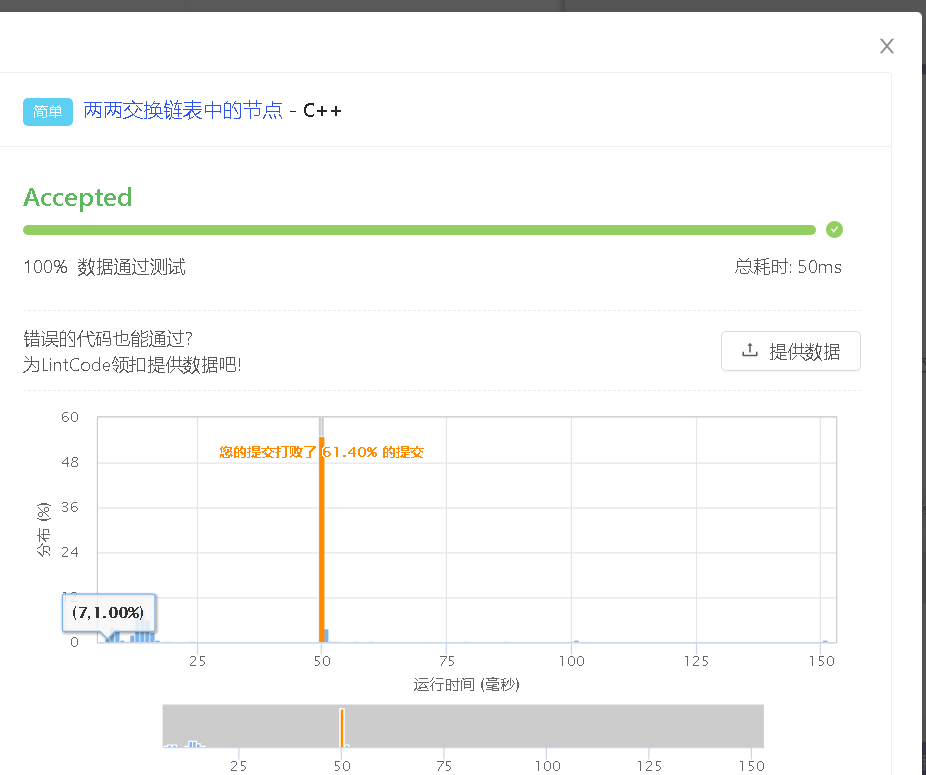
将List元素依次压入栈中,再将List元素依次压入栈中



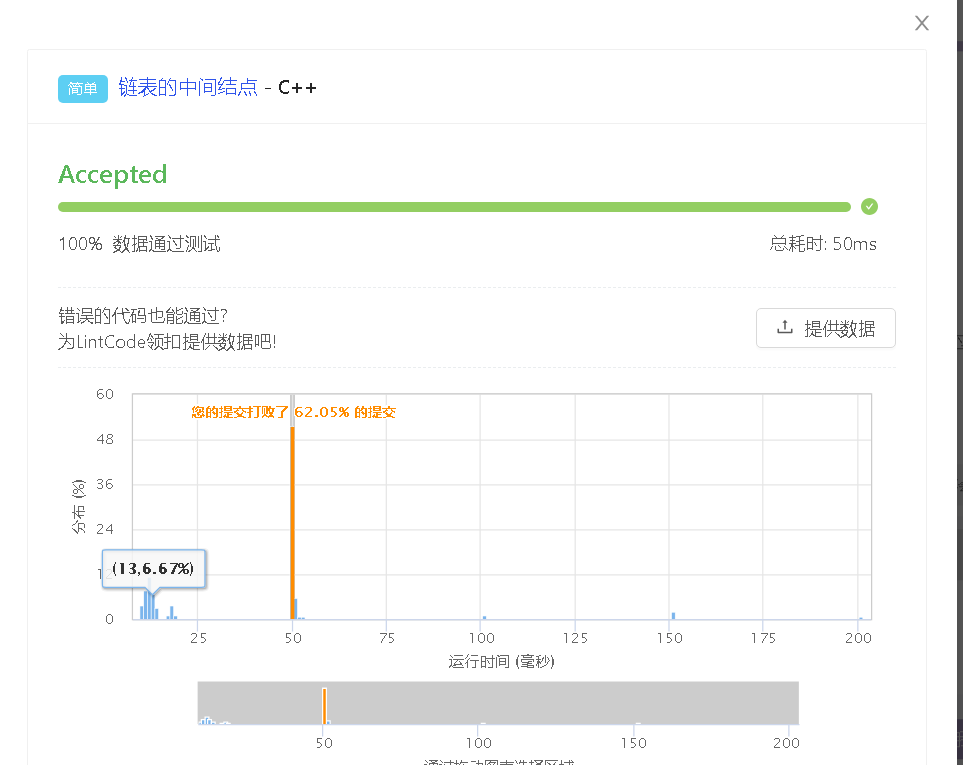
1.enqueue(item).将新元素放入队列中。

