Лекция 6. Алгоритм сортировки слиянием (Merge Sort)

Казахский национальный университет имени аль-Фараби

Кафедра компьютерных наук

Алгоритмы и структуры данных

Лектор: старший преподаватель, PhD, Кенжебек Ержан Ғалымжанұлы

Цель и план лекции

Цель лекции

Изучить алгоритм сортировки слиянием (Merge Sort), этапы разделения и слияния, а также его временную сложность, достоинства и недостатки.

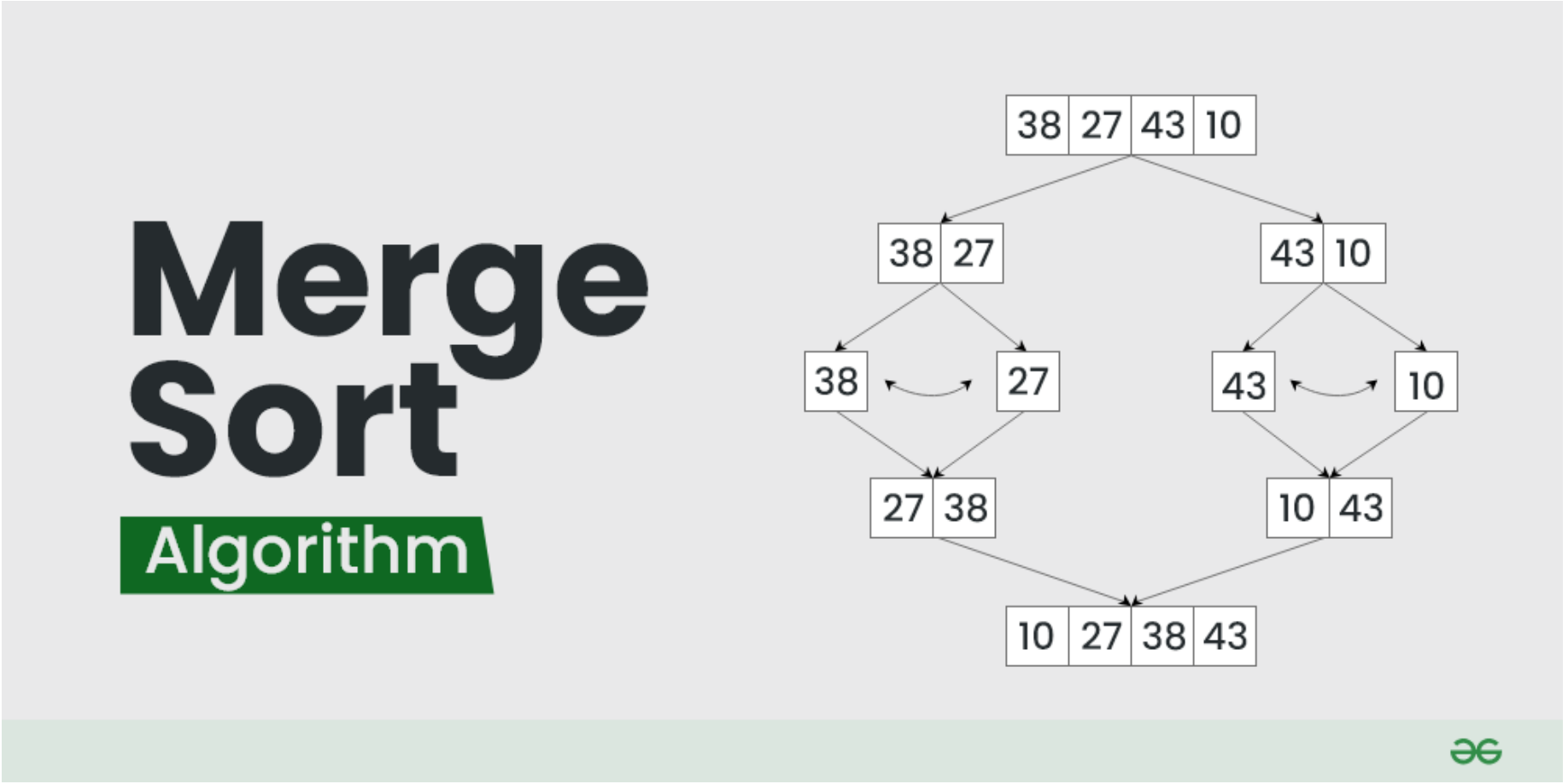
План лекции

1. Определение и идея сортировки слиянием
2. Этап разделения массива
3. Этап слияния отсортированных подмассивов
4. Пошаговые примеры работы
5. Временная и пространственная сложность
6. Преимущества и недостатки

