欢迎大家参与本项目，贡献其他语言版本的代码，拥抱开源，让更多学习算法的小伙伴们收益！

> 又是一波总结

相信大家已经对双指针法很熟悉了，但是双指针法并不隶属于某一种数据结构，我们在讲解数组，链表，字符串都用到了双指针法，所有有必要针对双指针法做一个总结。

数组篇

在数组：就移除个元素很难么？中，原地移除数组上的元素，我们说到了数组上的元素，不能真正的删除，只能覆盖。

一些同学可能会写出如下代码（伪代码）：

```
for (int i = 0; i < array.size(); i++) {
    if (array[i] == target) {
        array.erase(i);
    }
}
```

这个代码看上去好像是O(n)的时间复杂度，其实是O(n^2)的时间复杂度，因为erase操作也是O(n)的操作。

所以此时使用双指针法才展现出效率的优势：**通过两个指针在一个for循环下完成两个for循环的工作。**

字符串篇

在字符串：这道题目，使用库函数一行代码搞定中讲解了反转字符串，注意这里强调要原地反转，要不然就失去了题目的意义。

使用双指针法，**定义两个指针（也可以说是索引下表），一个从字符串前面，一个从字符串后面，两个指针同时向中间移动，并交换元素。**，时间复杂度是O(n)。

在替换空格 中介绍使用双指针填充字符串的方法，如果想把这道题目做到极致，就不要只用额外的辅助空间了！

思路就是**首先扩充数组到每个空格替换成"%20"之后的大小。然后双指针从后向前替换空格。**

有同学问了，为什么要从后向前填充，从前向后填充不行么？

从前向后填充就是O(n^2)的算法了，因为每次添加元素都要将添加元素之后的所有元素向后移动。

其实很多数组（字符串）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。

那么在字符串：花式反转还不够！中，我们使用双指针法，用O(n)的时间复杂度完成字符串删除类的操作，因为题目要产出冗余空格。

在删除冗余空格的过程中，如果不注意代码效率，很容易写成了O(n^2)的时间复杂度。其实使用双指针法O(n)就可以搞定。

主要还是大家用erase用的比较随意，一定要注意for循环下用erase的情况，一般可以用双指针写效率更高！

链表篇

翻转链表是现场面试，白纸写代码的好题，考察了候选者对链表以及指针的熟悉程度，而且代码也不长，适合在白纸上写。

在链表：听说过两天反转链表又写不出来了？中，讲如何使用双指针法来翻转链表，**只需要改变链表的next指针的指向，直接将链表反转 ，而不用重新定义一个新的链表。**

思路还是很简单的，代码也不长，但是想在白纸上一次性写出bugfree的代码，并不是容易的事情。

在链表中求环，应该是双指针在链表里最经典的应用，在链表：环找到了，那入口呢？中讲解了如何通过双指针判断是否有环，而且还要找到环的入口。

使用快慢指针（双指针法），分别定义 fast 和 slow指针，从头结点出发，fast指针每次移动两个节点，slow指针每次移动一个节点，如果 fast 和 slow指针在途中相遇 ，说明这个链表有环。

那么找到环的入口，其实需要点简单的数学推理，我在文章中把找环的入口清清楚楚的推理的一遍，如果对找环入口不够清楚的同学建议自己看一看链表：环找到了，那入口呢？。

N数之和篇

在哈希表：解决了两数之和，那么能解决三数之和么？中，讲到使用哈希法可以解决1.两数之和的问题

其实使用双指针也可以解决1.两数之和的问题，只不过1.两数之和求的是两个元素的下标，没法用双指针，如果改成求具体两个元素的数值就可以了，大家可以尝试用双指针做一个leetcode上两数之和的题目，就可以体会到我说的意思了。

使用了哈希法解决了两数之和，但是哈希法并不使用于三数之和！

使用哈希法的过程中要把符合条件的三元组放进vector中，然后在去去重，这样是非常费时的，很容易超时，也是三数之和通过率如此之低的根源所在。

去重的过程不好处理，有很多小细节，如果在面试中很难想到位。

时间复杂度可以做到O(n^2)，但还是比较费时的，因为不好做剪枝操作。

所以这道题目使用双指针法才是最为合适的，用双指针做这道题目才能就能真正体会到，**通过前后两个指针不算向中间逼近，在一个for循环下完成两个for循环的工作。**

只用双指针法时间复杂度为O(n^2)，但比哈希法的O(n^2)效率高得多，哈希法在使用两层for循环的时候，能做的剪枝操作很有限。

在双指针法：一样的道理，能解决四数之和中，讲到了四数之和，其实思路是一样的，**在三数之和的基础上再套一层for循环，依然是使用双指针法。**

对于三数之和使用双指针法就是将原本暴力O(n^3)的解法，降为O(n^2)的解法，四数之和的双指针解法就是将原本暴力O(n^4)的解法，降为O(n^3)的解法。

同样的道理，五数之和，n数之和都是在这个基础上累加。

总结

本文中一共介绍了leetcode上九道使用双指针解决问题的经典题目，除了链表一些题目一定要使用双指针，其他题目都是使用双指针来提高效率，一般是将O(n^2)的时间复杂度，降为O(n)。

建议大家可以把文中涉及到的题目在好好做一做，琢磨琢磨，基本对双指针法就不在话下了。

- 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