2.dequeue(). 将第一个元素移出队列，返回它。

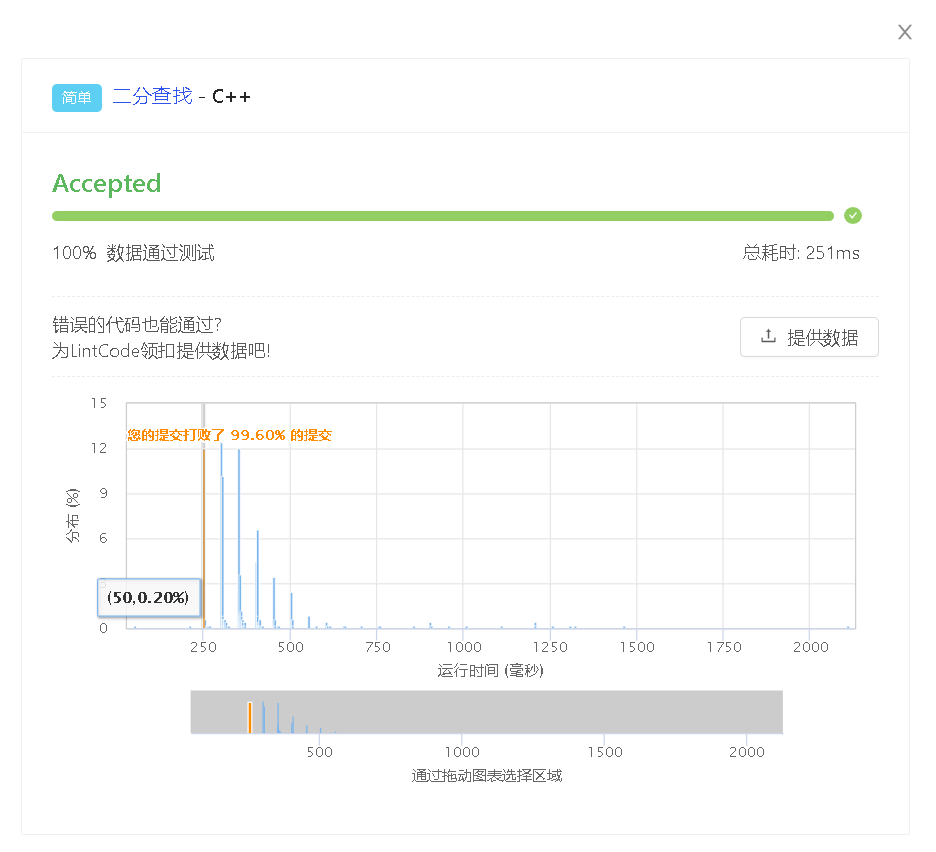


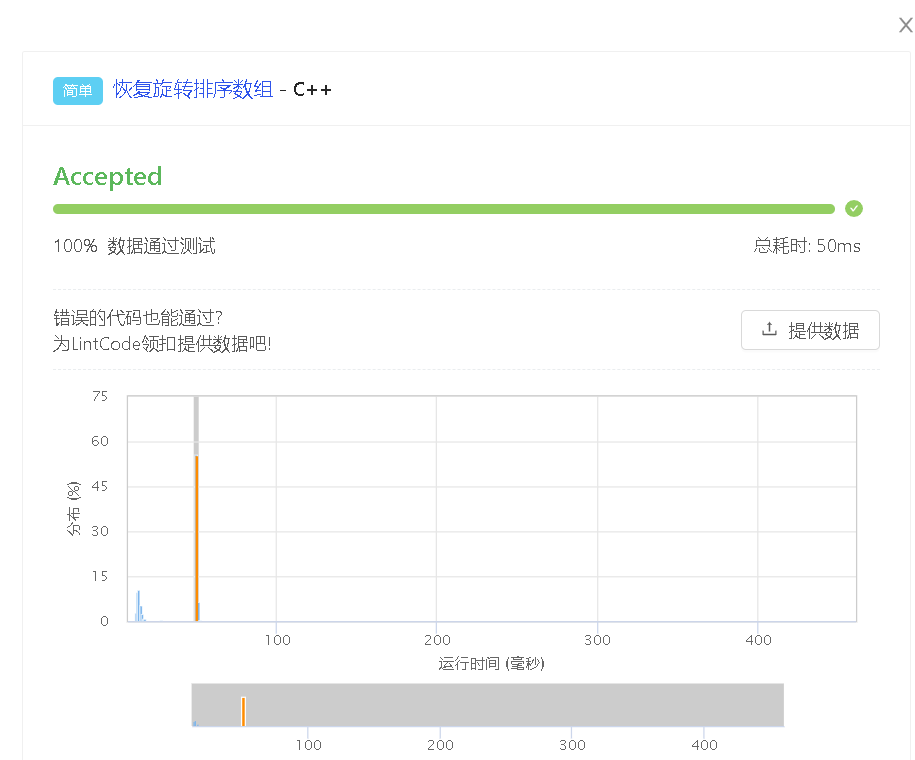


