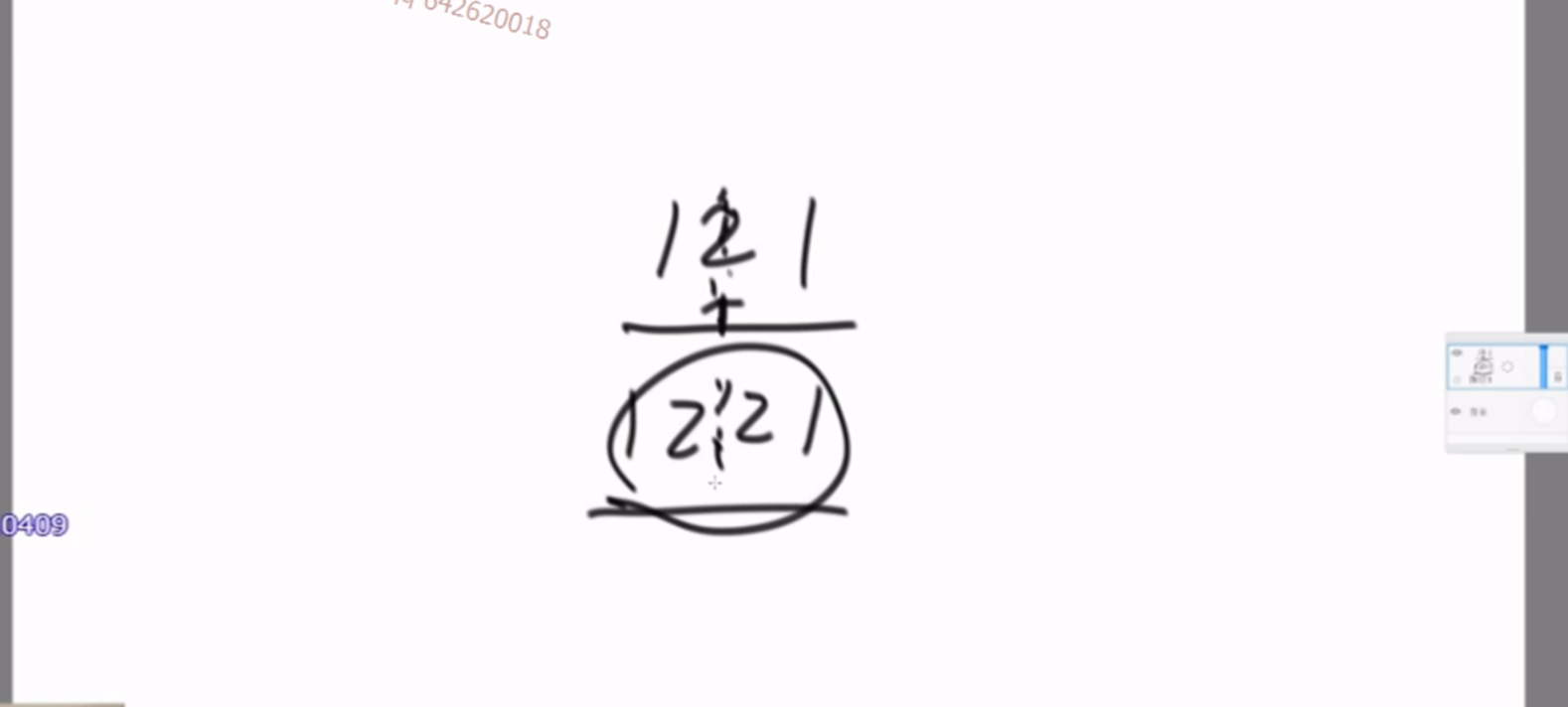
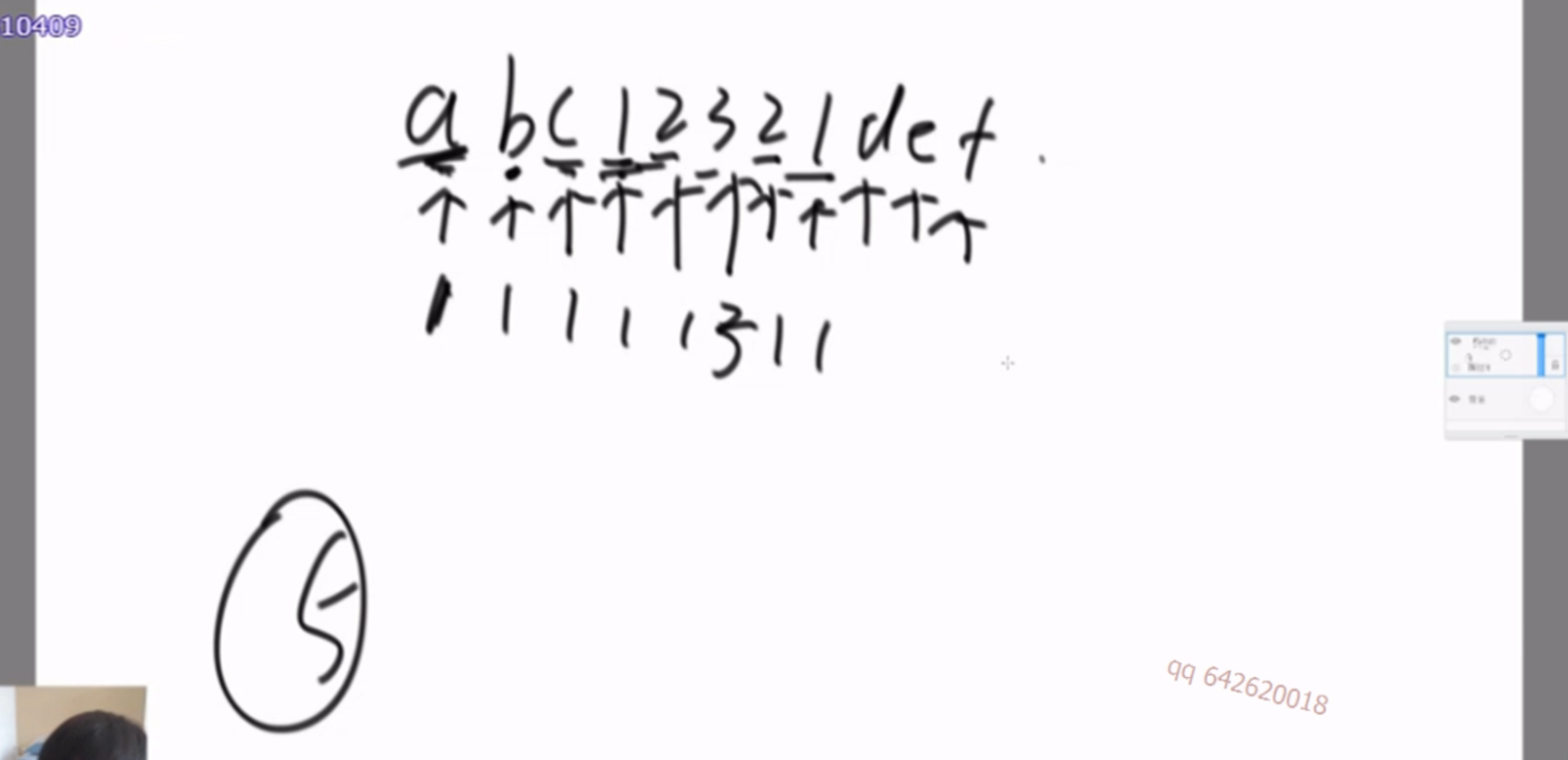
## 暴力解法

