

# Грокаем алгособесы: Паттерн Two Pointers

Эффективный алгоритм с двумя указателями для массивов и строк

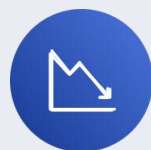


# Что такое паттерн Two Pointers?



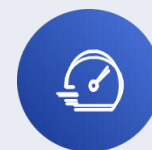
## Определение

Техника использования двух указателей для обхода структуры данных, обычно массива или строки, для решения задач за один проход



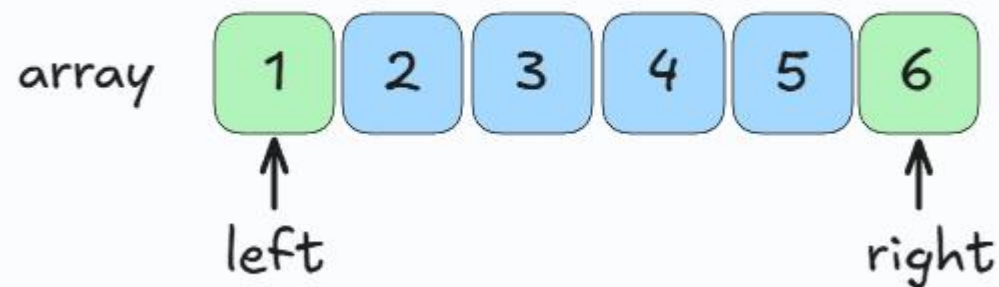
## Принцип работы

Указатели движутся навстречу друг другу, сужая область поиска



## Преимущества

Снижает временную сложность с  $O(n^2)$  до  $O(n)$ , экономит память, элегантное решение

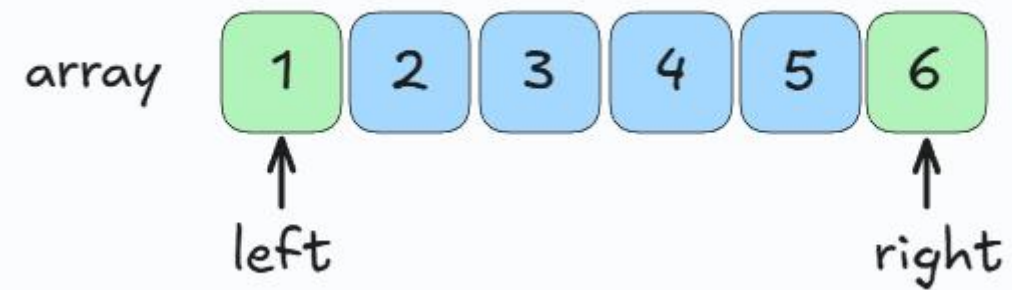


# Алгоритм встречного движения

01

## Инициализация

Устанавливаем  $left = 0$ ,  $right = n-1$  (начало и конец массива)



# Алгоритм встречного движения

01

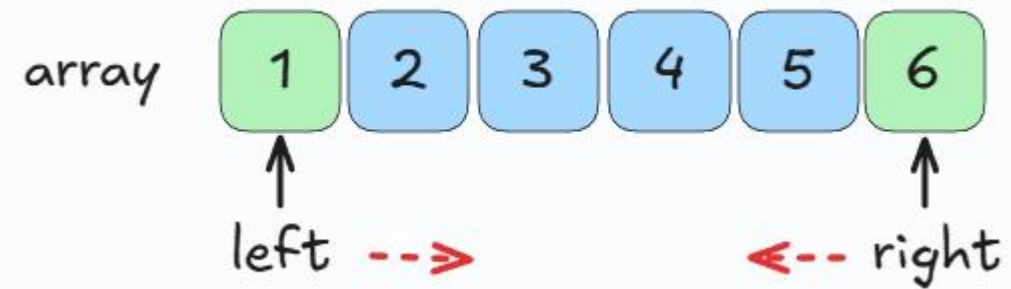
## Инициализация

Устанавливаем  $left = 0$ ,  $right = n-1$  (начало и конец массива)

02

## Условие цикла

Продолжаем пока  $left < right$  (указатели не встретились)



# Алгоритм встречного движения

01

## Инициализация

Устанавливаем  $left = 0$ ,  $right = n-1$  (начало и конец массива)

