1.态度：  
 不卑不亢，有自信  
  
2. 重复题目，确认要求：  
 具体步骤：  
 确认sample input/output  
 edge case:  
 输入：内容状态（无货，1个货，重复，顺序,'空1重序'）+ 取值范围   
 LinkedList：改变node数值，有序无序  
 二叉树：值是什么类型的  
 输出：'空1重序' 输入情况的结果/条件不满足，输出原输入数/结果数字重复怎么输出  
 明确注重time complexity or space complexity?  
 面试官提问环节：  
 会有基础知识的提问，先把自己会的说出来，再求教于面试官补充  
 与面试环交流有歧义，要及时纠正，不要就坡下驴  
  
3. 思考过程的交流  
  
4. 写出数据结构和算法，解释  
 整体步骤：  
 暴力+优化的层次分析  
 具体步骤：  
 写出idea及method(如有必要，请画图)  
 idea：  
 方法名 brute force / two pointer / change next pointer  
 method：  
 包含定义好的变量名的，traversal/recursion等过程名的过程表述。e.g. traversal the linklist to save node val in node\_list  
 用case1 case2 case3表示各个条件情况  
 要简单一些。eg.只要说traverse/scan index即可，不用说from i to j,  
 给出space/time complexity  
 对于dic等space comp，要具体分析。  
 tips：  
 为了保险，用for i in range(len(x)), 而不是 for elem in list  
 如果发现edge case不完全，要写到2部分，不要在本部分插入  
  
5. code之前：  
 'should I implement it in code? or you want me to continue to optimize it?'  
 'we have ... minutes. It seems that I can finish it in time.'  
  
6. right coding  
 具体步骤：  
 强调需要使用class，并对class和function命名，加功能注释  
 写出框架，再擦掉即可：  
 对变量进行及时定义，不要后补: temp/ans/res  
 书写代码，并同时书写注释：  
 对复杂操作和无法一目了然目的的代码，再行尾添加描述代码目的的注释  
 不要描述代码，比如traversal i from 1 to 19  
 针对‘method’中列出的case，针对性的注释case1/case2/case3即可  
 状态:  
 写代码时不要自言自语，想什么就跟考官交流，说我要做什么什么了  
 tips：  
 发现method不完全，需要添加内容时：同时添加method和code  
 遇到特别大的数进行运算时，可以说“当使用python时，可以避免像c++那样的数据溢出问题（overflow）”  
 求list中的长度，j-i即可  
  
7. 跑test case  
 具体步骤：  
 先跑面试官的case  
 主函数和helper function的test要清晰的隔绝开，不乱  
 再跑自己的case  
 题目类型test：  
 链表：画图指示箭头 + table  
 array: 分为两个部分，总结果表示各个关键变量的值，用table表示；小结果写在code的右边，即写即擦，展现具体的计算过程  
 横行书写 ‘100100’  
 i  
 s[i]  
8. 发音和书写  
 发音：慢而清晰  
 书写：行距适中，前后一致，缩进4个空格，行距！！！  
  
9. 考官交流  
 问了必答，比如在idea区域问space comp时，也必须要回答，虽然不清楚具体的结果，也要说我预计xxx，下面可能会有改变  
 说啥是啥，大部分是想给我提示的  
 following up没时间写时，可以只讲思路，怎么做就行。等面试结束后，可以查面试官邮箱，把具体方法发到他邮箱里。