给一个链表，两两交换其中的节点，然后返回交换后的链表。



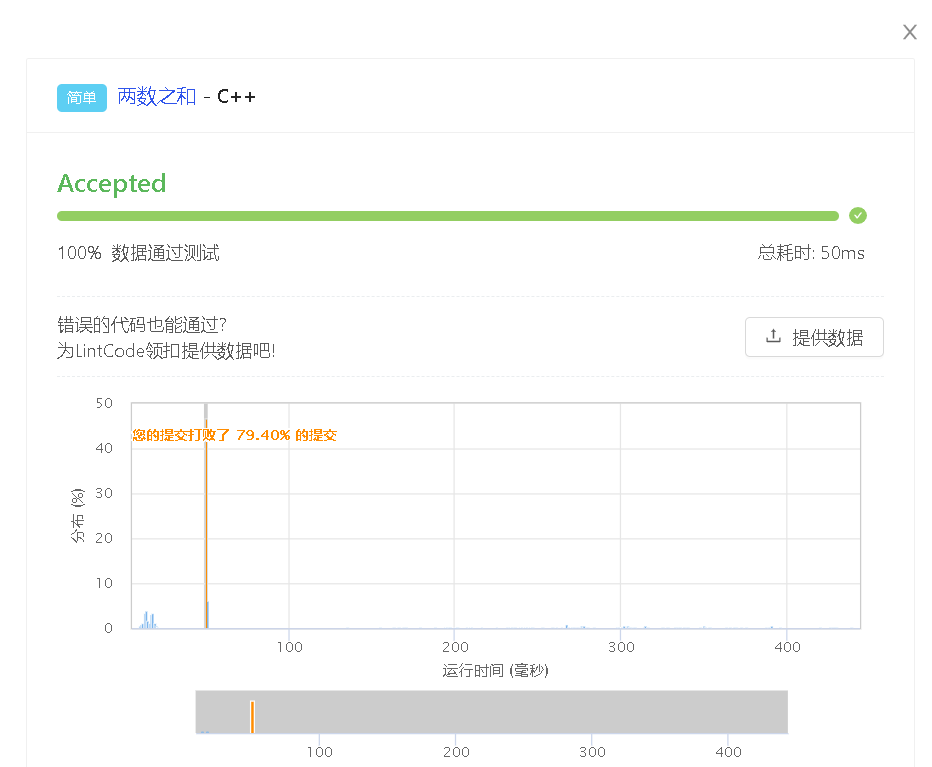
给定一个带有头结点 head 的非空单链表，返回链表的中间结点。如果有两个中间结点，则返回第二个中间结点。