回文：奇回文和偶回文



笨办法：遍历每个字符进行暴力扩

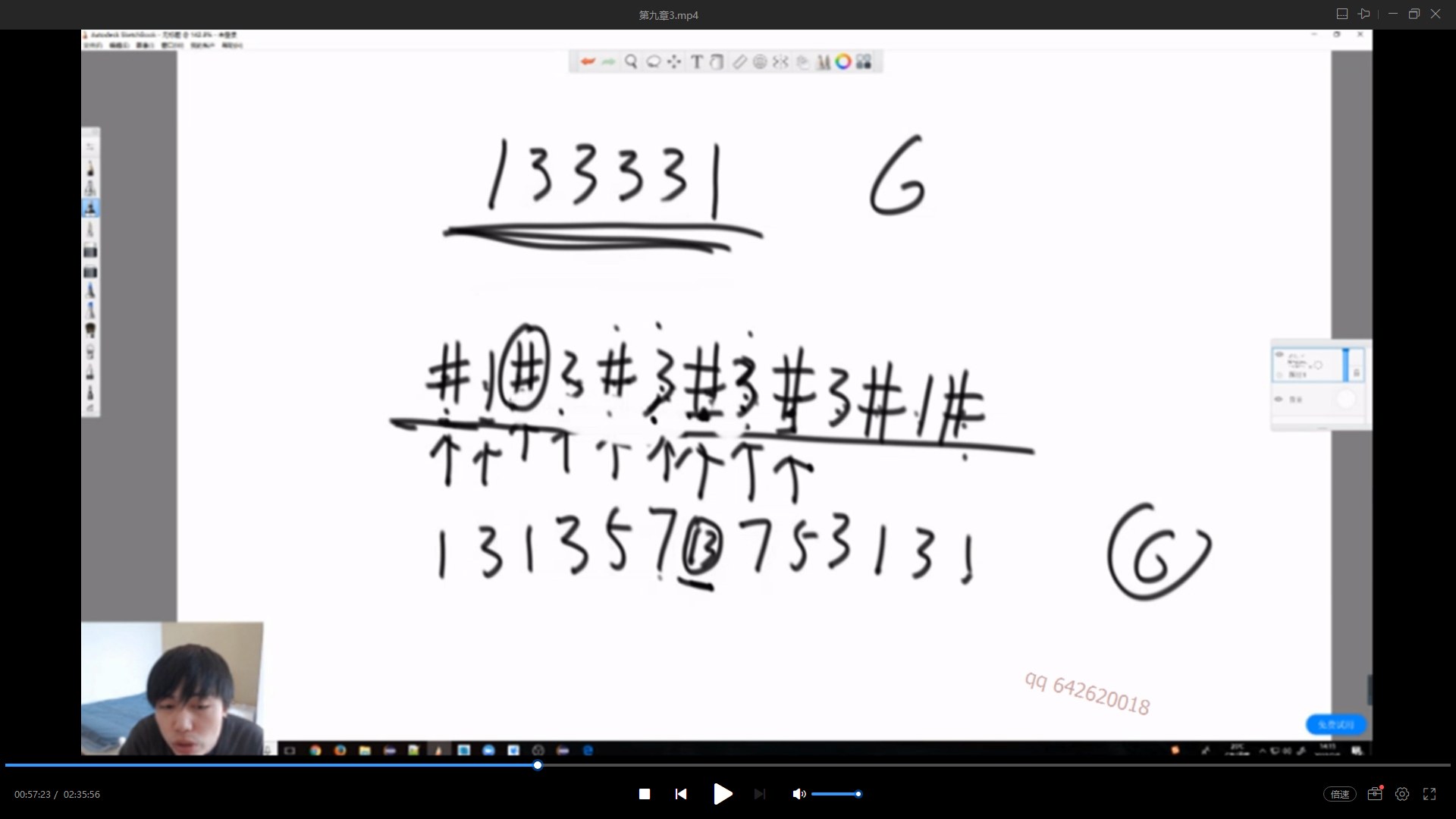
奇回文：扩的动



偶回文：扩不动



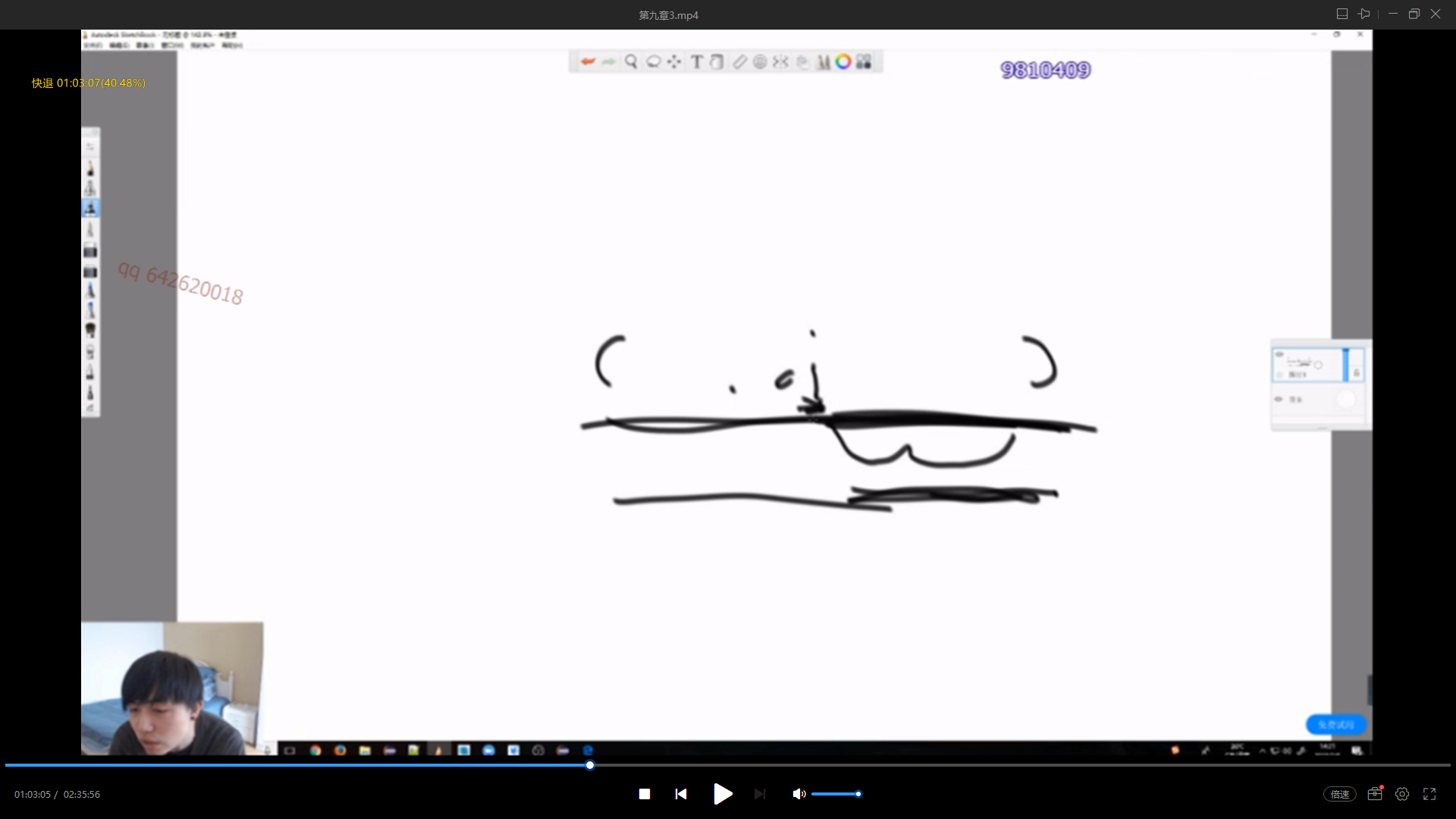
加上特殊字符来解决奇回文和偶回文的问题，实轴和虚轴互不影响，实轴只和实轴进行比对，虚轴只和虚轴进行比对，空间上是进行隔离的，所以加什么样的特殊字符是无所谓的；这个时候求出来的最大值除以2就可以得到原字符串的最大回文；时间复杂度为O(N2)



