

Монотонный стэк

Элегантный трюк для задач поиска «следующего большего элемента»



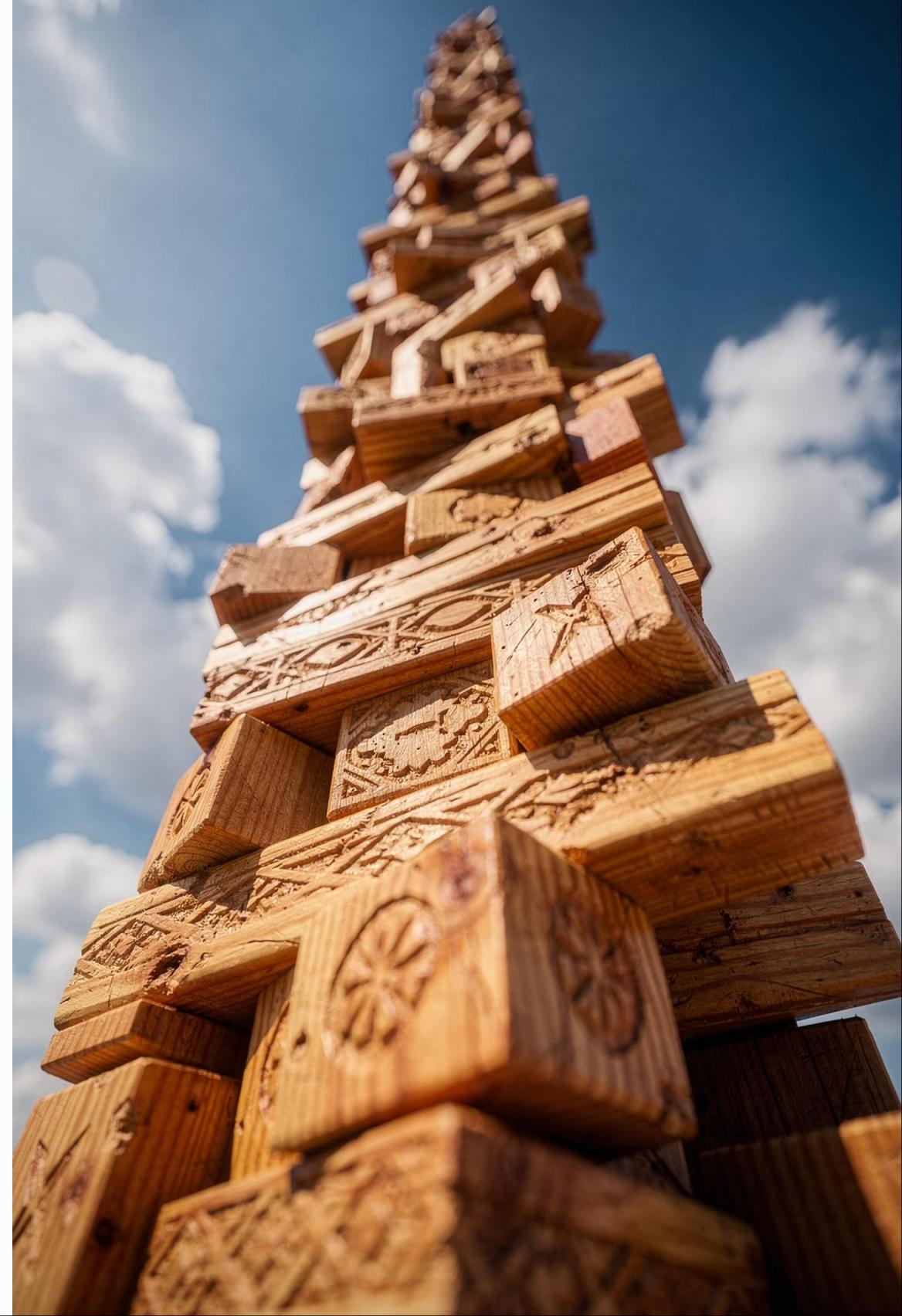
Что такое монотонный стэк?

Определение

Монотонный стэк – это специальная структура данных, представляющая собой стек с упорядоченным по возрастанию или убыванию содержимым. Элементы в нём всегда сохраняют монотонное свойство.

Основное применение

Позволяет эффективно решать задачи поиска максимумов, минимумов и других монотонных свойств на отрезках массивов, обеспечивая оптимальную производительность.





Что такое стэк?