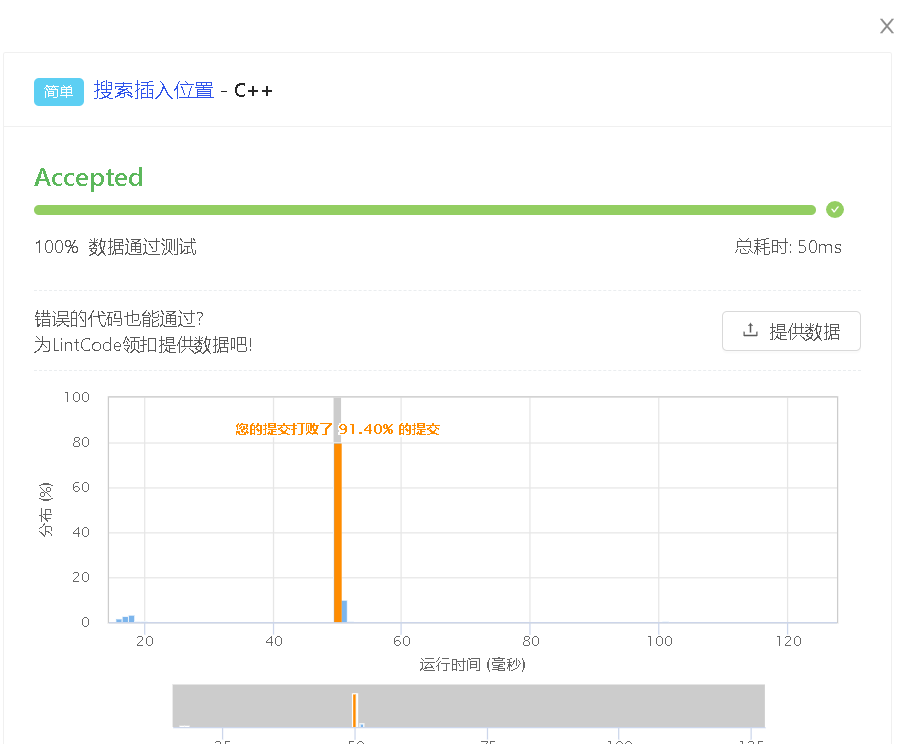


1. 先找到最小数的位置.然后分部分翻转

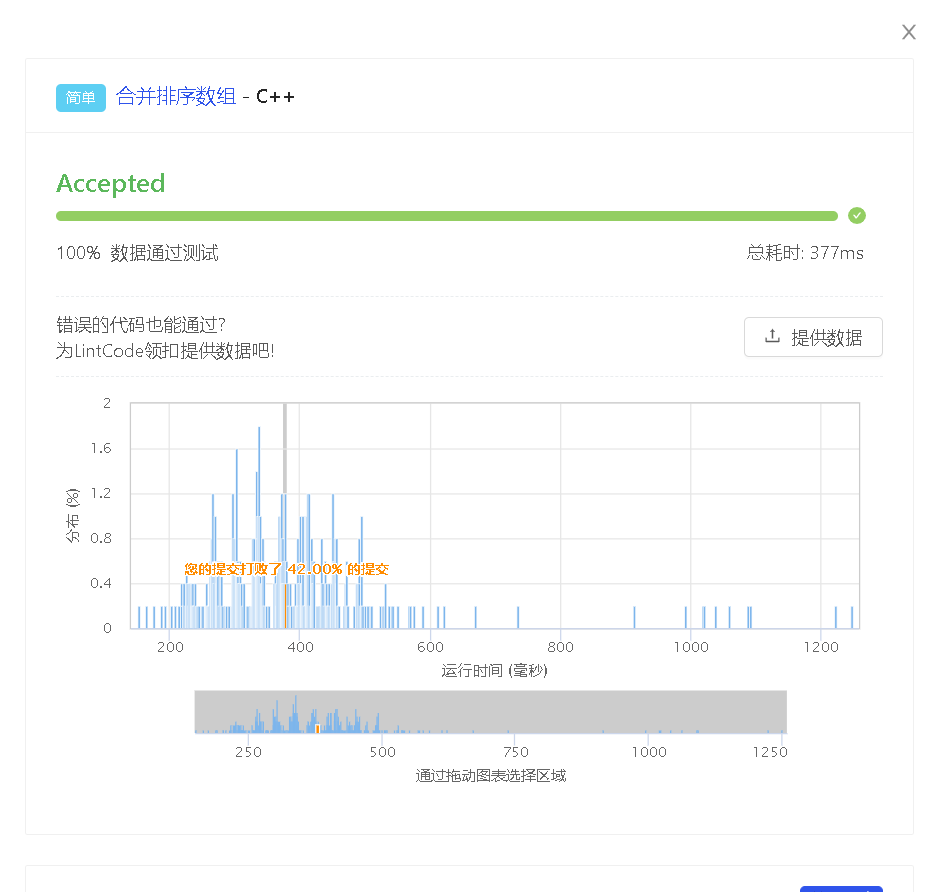
2. 此时数组为倒序, 将其翻转

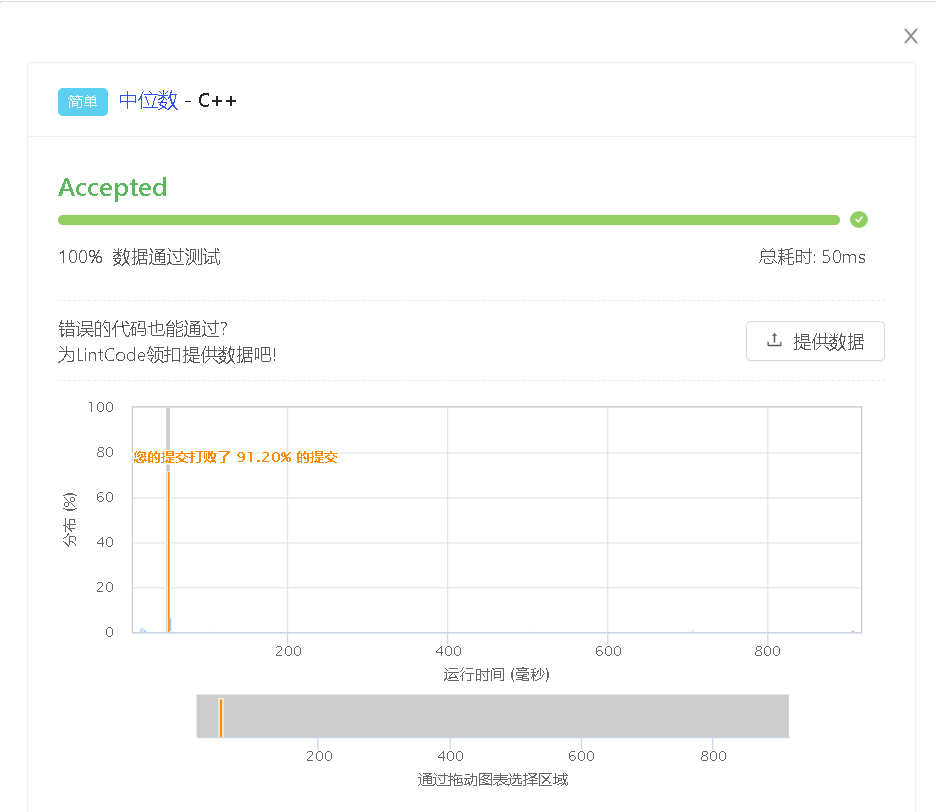