## manacher算法

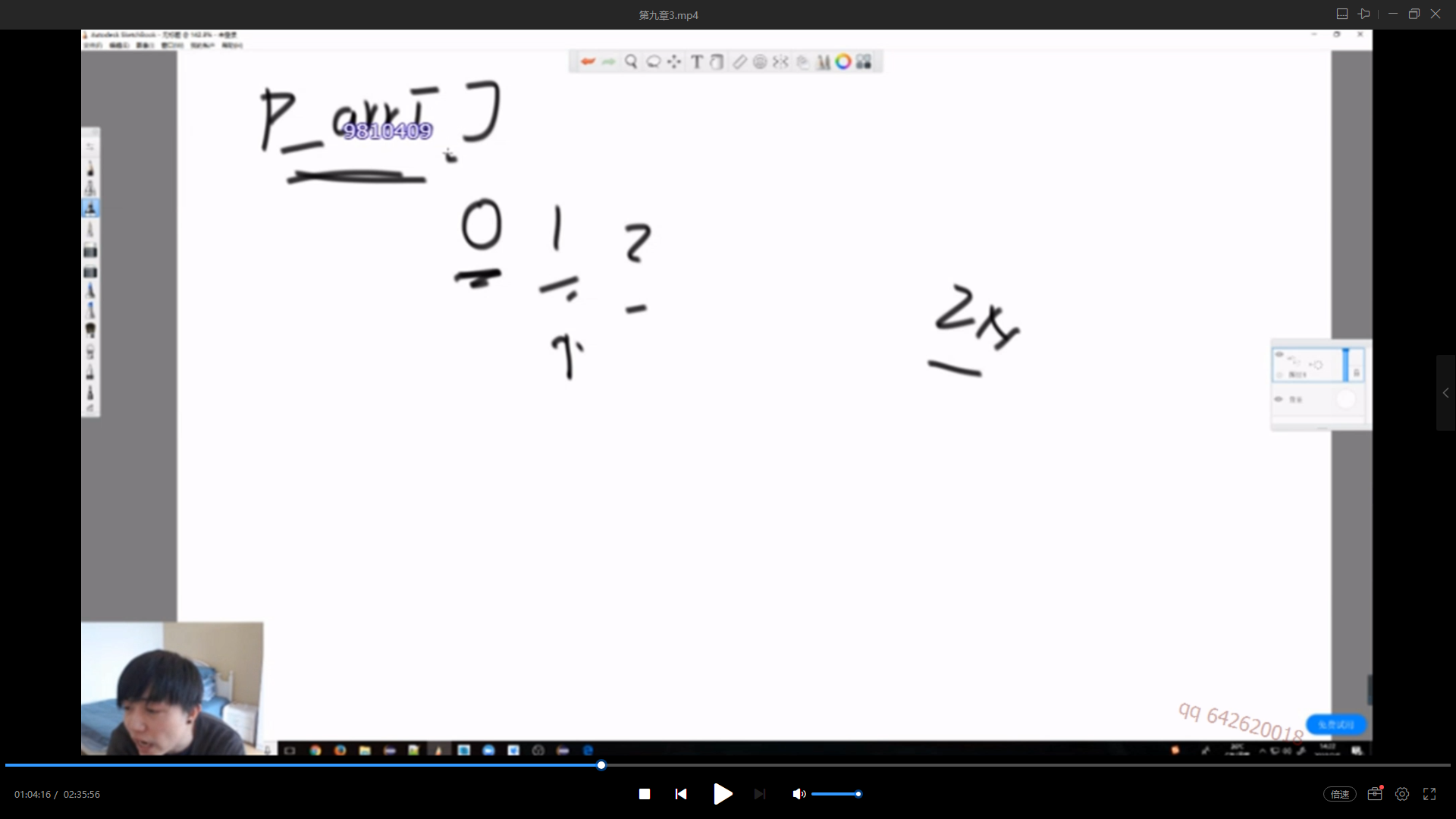
思路和暴力扩的解法是一样的，但是它有一个加速的过程；

**回文半径**

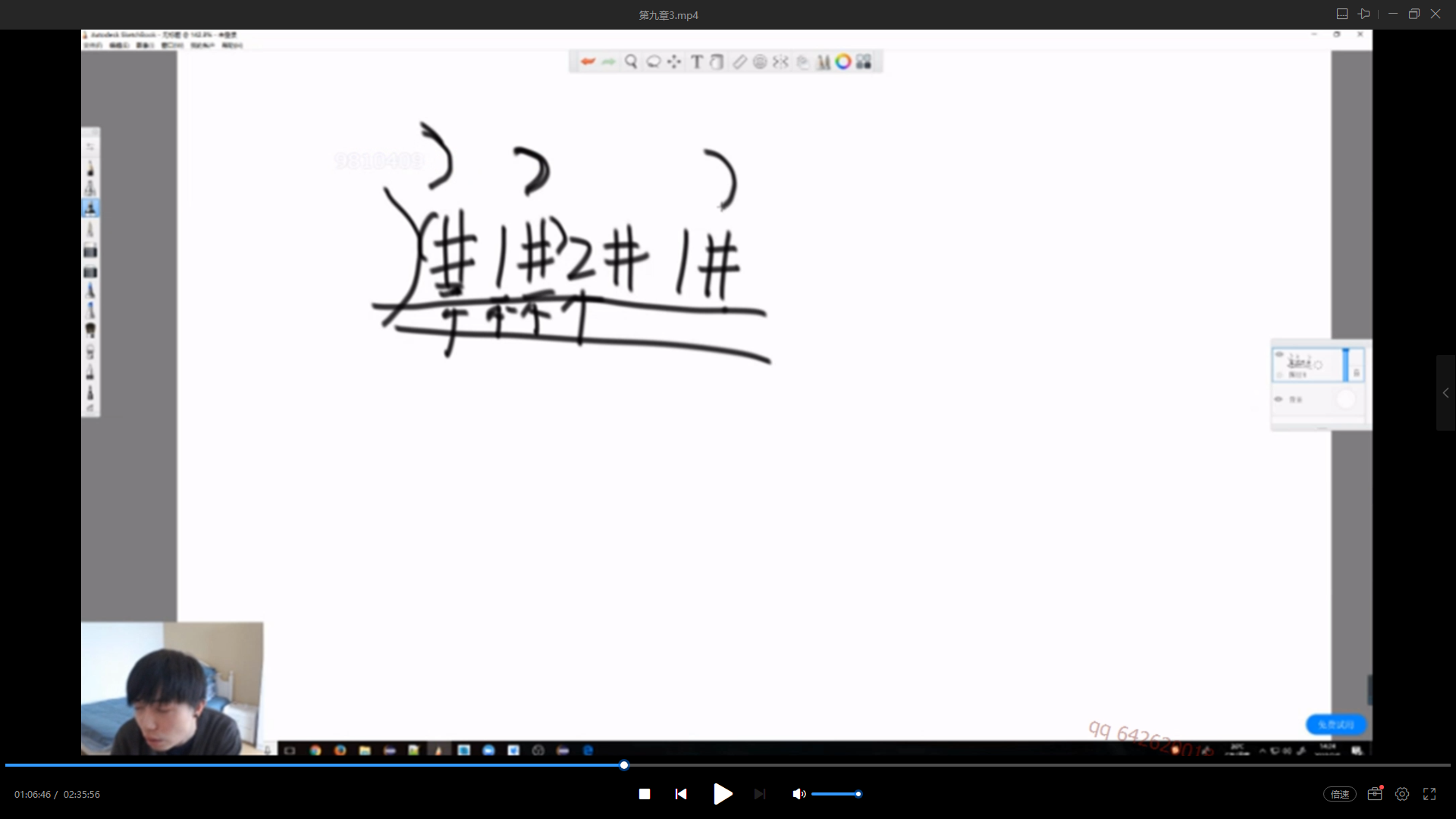


