字符串：总结篇

其实我们已经学习了十天的字符串了，从字符串的定义到库函数的使用原则，从各种反转到KMP算法，相信大家应该对字符串有比较深刻的认识了。

那么这次我们来做一个总结。

什么是字符串

字符串是若干字符组成的有限序列，也可以理解为是一个字符数组，但是很多语言对字符串做了特殊的规定，接下来我来说一说C/C++中的字符串。

在C语言中，把一个字符串存入一个数组时，也把结束符 '\0'存入数组，并以此作为该字符串是否结束的标志。

例如这段代码：

```
char a[5] = "asd";
for (int i = 0; a[i] != '\0'; i++) {
}
```

在C++中，提供一个string类，string类会提供 size接口，可以用来判断string类字符串是否结束，就不用'\0'来判断是否结束。

例如这段代码:

```
string a = "asd";
for (int i = 0; i < a.size(); i++) {
}
```

那么vector< char > 和 string 又有什么区别呢?

其实在基本操作上没有区别，但是string提供更多的字符串处理的相关接口，例如string 重载了+，而vector却没有。

所以想处理字符串，我们还是会定义一个string类型。

要不要使用库函数

在文章344.反转字符串中强调了**打基础的时候，不要太迷恋于库函数。**

甚至一些同学习惯于调用substr，split，reverse之类的库函数，却不知道其实现原理，也不知道其时间复杂度，这样实现出来的代码，如果在面试现场，面试官问：“分析其时间复杂度”的话，一定会一脸懵逼！

所以建议**如果题目关键的部分直接用库函数就可以解决，建议不要使用库函数。**

如果库函数仅仅是 解题过程中的一小部分，并且你已经很清楚这个库函数的内部实现原理的话，可以考虑使用库函数。

双指针法

在344.反转字符串 ，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27. 移除元素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151.翻转字符串里的单词中我们使用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

在反转上还可以在加一些玩法，其实考察的是对代码的掌控能力。

541. 反转字符串II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151.翻转字符串里的单词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。

KMP

KMP的主要思想是**当出现字符串不匹配时，可以知道一部分之前已经匹配的文本内容，可以利用这些信息避免从头再去做匹配了。**

KMP的精髓所在就是前缀表，在KMP精讲中提到了，什么是KMP，什么是前缀表，以及为什么要用前缀表。

前缀表：起始位置到下表i之前（包括i）的子串中，有多大长度的相同前缀后缀。

那么使用KMP可以解决两类经典问题：

1. 匹配问题：28. 实现 strStr()
2. 重复子串问题：459.重复的子字符串

再一次强调了什么是前缀，什么是后缀，什么又是最长相等前后缀。

前缀：指不包含最后一个字符的所有以第一个字符开头的连续子串。

后缀：指不包含第一个字符的所有以最后一个字符结尾的连续子串。

然后**针对前缀表到底要不要减一，这其实是不同KMP实现的方式**，我们在KMP精讲中针对之前两个问题，分别给出了两个不同版本的的KMP实现。

其中主要**理解j=next[x]这一步最为关键！**

总结

字符串类类型的题目，往往想法比较简单，但是实现起来并不容易，复杂的字符串题目非常考验对代码的掌控能力。

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。

好了字符串相关的算法知识就介绍到了这里了，明天开始新的征程，大家加油！

- 作者微信：程序员Carl
- B站视频：代码随想录
- 知识星球：代码随想录