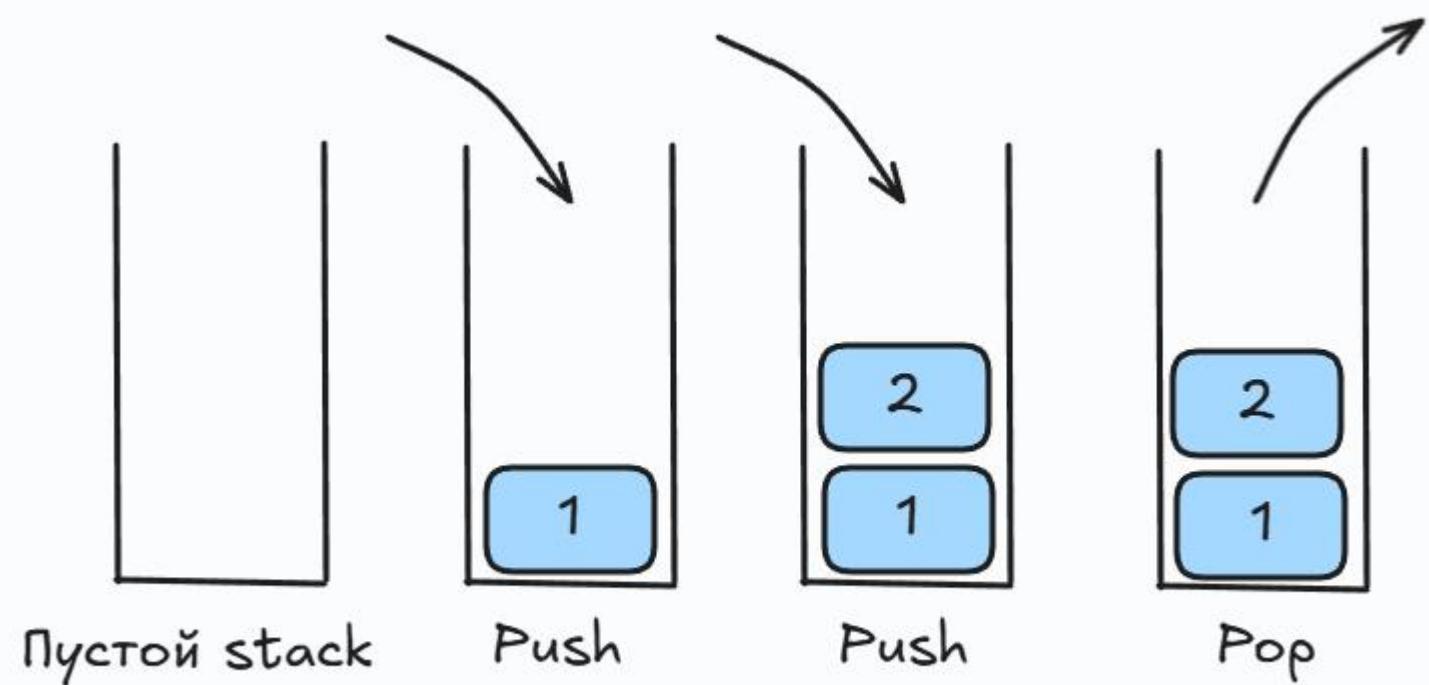
Стэк (stack) – это структура данных, которая работает по принципу LIFO (Last In, First Out), что означает «последним пришёл – первым вышел».

Основные операции:

`push(x)` - добавить элемент наверх

`pop(x)` - удалить верхний элемент

`peek()/top()` - посмотреть верхний элемент (без удаления)





Обычный стэк vs Монотонный стэк

	Правила добавления	Порядок элементов	Цель использования
Обычный стэк	Можно кладь любые элементы	Порядок элементов – произвольный	Цель: хранение + LIFO
Монотонный стэк	Перед push мы поддерживаем порядок: – убывающий: каждый новый \leq предыдущего – или возрастающий	Порядок – гарантирован (монотонность)	Цель: быстрое принятие решений на основе относительных значений

Пример задачи: поиск ближайшего большего элемента справа

Постановка задачи

1

Для каждого элемента массива найти ближайший справа больший элемент. Если такого нет – вернуть -1.

2

Наивное решение

Перебор всех элементов справа для каждой позиции. Сложность: $O(n^2)$. Неэффективно для больших массивов.

3

Оптимальное решение

Использование монотонного стэка. Сложность: $O(n)$. Каждый элемент обрабатывается максимум дважды.



Ускорение в n раз! Монотонный стэк превращает квадратичный алгоритм в линейный



Как работает алгоритм?