用for循环在数组中寻找两个元素相加等于给定数值的元素，如果相加等于给定target，就将下标赋到新的数组中，返回新数组。注意下标范围是1到n，不是0到n-1。

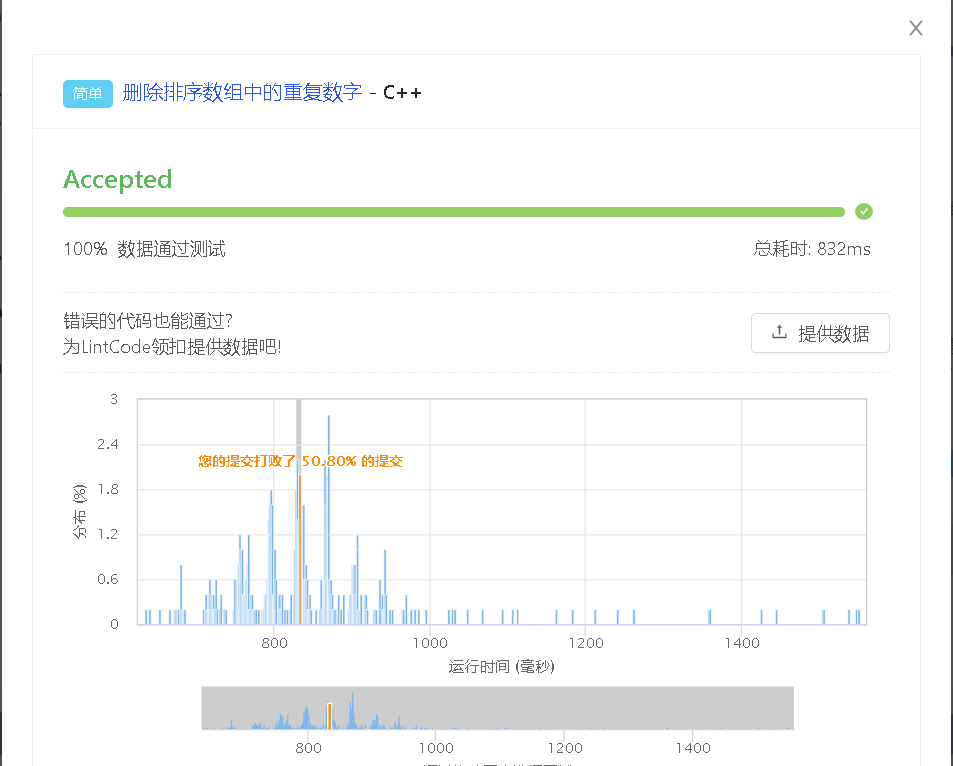


用for循环进行数组与target的比较

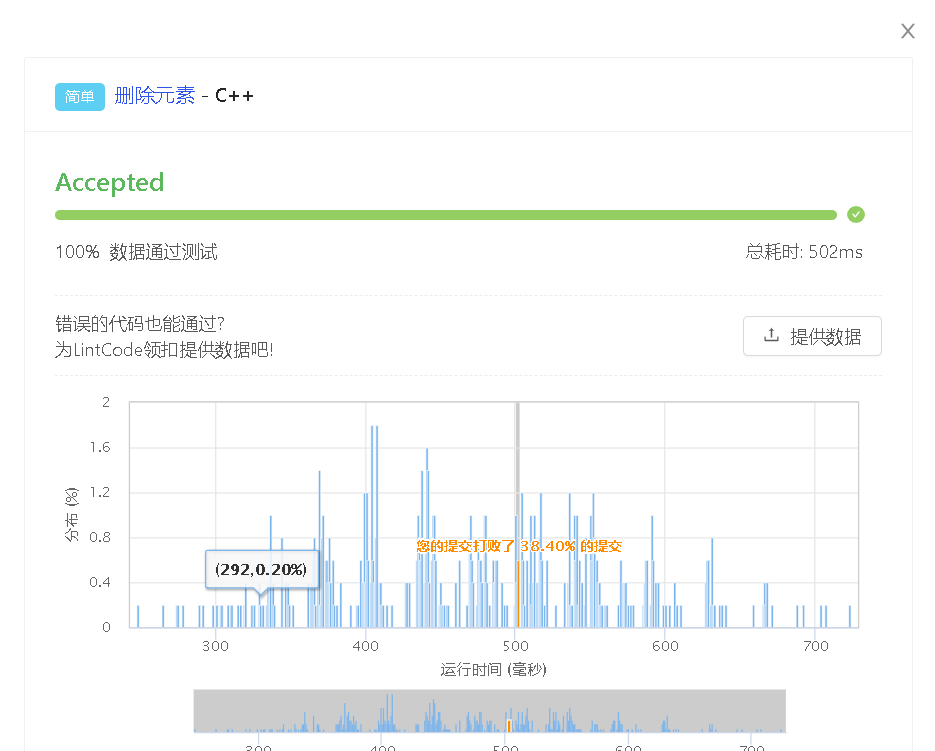


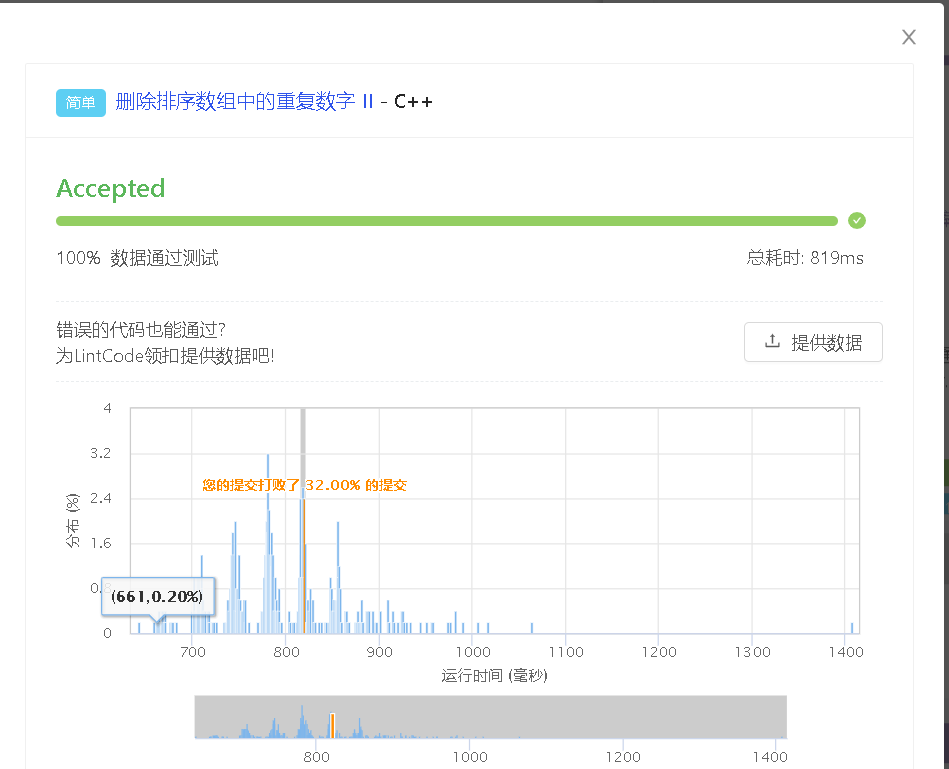