**回文半径数组**

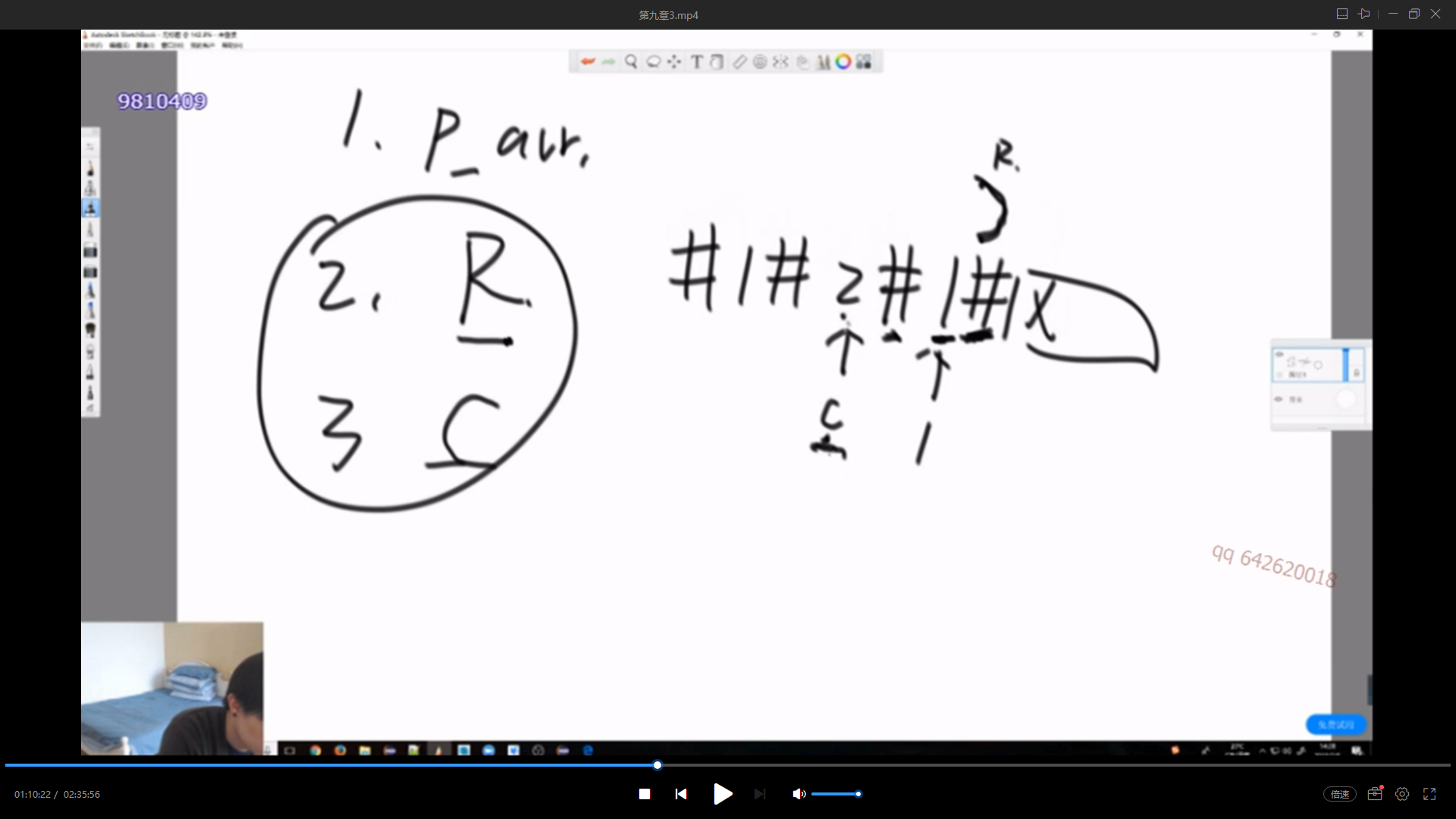
大小和处理后的字符串的大小是一样的，长度为2N



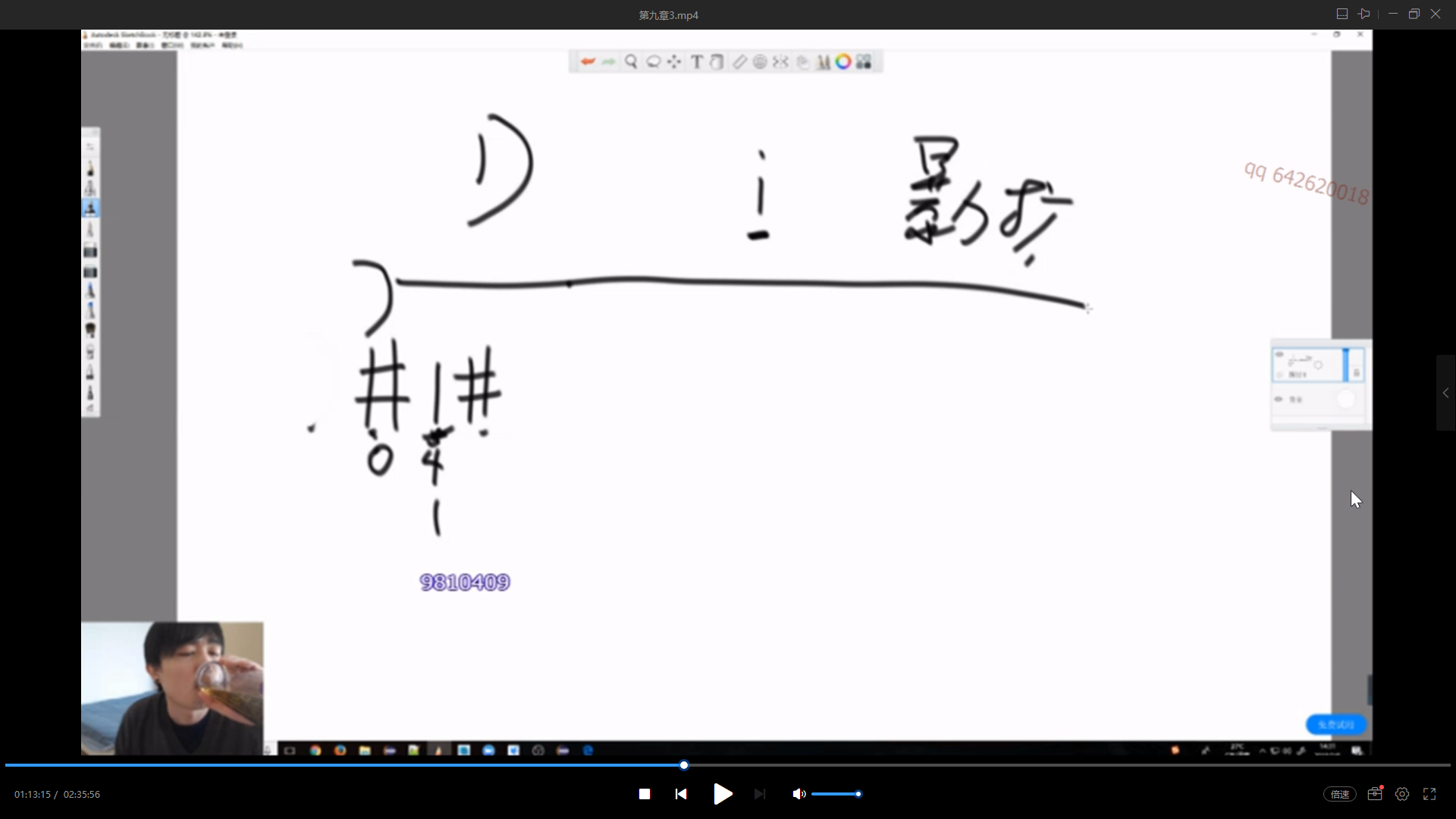
**回文半径的最右边界**:所有的位置在扩的过程中能够到达的最右的位置，即所有回文半径最右的右边界