Определение

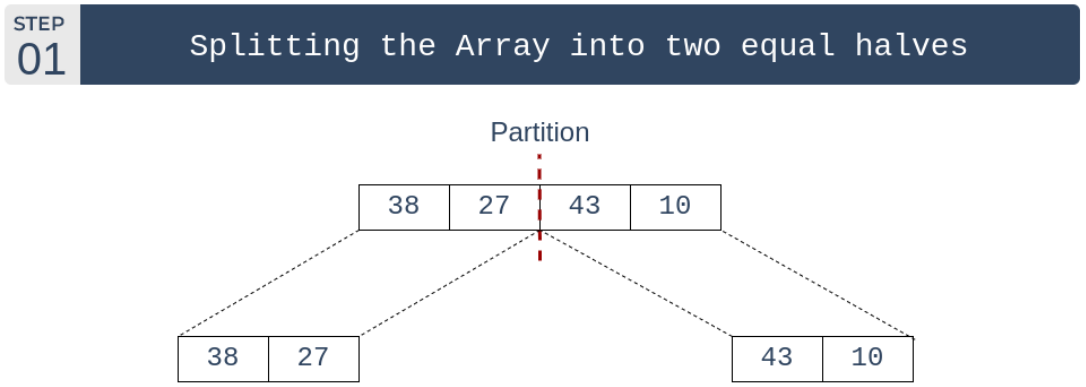
- Сортировка слиянием определяется как алгоритм сортировки , который работает путем разделения массива на более мелкие подмассивы, сортировки каждого подмассива, а затем обратного **слияния отсортированных подмассивов** для формирования окончательного отсортированного массива.
- Проще говоря, мы можем сказать, что процесс сортировки слиянием состоит в том, чтобы разделить массив на две половины, отсортировать каждую половину, а затем снова объединить отсортированные половины. Этот процесс повторяется до тех пор, пока весь массив не будет отсортирован.

Пример


Merge
Sort
Algorithm
38 27 43 10
38 27
43 10
38
27
43
10
27 38
10 43
10 27 38 43

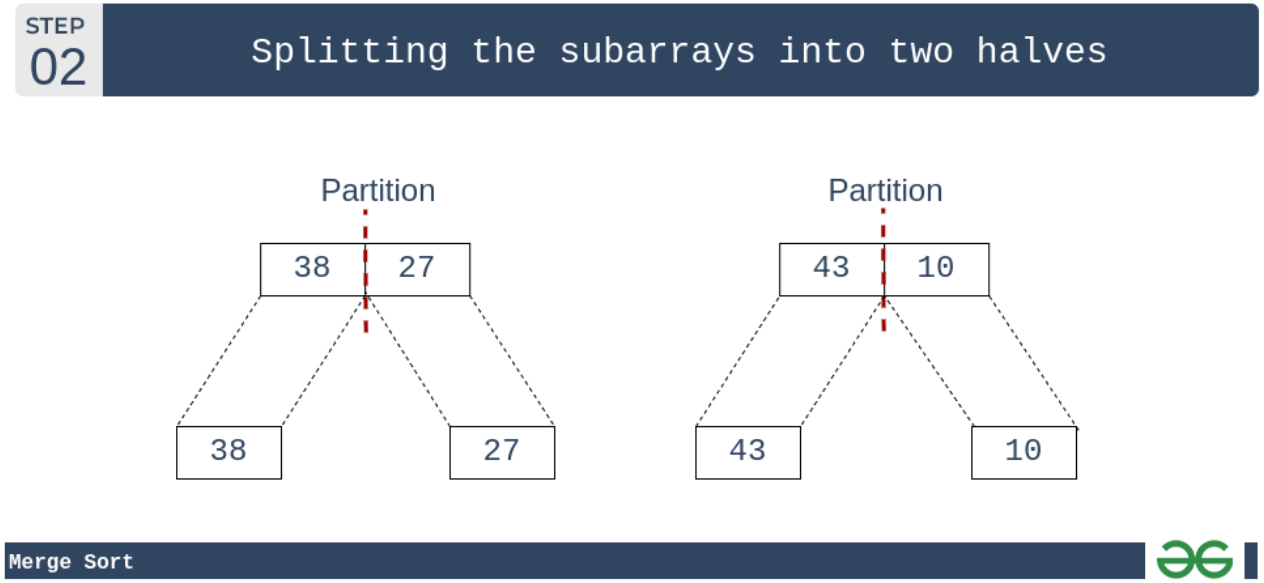
Как работает сортировка слиянием?

