# 概述

# 特点

优点：

- 降低时间复杂度：将嵌套的循环问题转换为单循环问题，减少了不必要的重复计算；

- 提高效率：在满足条件的情况下，窗口可以快速移动，提高了算法的效率。

缺点：

- 需要额外的空间：通常需要维护一个窗口的副本，占用了额外的空间；

- 实现较复杂：对于复杂的问题，需要设计合适的窗口移动策略，实现起来相对复杂。

# 原理

滑动窗口算法是一种用于解决数组或字符串的子元素问题的有效算法。它可以将嵌套的循环问题，转换为单循环问题，降低时间复杂度。其基本思想是维护一个窗口，通过移动窗口的起始位置和结束位置来求解问题。

算法步骤如下：

1、初始化窗口的起始位置和结束位置；

2、将窗口向右滑动，扩大窗口范围，直到满足某个条件（通常是包含目标元素或满足某个条件）；

3、缩小窗口范围，直到不再满足条件，然后继续移动窗口的结束位置；

4、重复步骤2和步骤3，直到遍历完整个数组或字符串。

# 适用场景

适用场景：

- 字符串或数组中的子元素问题，如找到最长的连续子数组等；

- 需要满足某个条件的子串或子数组问题，如满足条件的最短子数组等。

# 应用

## 查找

### 学生分数的最小差值

注：LeetCode 1984

### 找到一个数字的 K 美丽值

注：LeetCode 2269

## 子串/子数组

### 子数组最大平均数 I

注：LeetCode 643

### 无重复字符的最长子串

注：LeetCode 3

### 最长和谐子序列

注：LeetCode 594

### 最长的美好子字符串

注：LeetCode 1763

### 长度为三且各字符不同的子字符串

注：LeetCode 1876

### 替换后的最长重复字符

注：LeetCode 424

### 长度最小的子数组

注：LeetCode 209

### 最长重复子数组

注：LeetCode 718

### 最长奇偶子数组

注：LeetCode 2760

## 删除

### 存在重复元素II

注：LeetCode 219

### 存在重复元素III

注：LeetCode 220