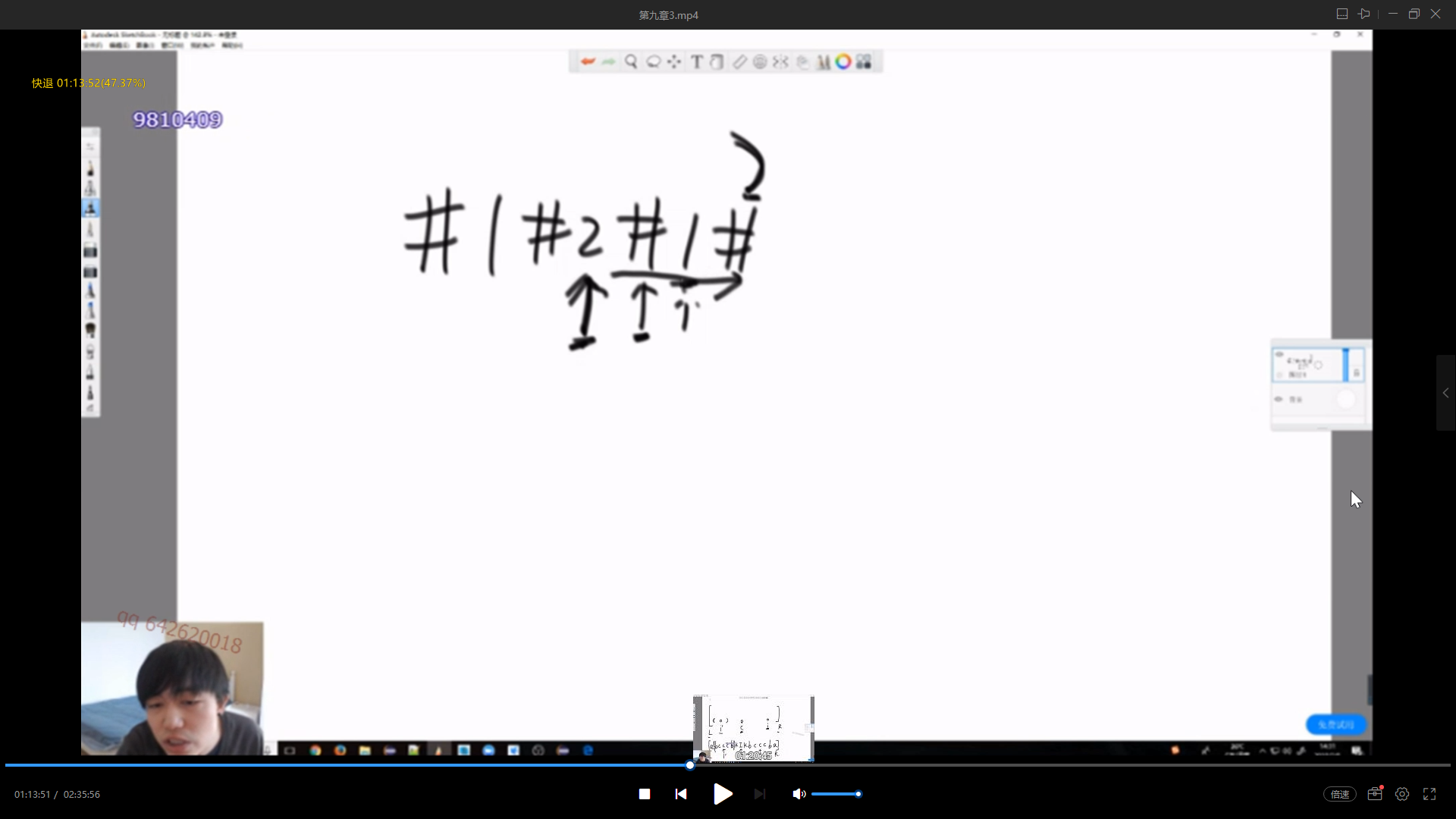
**取得最右回文半径的中心C**，只要R变得更往右时更新时就会更新中心C的值；如果多个位置它们能够到达的最右回文边界是一个的话，记录下最早的那个；如果一个位置突破了回文右边界，就记录下它突破的位置，即突破边界的中心所在；