02

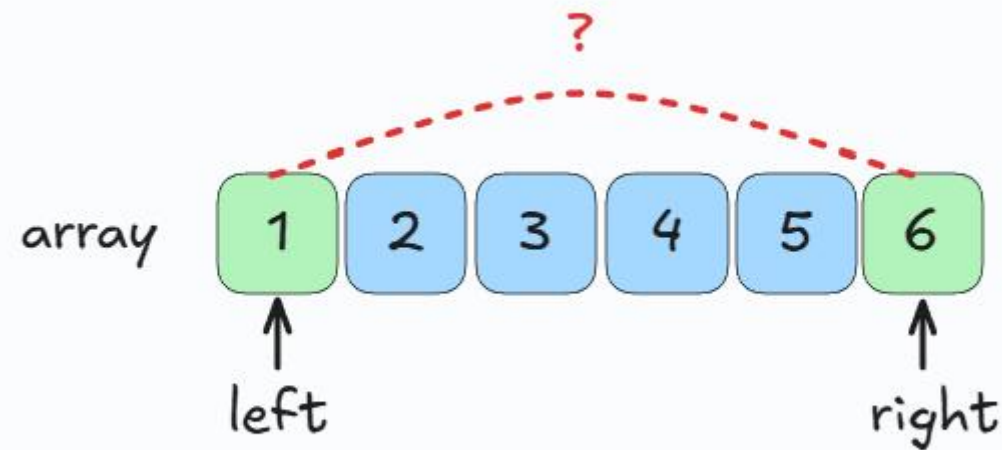
## Условие цикла

Продолжаем пока  $left < right$  (указатели не встретились)

03

## Проверка условия

Анализируем текущую пару элементов  $arr[left]$  и  $arr[right]$



# Алгоритм встречного движения

01

## Инициализация

Устанавливаем  $left = 0$ ,  $right = n-1$  (начало и конец массива)

03

## Проверка условия

Анализируем текущую пару элементов  $arr[left]$  и  $arr[right]$

02

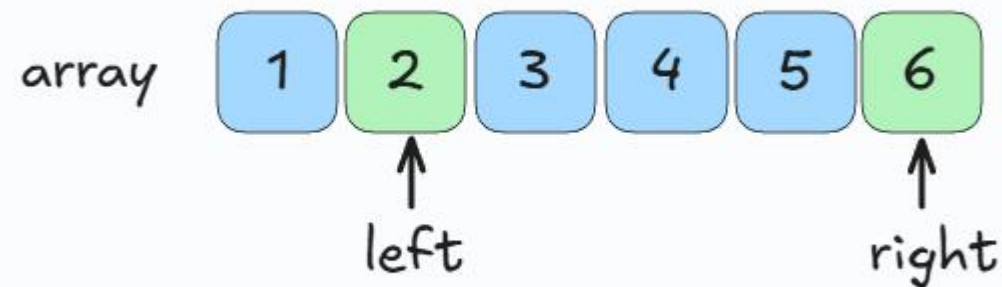
## Условие цикла

Продолжаем пока  $left < right$  (указатели не встретились)

04

## Движение указателей

Сдвигаем  $left++$  или  $right--$  в зависимости от условия задачи



# Когда применять Two Pointers?

## Когда применять?

- Ищите такие подсказки в условии:
- **Массив отсортирован:** почти всегда сигнал к использованию двух указателей.
  - **Нужно найти** пару, тройку, подмассив с определённой суммой/свойством.
  - **Требуется  $O(1)$  дополнительной памяти:** значит, нельзя использовать хеш-таблицы.

## Когда НЕ работает?

- Сортировка портит индексы (например, нужно вернуть исходные позиции).
- Если нужно найти произвольные комбинации, где порядок важен, и структура задачи не позволяет «сближать» указатели (например, массив не отсортирован).





# Пример задачи: Two Sum II



## Two Sum II - Input Array Is Sorted ([№167](#))

**Условие:** Дан отсортированный массив целых чисел. Найти два числа, сумма которых равна target. Вернуть индексы этих чисел.

**Требования:** Использовать только  $O(1)$  дополнительной памяти.

### Наивный подход

Перебор всех пар элементов

Сложность:  $O(n^2)$

### Two Pointers

Встречное движение указателей

Сложность:  $O(n)$



# Решение Two Sum II на Python

```
1 class Solution:
2     def twoSum(self, numbers: List[int], target: int) -> List[int]:
3         # Два указателя:
4         # - left_i начинается с начала массива (самый маленький элемент)
5         # - right_i начинается с конца массива (самый большой элемент)
6         # Массив numbers гарантированно отсортирован по возрастанию (условие задачи LeetCode #167).
7         left_i = 0
8         right_i = len(numbers) - 1
9
10        # Основной цикл: движение указателей навстречу друг другу
11        # Цикл завершится, как только найдём искомую пару (гарантировано по условию задачи).
12        while True:
13            current_sum = numbers[left_i] + numbers[right_i]
14
15            # Если сумма равна целевому значению – задача решена.
16            # Возвращаем индексы в индексации с единица (требование задачи).
17            if current_sum == target:
18                return [left_i + 1, right_i + 1] # Оборачиваем в список для соответствия типу возврата
19
20            # Если сумма БОЛЬШЕ target:
21            # – значит, нужно уменьшить сумму.
22            # – так как массив отсортирован, уменьшить сумму можно,
23            #   только сдвинув ПРАВЫЙ указатель влево (берём меньшее значение).
24            elif current_sum > target:
25                right_i -= 1
26
27            # Если сумма МЕНЬШЕ target:
28            # – нужно увеличить сумму.
29            # – сдвигаем ЛЕВЫЙ указатель вправо (берём большее значение).
30            else:
31                left_i += 1
```