Шаг 1: Инициализация

Создать пустой стек `stack` (будем хранить **индексы** элементов).

Создать массив результата `res` той же длины, что и входной массив, заполненный `-1`.

```
stack = []
res = [-1] * len(arr)
```

Шаг 2: Проверка стека

Пройти по массиву слева направо

Пока стек не пуст и текущий элемент `arr[i]` больше, чем элемент по индексу на вершине стека:

Извлечь индекс `j = stack.pop()`.

Записать ответ: `res[j] = arr[i]`.

Шаг 3: Добавление в стек

После обработки всех "разрешённых" ожиданий – добавить текущий индекс `i` в стек.

Шаг 4: Завершение

Цикл завершается после прохода по всему массиву.

Все индексы, оставшиеся в стеке, уже имеют `-1` в `res` (благодаря инициализации).

Вернуть `res`.



Постановка задачи

Задача:

Для каждого элемента массива найти ближайший справа больший элемент. Если такого нет – вернуть -1.

Вход: [2, 1, 4, 3]

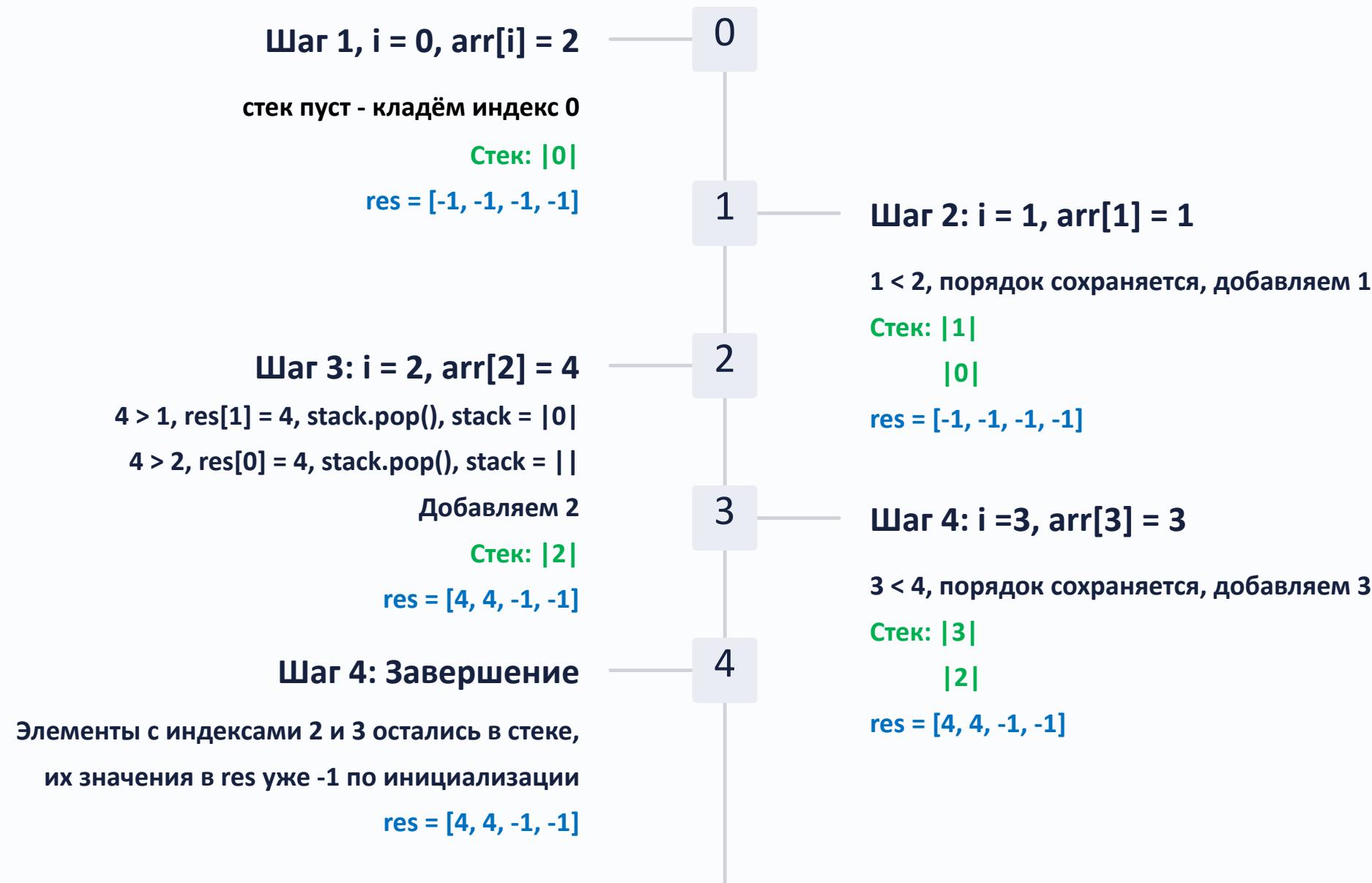
Ожидаемый выход: [4, 4, -1, -1]