i位置不在回文右边界里面，暴力扩



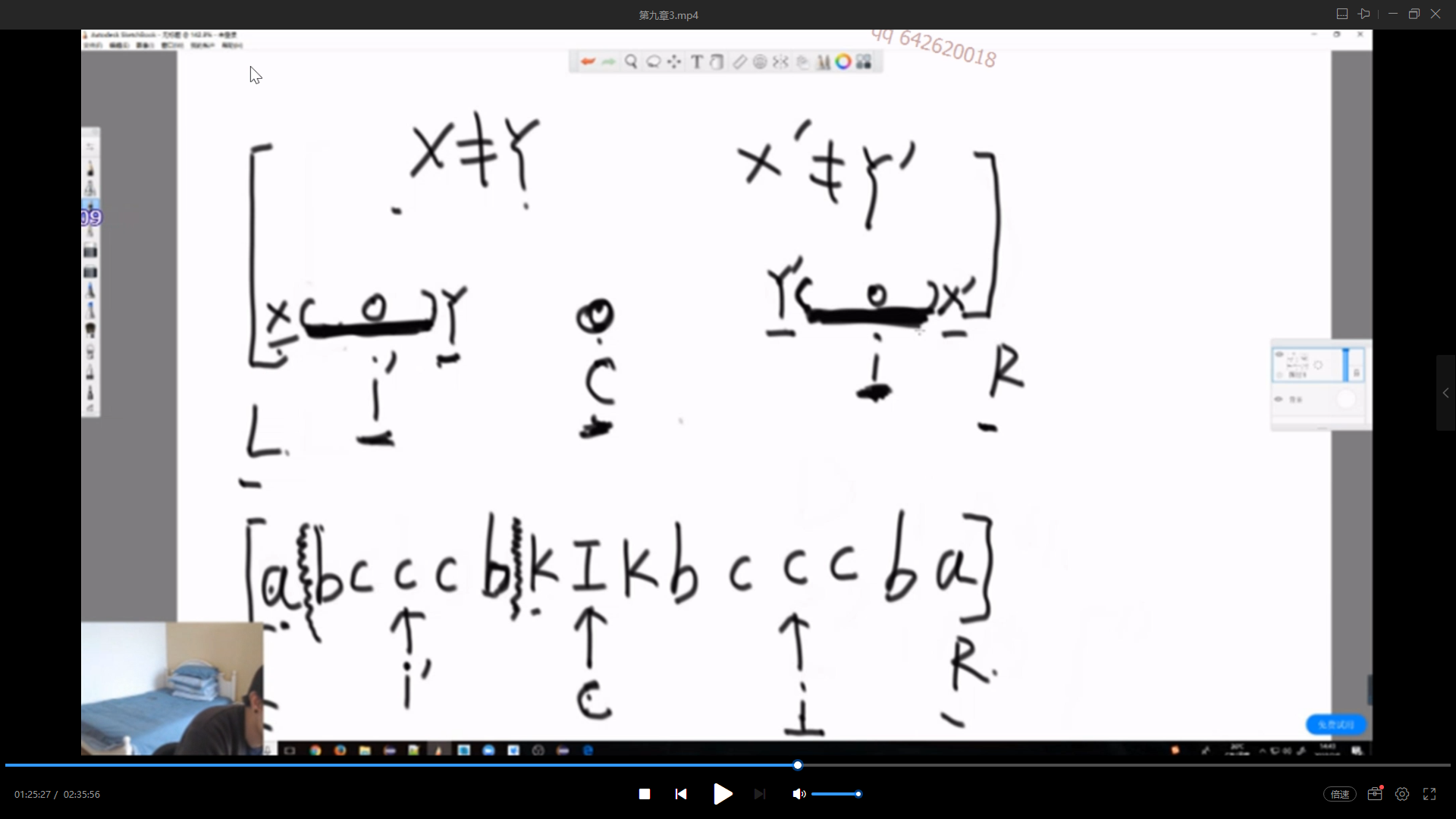
i位置在回文右边界里面，这种情况下可以进行加速



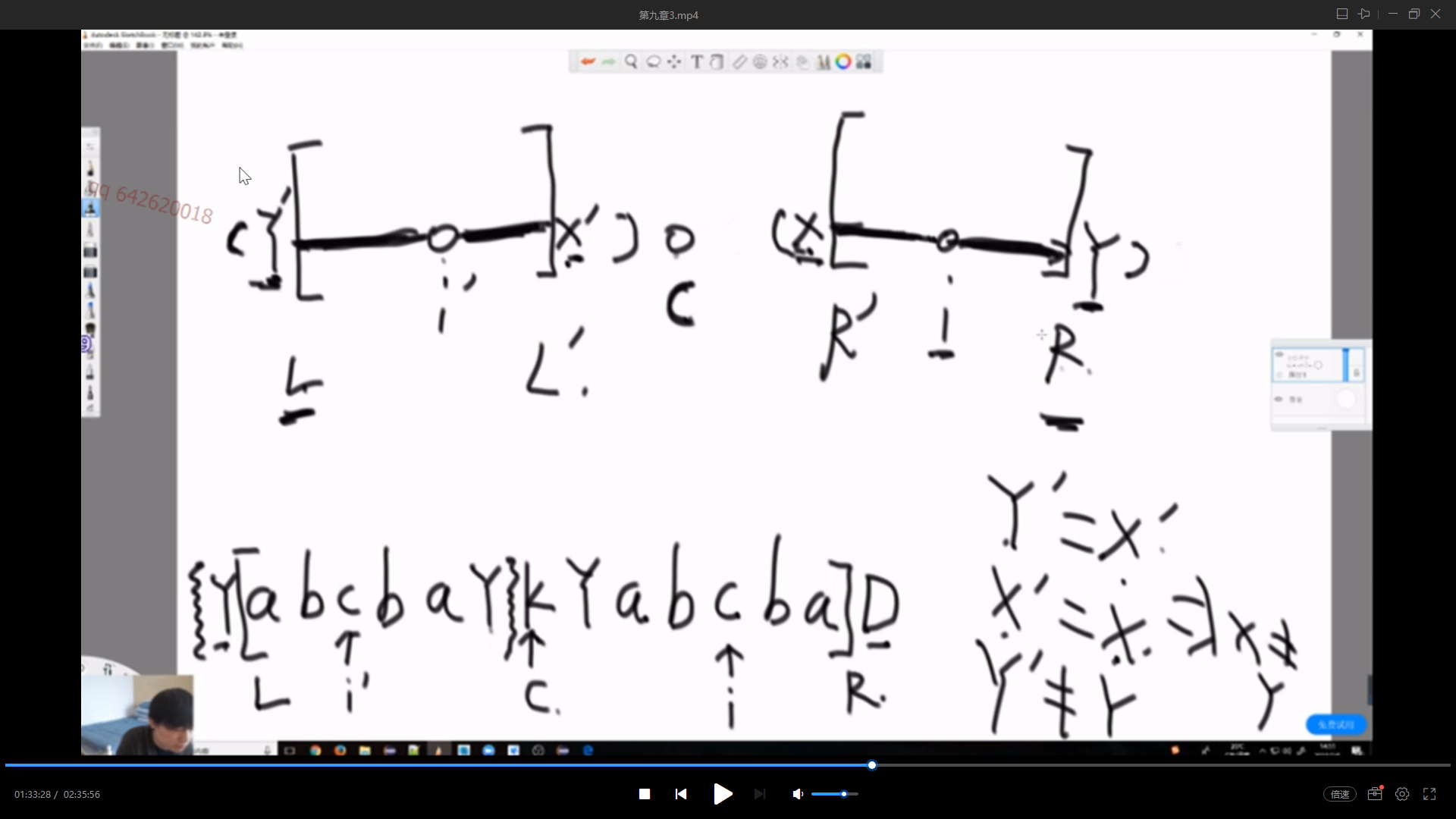
C是取得回文右边界R是的中心，i是要求的位置；回文左边界L，L和R是相对于C对称的，C到R是回文半径，L到R是回文直径，所以推断出L到R整个一定都是回文串；找到i关于C的对称点I’,