**$O(n)$**

**Время**

Один проход массива

**$O(1)$**

**Память**

Только два указателя



**Результат**

Оптимальное решение

# Другие популярные задачи

## Valid Palindrome ([№125](#))

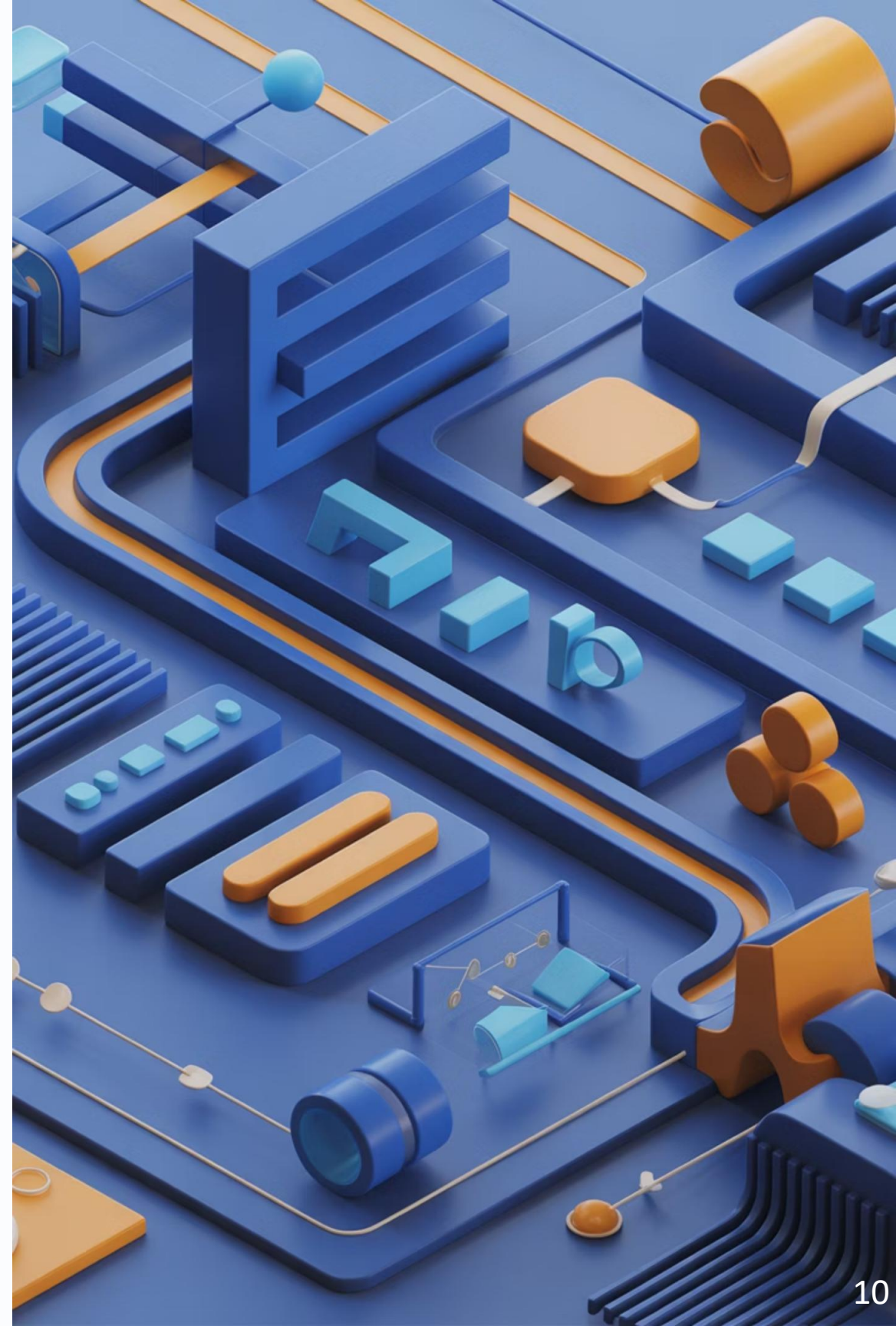
Проверить является ли строка палиндромом. Движение указателей с проверкой на равенство

## Container With Most Water ([№11](#))

Найти максимальную площадь контейнера: встречное движение с выбором оптимальной высоты

## 3Sum ([№15](#))

Найти все уникальные триплеты с суммой 0. Комбинация Two Pointers + фиксированный элемент



# Спасибо за внимание!

- Увидимся в следующих выпусках серии "Грокаем алгособесы"
- Следующая серия: Разберем технику - Быстрый и медленный указатели, и узнаем придерживаются ли указатели стратегии - *Думай медленно, решай быстро!*

## Подписывайтесь!

- Telegram-канал: [t.me/marat\\_notes](https://t.me/marat_notes)
- Обучающие видео:
  - <https://vkvideo.ru/@club231048746>
  - [https://www.youtube.com/@marat\\_notes](https://www.youtube.com/@marat_notes)
- Репозиторий:
  - [https://github.com/MaratNotes/marat\\_notes](https://github.com/MaratNotes/marat_notes)

## Призыв к действию:

- Практикуйтесь на LeetCode
- Применяйте изученные паттерны
- Готовьтесь к собеседованиям уверенно