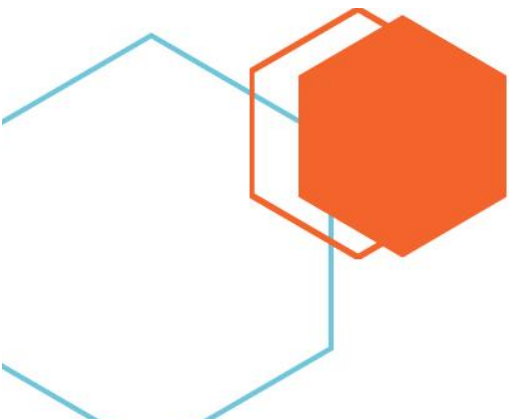




Algorithms With Java

Week 0 Day 2: Merge Sort

Академия Ковалевского





Содержание

1. Теория	3
Сортировка слиянием — Merge Sort	3
2. Практическая работа	4
Задачи	5





1. Теория

Сортировка слиянием — Merge Sort

Сегодня мы реализуем новый тип сортировки - сортировка слиянием. Список хороших материалов:

- Общее [видео](#) по теме
- Пример [реализации](#) в Java

Вопрос, на который нужно быть готовым ответить: какова сложность алгоритма с точки зрения памяти?



2. Практическая работа

В данном задании нужно переопределить метод `createSortedArray` интерфейса `Sort`:

```
package academy.kovalevskiy.algorithms.week0.day2;

import academy.kovalevskiy.algorithms.week0.day0.Sort;
import java.util.Arrays;
import java.util.Comparator;

public class MergeSort implements Sort {

    @Override
    public <T> void sort(T[] target, Comparator<T> comparator) {
        // TODO
    }

    @Override
    public <T> T[] createSortedArray(T[] target, Comparator<T> comparator) {
        // TODO
    }
}
```

Так как этот метод сортировки создает новый массив, то нам еще нужно реализовать метод сортировки “на месте”. Избежать выделения памяти не сможем, подумайте как можно в несколько строк написать метод (вызывая метод `createSortedArray`).

Не забывайте про переопределение методов сложности.



Задачи

```
package academy.kovalevskiy.algorithms.week0.day2;

public class Tasks {

    public static OptionalInt findIndex(int[] sortedArray, int target) {
        // TODO
    }
}
```

- **findIndex** — метод, который на вход принимает **уже отсортированный** массив и число, а возвращает индекс, по которому это число находится в массиве путем выполнения бинарного поиска (не применяйте тупой перебор всех элементов массива!).

Сегодня можно использовать **только компаратор** и **собственный код** из предыдущего дня!