分为两大类： i在R外面，i在R里面；其中i在R里面时分类，

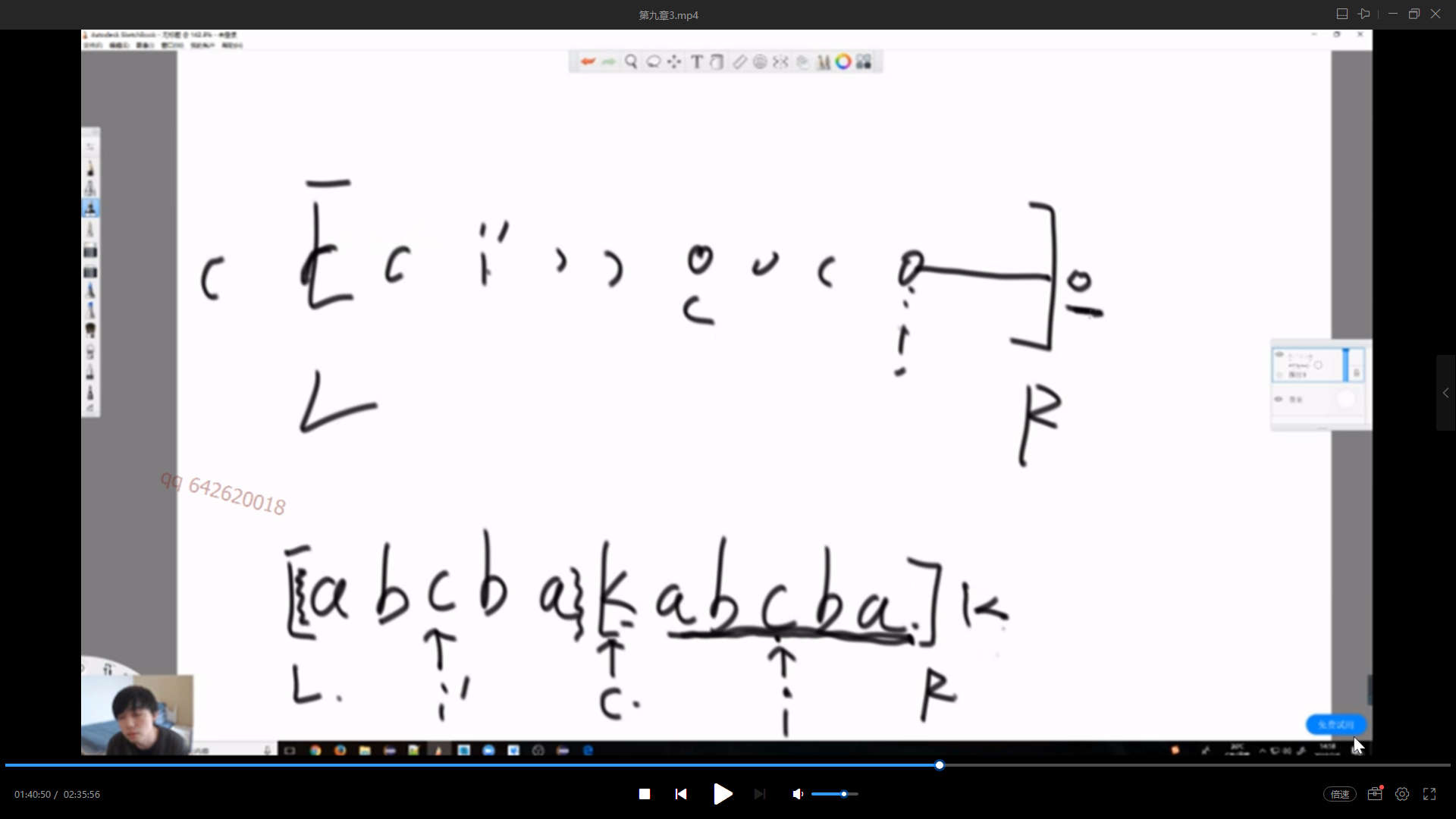
1. I’的回文范围整体在L和R的里面，这个时候直接出答案，即i扩的大小和I’是一样大；



I’的回文范围不完全在L和R里面，I’的回文最左边界在L外边；这种情况也不用求i的回文半径，直接出答案，即i的回文半径为i到R，证明是因为之前中心为C时扩到了L到R，推出Y’和Y是不相等的，X不等于Y，所以i的回文半径只能是i到R