Идея:

Храним в стеке индексы элементов, для которых ещё не нашли ответ. Стек – убывающий по значениям.

Пошаговая визуализация

Вход: [2, 1, 4, 3], Res: [-1, -1, -1, -1]



Пример кода на Python

```
def next_greater_element(arr):
    n = len(arr)
    res = [-1] * n          # Шаг 1: инициализация
    stack = []               # будем хранить индексы

    for i in range(n):      # Шаг 2-3: проход по массиву
        # Пока стек не пуст и текущий элемент больше вершины
        while stack and arr[i] > arr[stack[-1]]:
            j = stack.pop()      # достаём индекс
            res[j] = arr[i]       # записываем ответ
            stack.append(i)       # добавляем текущий индекс

    return res                # Шаг 4: завершение – всё уже готово
```

Функция находит для каждого элемента массива ближайший больший элемент справа, используя монотонный стэк. Сложность алгоритма – $O(n)$.

Классические задачи

1

Next Greater Element (LeetCode #496)

Требует дополнительного маппинга через словарь, так как вход состоит из двух массивов.

2

Next Greater Element II (LeetCode #503)

Необходимо пройти массив дважды.

3

Daily Temperatures (LeetCode #739)

В массив `res` необходимо записывать расстояние.

ШПАРГАЛКА

Как адаптировать под разные задачи?

Задача	Условие в while	Что записывать в res
Найти следующий больший элемент	<code>arr[i] > arr[stack[-1]]</code>	<code>res[j] = arr[i]</code>
Найти следующий меньший элемент	<code>arr[i] < arr[stack[-1]]</code>	<code>res[j] = arr[i]</code>
Найти расстояние до следующего большого	<code>arr[i] > arr[stack[-1]]</code>	<code>res[j] = i - j</code>

Типичные ошибки и полезные советы

✗ Частые ошибки

- Хранить значения вместо индексов: нельзя обновить `res` по позиции
- Использовать `if` вместо `while`: пропустите несколько ответов за один шаг
- Забыть, что стек может быть не пуст в конце: но это нормально! (`res` уже -1)
- Путать направление: искать «больший» при возрастающем стеке: не работает

✓ Полезные советы

- Всегда инициализируйте `res = [-1] * n`: тогда финальная обработка не нужна
- Рисуйте стек на бумаге для `[2, 1, 4, 3]`: 90% понимания приходит от этого
- Начинайте с задачи «следующий больший»: остальные – её вариации
- Проверяйте крайние случаи: пустой массив, убывающая/возрастающая последовательность

Как распознать задачу для монотонного стека?

🔍 Когда думать о монотонном стеке?

Задача, скорее всего, подходит, если:

- Нужно найти ближайший элемент слева/справа, удовлетворяющий условию (больше / меньше)
- Говорится о максимуме/минимуме на подотрезке, особенно если отрезки «скользят» или зависят от позиции
- Есть фразы вроде: «для каждого элемента...», «ближайший справа...», «сколько дней до более тёплого...», «максимальная площадь под гистограммой»

❑ 🔍 Ключевой сигнал: решение в лоб – $O(n^2)$, но ответ для i зависит только от ближайших "конкурентов".

Подписывайся и оставайся на связи!

Сегодня ты узнал:

Монотонный стек – это не новая структура, а умное правило использования обычного стека, чтобы решать задачи за $O(n)$ вместо $O(n^2)$.

Не откладывай практику:

- Реши LeetCode #739 – Daily Temperatures
- Модифицируй код для поиска следующего меньшего элемента

В следующем видео:

Топ K элементов: как найти самые частые/крупные/важные за $O(n \log k)$?

 Все примеры и код – в описании.

Контакты и ресурсы:

Телеграм: t.me/marat_notes

Репозиторий с кодом: github.com/MaratNotes/marat_notes

YouTube: youtube.com/@marat_notes

VK Видео: vkvideo.ru/@club231048746