- Сортировка слиянием — это рекурсивный алгоритм, который непрерывно разбивает массив пополам до тех пор, пока его невозможно будет разделить, т. е. в массиве останется только один элемент (массив с одним элементом всегда сортируется). Затем отсортированные подмассивы объединяются в один отсортированный массив.
- Рассмотрим пример:
  - Давайте рассмотрим массив arr[] = {38, 27, 43, 10}
  - Первоначально разделите массив на две равные половины:



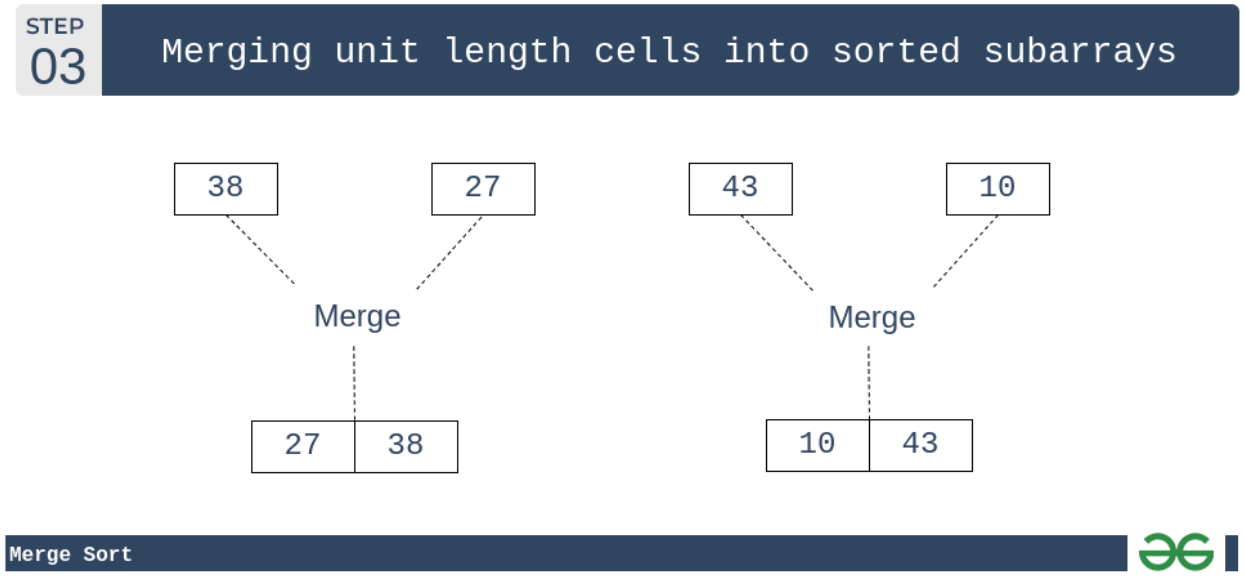
Разделение

- Эти подмассивы далее делятся на две половины. Теперь они становятся массивом единичной длины, который больше нельзя разделить, а массив единичной длины всегда сортируется.