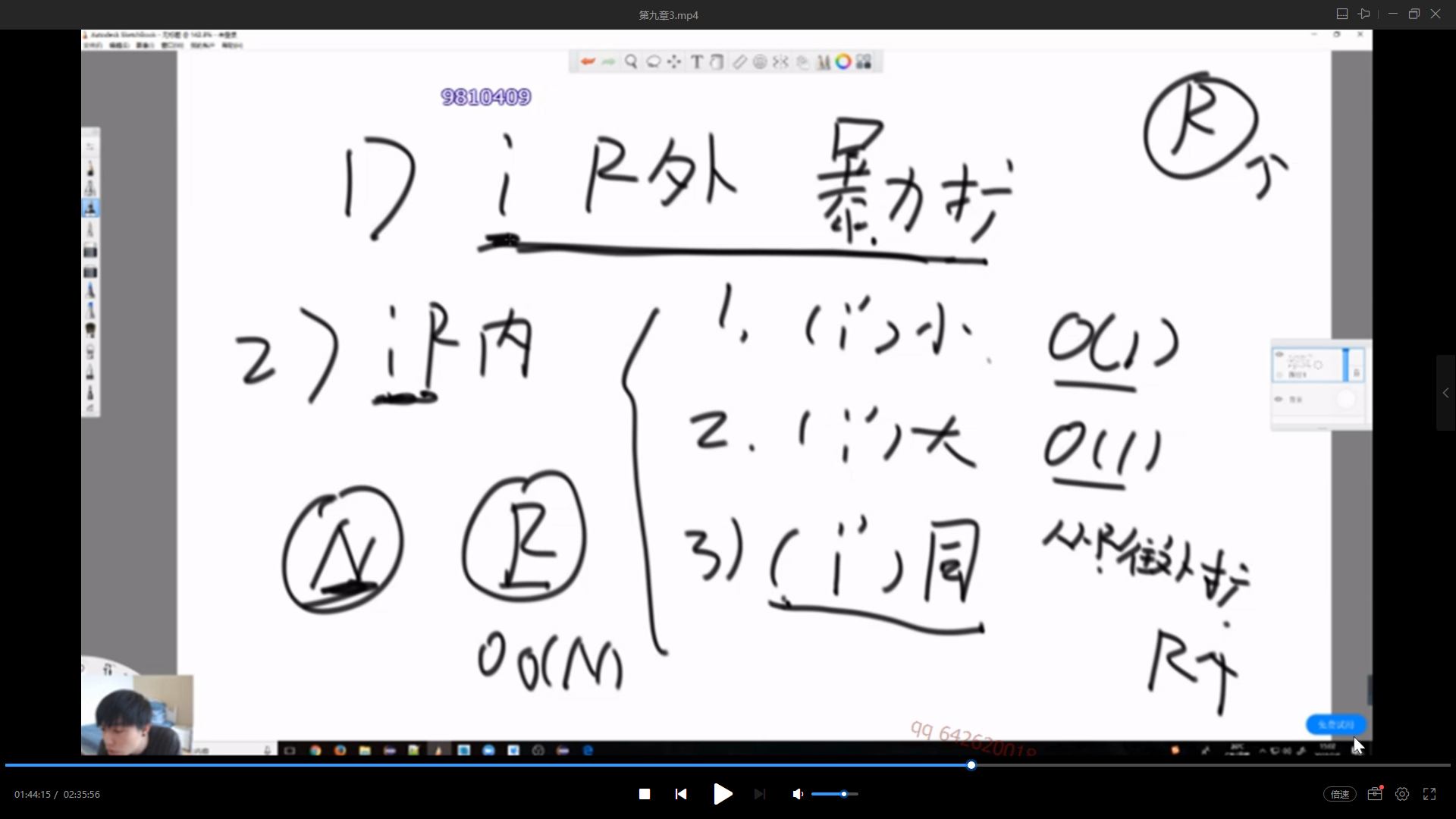


I’的回文范围和L压线，这种情况下，可以确定i到R上这段距离是不用验的，但是i是有可能会扩的更远的，如果R的下一个位置是K的话，就可以，所以这种情况需要扩

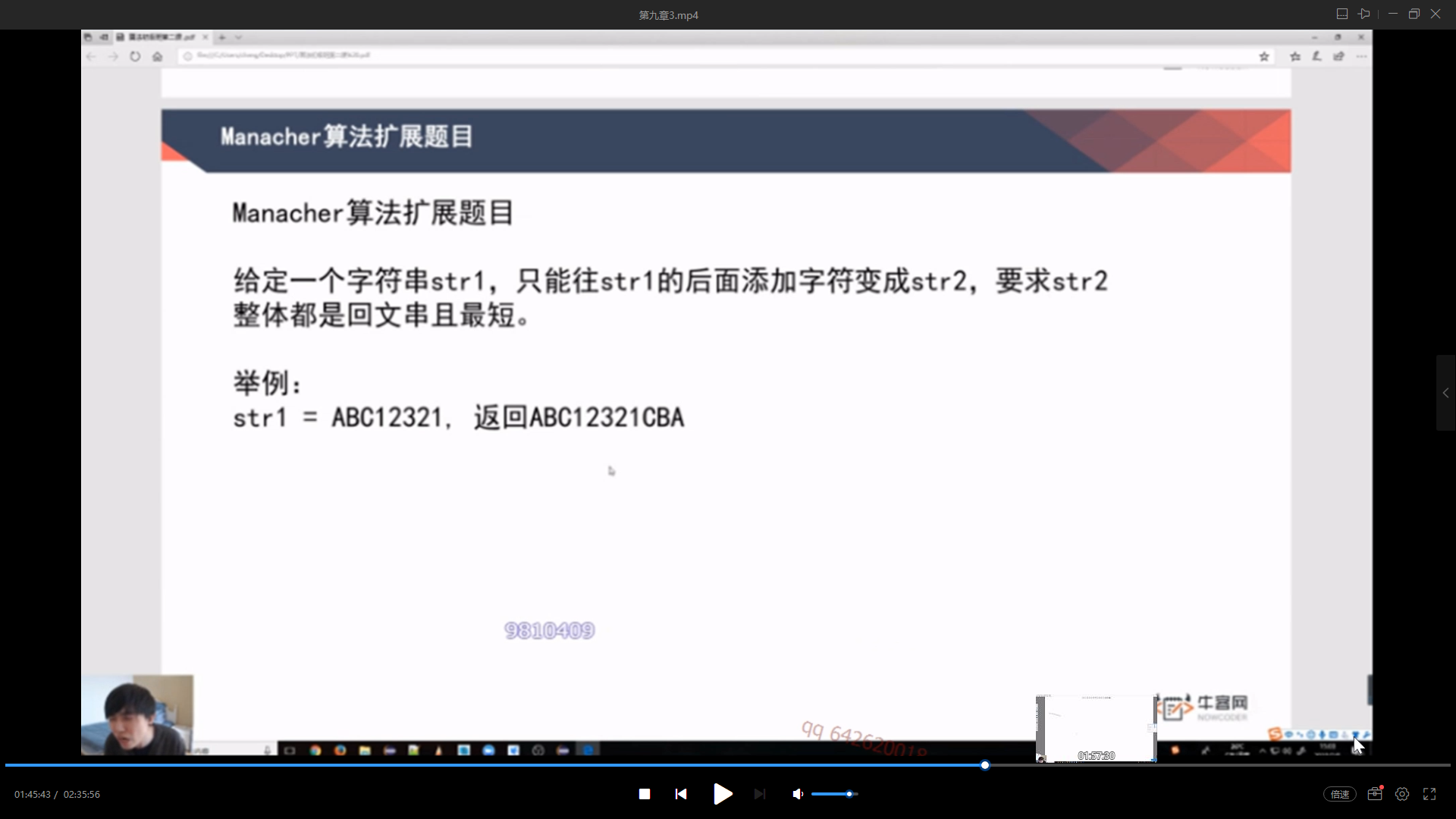


## 复杂度

通过计算R的变化可以求出整体的时间复杂度是O(N)，因为R永远不回退，R最多能够扩N次



扩展题目



这个题的解题思路就是在必须包含最后一个字符时的情况下，最长的回文串是多少，然后把前面不是回文串的整体逆序过来添加到最后即可，改写manacher算法，回文最右的边界已经达到最后一个字符了，停止即可，这个时候的C就是所要求的中心，

