

Билет 14. Merge Sort

Билет 14. Сортировка слиянием (Merge Sort)

Основная идея: "Разделяй и властвуй" + слияние отсортированных массивов

Ключевая операция: Слияние двух отсортированных массивов

Если у нас есть два отсортированных массива – можно их слить в третий, который будет тоже отсортирован за $O(n + m)$, где n, m – размеры массивов.

Метод двух указателей:

- Ставим указатели в начало обоих массивов
- На каждом шаге выбираем меньший элемент и двигаем соответствующий указатель
- Повторяем, пока не пройдем оба массива

Алгоритм сортировки слиянием

Рекурсивный подход (сверху-вниз)

Шаг 1: Разделить массив на две примерно равные половины

Шаг 2: Рекурсивно решить подзадачи, применив шаги 1 и 3, пока не дойдем до базы (массив из одного элемента)

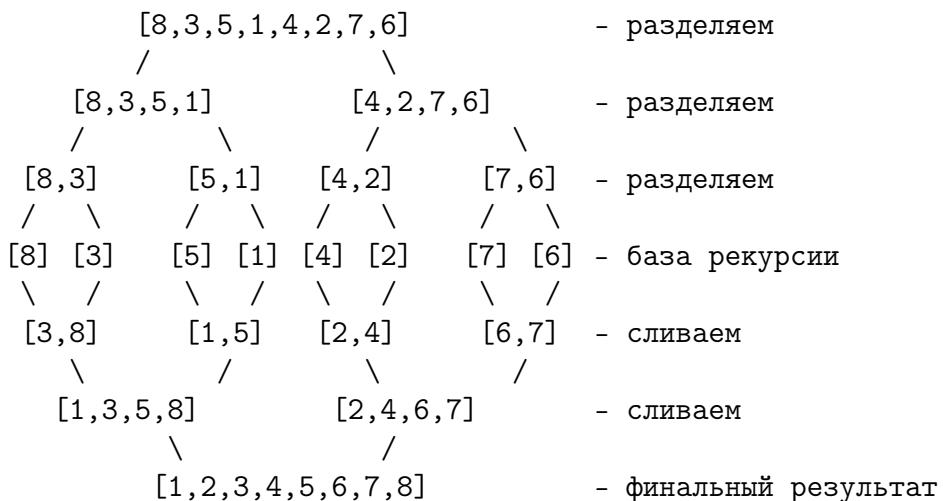
Шаг 3: Объединить решения подзадач (слить два отсортированных массива в один большой)

База рекурсии

Массив из 0 или 1 элемента уже отсортирован!

Визуализация процесса

Исходный массив: [8, 3, 5, 1, 4, 2, 7, 6]



Анализ времени работы

Рекуррентное соотношение:

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

По мастер-теореме:

- $a = 2$ (количество подзадач)
- $b = 2$ (во сколько раз уменьшается задача)
- $f(n) = O(n)$ (время на разделение и слияние)

- $n^{\log_b a} = n^{\log_2 2} = n^1 = n$
- Случай 2: $f(n) = \Theta(n^{\log_b a}) \Rightarrow T(n) = \Theta(n^{\log_b a} \log n)$

Итог: $T(n) = O(n \log n)$

Потребление памяти

- **Дополнительная память:** $O(n)$ – необходим временный массив для слияний
- **Стек вызовов:** $O(\log n)$ – глубина рекурсии
- **Общая память:** $O(n)$

Итеративная реализация

Алгоритм можно реализовать **без рекурсии**, начав решать задачу снизу-вверх:

1. Сливаем соседние массивы размеров 1
2. Затем сливаем соседние массивы размеров 2
3. Затем сливаем соседние массивы размеров 4
4. И так далее до полного массива