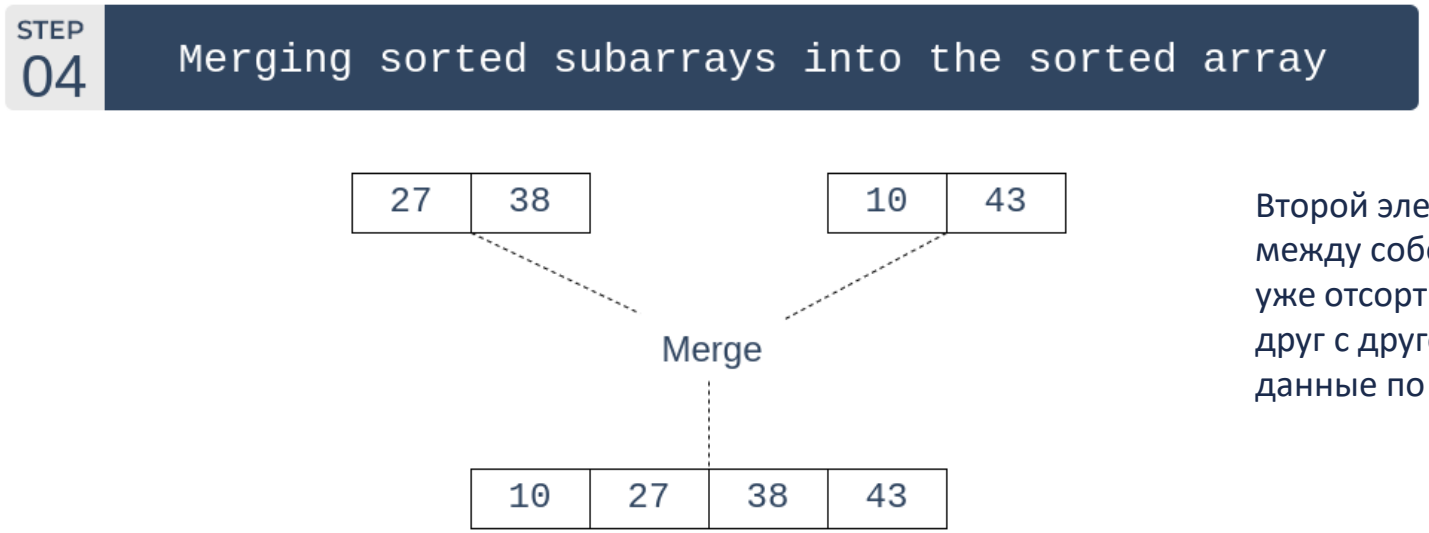


Объединение

- *Эти отсортированные подмассивы объединяются, и мы получаем отсортированные подмассивы большего размера.*



- Этот процесс слияния продолжается до тех пор, пока отсортированный массив не будет построен из меньших подмассивов.

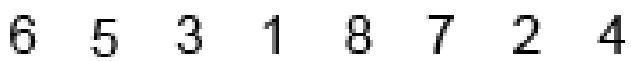


Второй элемент — соединение отсортированных элементов между собой, причём тоже хитрым способом: раз оба массива уже отсортированы, то нам достаточно сравнивать элементы друг с другом по очереди и заносить в итоговый массив данные по порядку.

Пример с объяснением

- Если представить алгоритм по шагам, он будет выглядеть так:
  1. Исходный массив: [4 2 5 1].
  2. Делим пополам: [4 2] ←→ [5 1] (у нас появилось два новых массива, значит, к ним применяем тоже сортировку слиянием).
  3. Делим пополам **первый массив**: [4] ←→ [2].
  4. Раз в каждом по одному элементу, то сортируем и склеиваем в один: [2 4].
  5. Делим пополам **второй массив**: [5] ←→ [1].
  6. Раз в каждом по одному элементу — сортируем и склеиваем в один: [1 5].
  7. К нам вернулись два отсортированных подмассива, а значит, нам нужно их отсортировать и склеить в один.
  8. Сравниваем первые элементы: 1 и 2. Единица меньше двух, значит, в итоговый массив записываем 1, и у нас остаются два массива: [2 4] и [5].
  9. Сравниваем первые элементы: 2 и 5. Двойка меньше пяти, значит, в итоговый массив записываем 2, и у нас остаются два массива: [4] и [5].
  10. Точно так же сравниваем первые элементы до тех пор, пока в обоих массивах не исчезнут все элементы. Как только это произойдёт — массив отсортирован.
-

Пример с анимацией



Особенность алгоритма Merge Sort

- Главное преимущество сортировки слиянием — она работает всегда с одной и той же скоростью на любых массивах. Допустим, у нас есть два массива:
    1. В одном будут числа в случайном порядке.
    2. В другом — уже отсортированные числа, но последний и предпоследний элементы (разные) просто поменяны местами.

