双指针法

在字符串：这道题目，使用库函数一行代码搞定 ，我们使用双指针法实现了反转字符串的操作，**「双指针法在数组，链表和字符串中很常用。」**

接着在字符串：替换空格，同样还是使用双指针法在时间复杂度O(n)的情况下完成替换空格。

「其实很多数组填充类的问题，都可以先预先给数组扩容带填充后的大小，然后在从后向前进行操作。」

那么针对数组删除操作的问题，其实在数组：就移除个元素很难么？中就已经提到了使用双指针法进行移除操作。

同样的道理在字符串：花式反转还不够！中我们使用O(n)的时间复杂度，完成了删除冗余空格。

一些同学会使用for循环里调用库函数erase来移除元素，这其实是O(n^2)的操作，因为erase就是O(n)的操作，所以这也是典型的不知道库函数的时间复杂度，上来就用的案例了。

反转系列

在反转上还可以再加一些玩法，其实考察的是对代码的掌控能力。

字符串：简单的反转还不够！中，一些同学可能为了处理逻辑：每隔2k个字符的前k的字符，写了一堆逻辑代码或者再搞一个计数器，来统计2k，再统计前k个字符。

其实**「当需要固定规律一段一段去处理字符串的时候，要想想在在for循环的表达式上做做文章」**。

只要让 i += (2 * k)，i 每次移动 2 * k 就可以了，然后判断是否需要有反转的区间。

因为要找的也就是每2 * k 区间的起点，这样写程序会高效很多。

在字符串：花式反转还不够！中要求翻转字符串里的单词，这道题目可以说是综合考察了字符串的多种操作。是考察字符串的好题。

这道题目通过 **「先整体反转再局部反转」**，实现了反转字符串里的单词。

后来发现反转字符串还有一个牛逼的用处，就是达到左旋的效果。

在字符串：反转个字符串还有这个用处？中，我们通过**「先局部反转再整体反转」**达到了左旋的效果。

双指针法是字符串处理的常客。

KMP算法是字符串查找最重要的算法，但彻底理解KMP并不容易，我们已经写了五篇KMP的文章，不断总结和完善，最终才把KMP讲清楚。