# 概述

双指针算法是指在遍历数组时，使用两个指针来协助解决问题的一种算法。这两个指针可以是数组的两端，也可以是同一侧移动的两个指针。双指针算法通常用于寻找数组中满足特定条件的子数组、链表等问题。

常见的双指针算法包括：

1、快慢指针：一种特殊的双指针算法，通常用于链表中判断是否有环、找到环的入口等问题。

2、左右指针：在数组中，左指针一般从起始位置开始，右指针一般从末尾位置开始，根据问题的要求移动指针，解决问题。

3、滑动窗口：通过维护一个区间（窗口），根据题目要求移动窗口的左右边界，来寻找符合条件的子数组或子串。

4、对撞指针：两个指针分别从数组的两端开始向中间移动，根据题目的要求移动指针，通常用于有序数组或链表的问题。

双指针算法的优点是简单高效，时间复杂度通常为 O(n)。它们常用于数组和链表等数据结构的问题，例如求两数之和、反转数组、移除重复元素等。

# 特点

双指针算法是一种高效的解决问题的方法，但也有其优缺点和适用场景。

## 优点

优点：

1、时间复杂度低：双指针算法通常能够在 O(n) 的时间复杂度内解决问题，效率较高。

2、空间复杂度低：双指针算法通常只需要常数级别的额外空间，不需要额外的数据结构存储中间结果。

3、实现简单：双指针算法的实现比较直观，容易理解和编写。

## 缺点

缺点：

1、适用场景有限：双指针算法通常适用于一些特定类型的问题，如数组或链表相关的问题，对于其他类型的问题可能不太适用。

2、不适合所有情况：有些问题并不容易用双指针算法解决，可能需要其他更复杂的算法来解决。

# 原理

# 适用场景

适用场景：

1、有序数组或链表的问题：对撞指针常用于有序数组或链表中的查找、求和等问题。

2、滑动窗口问题：滑动窗口算法通常使用双指针来维护窗口的边界，用于求解子数组或子串的问题。

3、快慢指针问题：快慢指针算法通常用于链表中判断是否有环、找到环的入口等问题。

总的来说，双指针算法在某些特定的问题场景下具有较高的效率和简洁性，但并不适用于所有类型的问题。在解决问题时，需要根据具体情况选择合适的算法策略。

# 应用

## 查找

### 找出字符串中第一个匹配项的

## 删除

### 删除有序数组中的重复项

### 移除元素

## 数学计算

### 两数之和 II - 输入有序数组

### 三数之和

### 最接近三数之和

### 四数之和

### 平方数之和

## 接雨水