- В обоих случаях сортировка слиянием покажет одно и то же время: ей всё равно, какие данные обрабатывать, алгоритм всё равно сделает это за один проход и одинаковое время. Благодаря этому свойству алгоритм используют там, где нужно обработать данные за заранее известное время. Например, если для сортировки 1000 элементов серверу требуется 50 миллисекунд, то можно настроить ему отправку новой партии данных каждую 51 миллисекунду — так мы будем уверены, что к моменту отправки новых данных старые уже будут обработаны.

Временная сложность алгоритма

- Временная сложность: O(N log(N)), сортировка слиянием — это рекурсивный алгоритм, и временная сложность может быть выражена следующим рекуррентным соотношением.

$$T(n) = 2T(n/2) + \theta(n)$$

- Вышеупомянутое повторение можно решить либо с помощью Recurrence Tree, либо с помощью Master method. Он попадает в случай II Мастер-метода, и решением повторения является θ(Nlog(N)). Временная сложность сортировки слиянием равна θ(Nlog(N)) во всех трех случаях (худшем, среднем и лучшем), поскольку сортировка слиянием всегда делит массив на две половины, и для объединения двух половин требуется линейное время.
- **Вспомогательное пространство**: O(N). При сортировке слиянием все элементы копируются во вспомогательный массив. Таким образом, для сортировки слиянием требуется N вспомогательного пространства.

Преимущества и недостатки

- **Преимущества сортировки слиянием:**
  - **Стабильность** : сортировка слиянием — это стабильный алгоритм сортировки, что означает, что он поддерживает относительный порядок равных элементов во входном массиве.
  - **Гарантированная производительность в наихудшем случае:** временная сложность сортировки слиянием в наихудшем случае равна O(N logN), что означает, что она хорошо работает даже на больших наборах данных.
  - **Параллелизуемый** : сортировка слиянием — это естественный параллелизуемый алгоритм, что означает, что его можно легко распараллелить, чтобы использовать преимущества нескольких процессоров или потоков.
- **Недостатки сортировки слиянием:**
  - **Пространственная сложность:** сортировка слиянием требует дополнительной памяти для хранения объединенных подмассивов во время процесса сортировки.
  - **Не на месте:** сортировка слиянием не является алгоритмом сортировки на месте, что означает, что для хранения отсортированных данных требуется дополнительная память. Это может быть недостатком в приложениях, где использование памяти является проблемой.
  - **Не всегда оптимально для небольших наборов данных:** для небольших наборов данных сортировка слиянием имеет более высокую временную сложность, чем некоторые другие алгоритмы сортировки, такие как сортировка вставками. Это может привести к снижению производительности для очень маленьких наборов данных.

Выводы

1. Сортировка слиянием рекурсивно делит массив пополам до одного элемента, а затем сливает отсортированные части.
2. Время работы одинаково для любых входных данных: O(n log n) в лучшем, среднем и худшем случае.
3. Для слияния требуется дополнительная память O(n); алгоритм не является сортировкой «на месте».
4. Алгоритм устойчив и хорошо поддаётся распараллеливанию; на малых наборах данных бывает менее эффективным.

Контрольные вопросы

1. Опишите идею сортировки слиянием.
2. Как выполняется слияние двух отсортированных массивов и какова его сложность?
3. Запишите рекуррентное соотношение для времени работы Merge Sort и укажите его решение.
4. Почему время работы не зависит от исходной упорядоченности массива?
5. Назовите достоинства и недостатки сортировки слиянием.
6. Выполните сортировку слиянием массива {38, 27, 43, 10}.

Список литературы

Основная литература

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. 3-е изд. — М.: Вильямс, 2013.
2. Седжвик Р., Уэйн К. Алгоритмы на Java. 4-е изд. — М.: Вильямс, 2013.
3. Кнут Д. Искусство программирования. Т. 3. Сортировка и поиск. 2-е изд. — М.: Вильямс, 2007.

Дополнительная литература

4. Скиена С. Алгоритмы. Руководство по разработке. 2-е изд. — СПб.: БХВ-Петербург, 2011.
5. Левитин А. Алгоритмы: введение в разработку и анализ. — М.: Вильямс, 2006.
6. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms. 4th ed. — MIT Press, 2022.

Электронные ресурсы

7. VisuAlgo — визуализация алгоритмов и структур данных: visualgo.net