先排序后取中位数

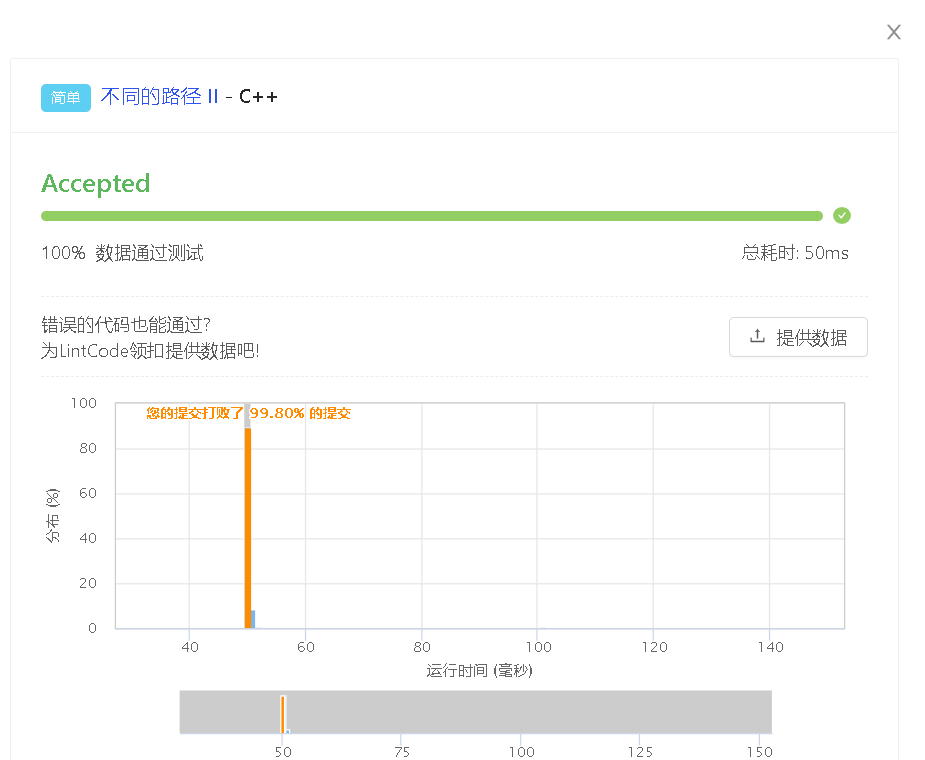


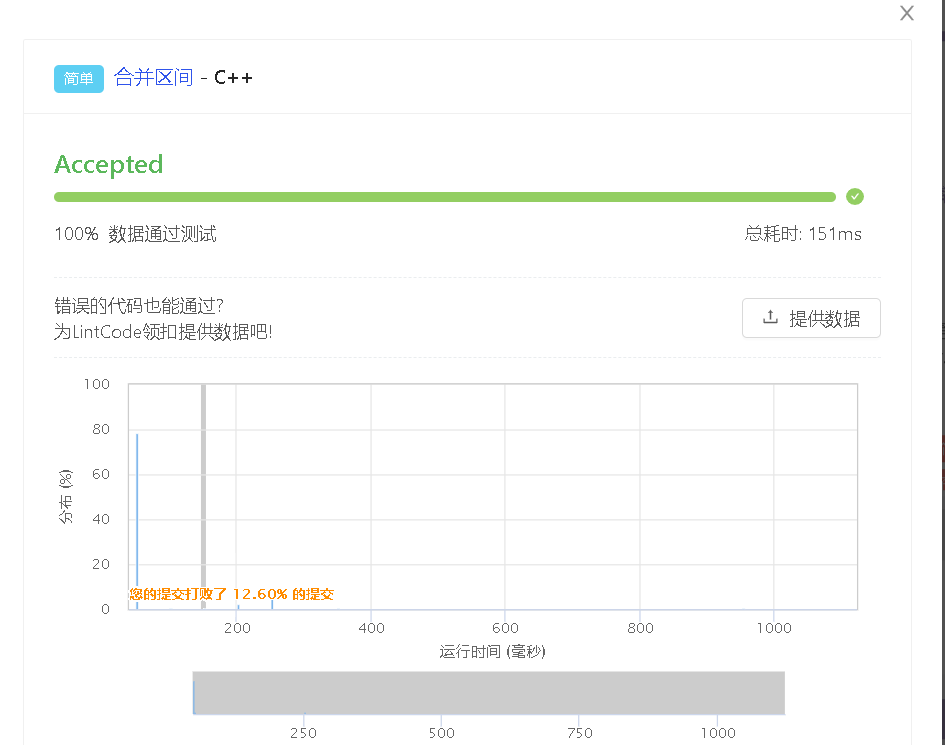
首先判断数组是否为空，若为空则返回0；不为空则用for循环进行覆盖。



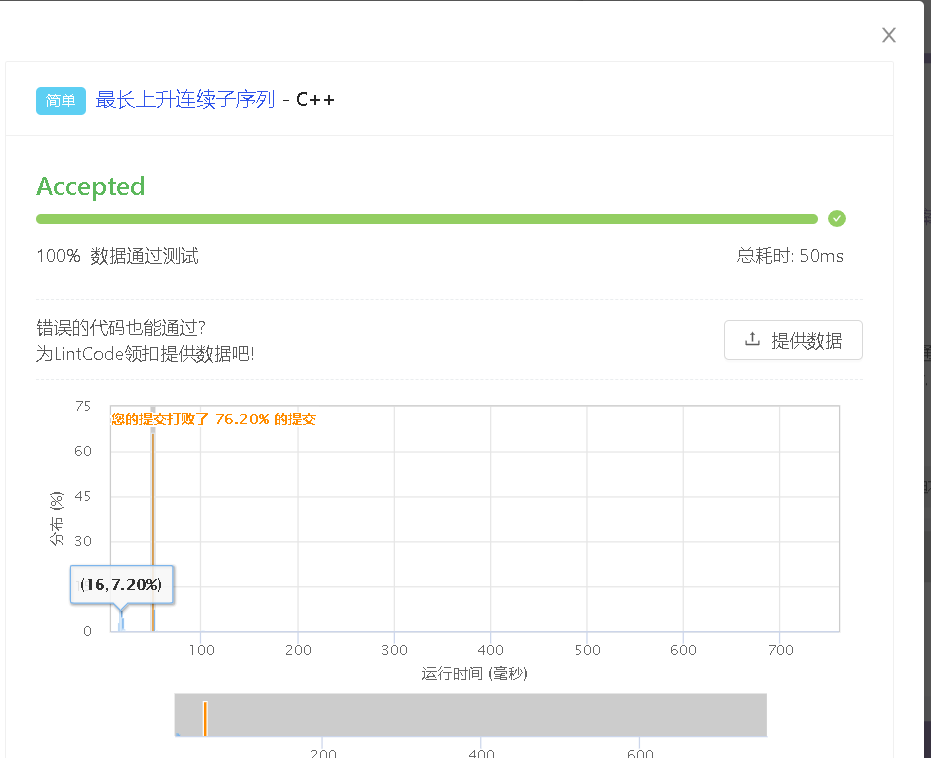


增加一个记录元素出现次数的变量，限制最多出现2次。

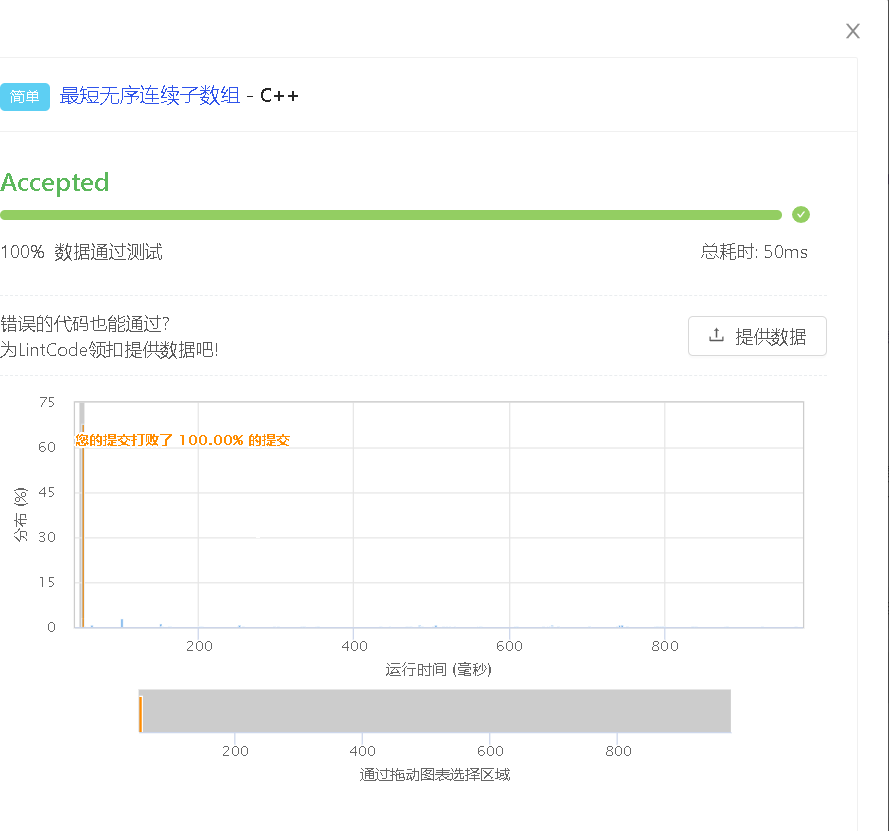




先按升序进行冒泡排序，再逐一进行比较直接在比较过程中进行排序比较



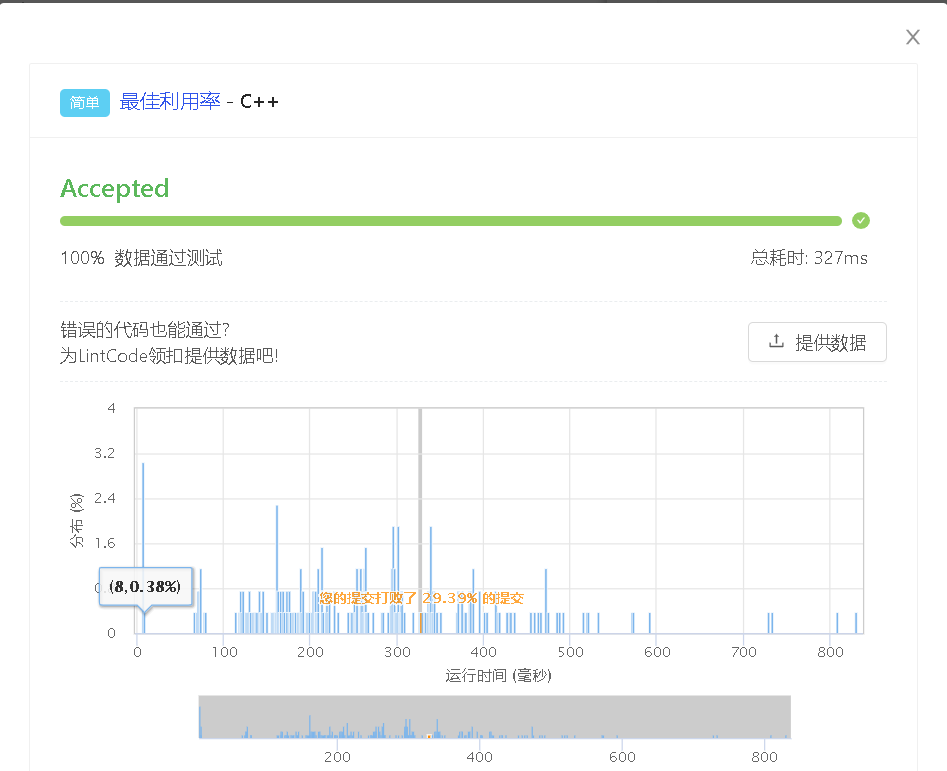
用两个整数分别储存升序的个数和降序个数，再将个数赋予result，如果后续有大于result则替换即可



个节点 存区间里最多出现的数并且出现次数高于区间总数量的一半，不存在则等于-1。查询时查询在此区间中，每个节点的值作为候选者，二分查找是否符合条件。



逐数字相加,采用进位标志标记是否需要进位



两个指针分别指向A,B，都从0开始，当sum<=t并且sum<count时记录结果。同时i也是从0开始的，这样就保证了两个数组的下标尽量小。