

# Algorithms and data structures

lecture #4. Merge sort

Mentor: <....>

## lecture #4. Merge sort

- Merge sort
  - Общая информация
  - Алгоритм разделения
  - Псевдокод
  - Детальный разбор на картинках
  - Реализация Java
  - Merge sort VS Quick sort

## Сортировка слиянием (Merge sort)

Подобно быстрой сортировке (Quick sort), сортировка слиянием представляет собой алгоритм «разделяй и властвуй».

В парадигме «разделяй и властвуй» проблема разбивается на более мелкие задачи, каждая из которых сохраняет все свойства более крупной проблемы, кроме своего размера.

Алгоритм делит входной массив на две половины, вызывает себя для этих двух половин, а затем объединяет две отсортированные половины.

1. Сортировка
  - Разделяем массив, пока не достигнем  $\text{length} = 1$
2. Слияние
  - Слияние отсортированных элементов.

## Алгоритм

Шаг 1 : Разделите несортированный список на  $n$  подсписков; каждый подсписок имеет один элемент (который сортируется по одному элементу)

Шаг 2 : Объедините два списка одновременно. При объединении сравните элементы для двух подсписков и подготовьте новый отсортированный список.

Шаг 3. Выполните шаг 2 для всех пар подсписков.

Шаг 4 : Повторяйте шаги со 2 по 3 для нового списка подсписков, пока у нас не будет только один список.

# How MergeSort Algorithm Works Internally

1. Divide the array into two parts

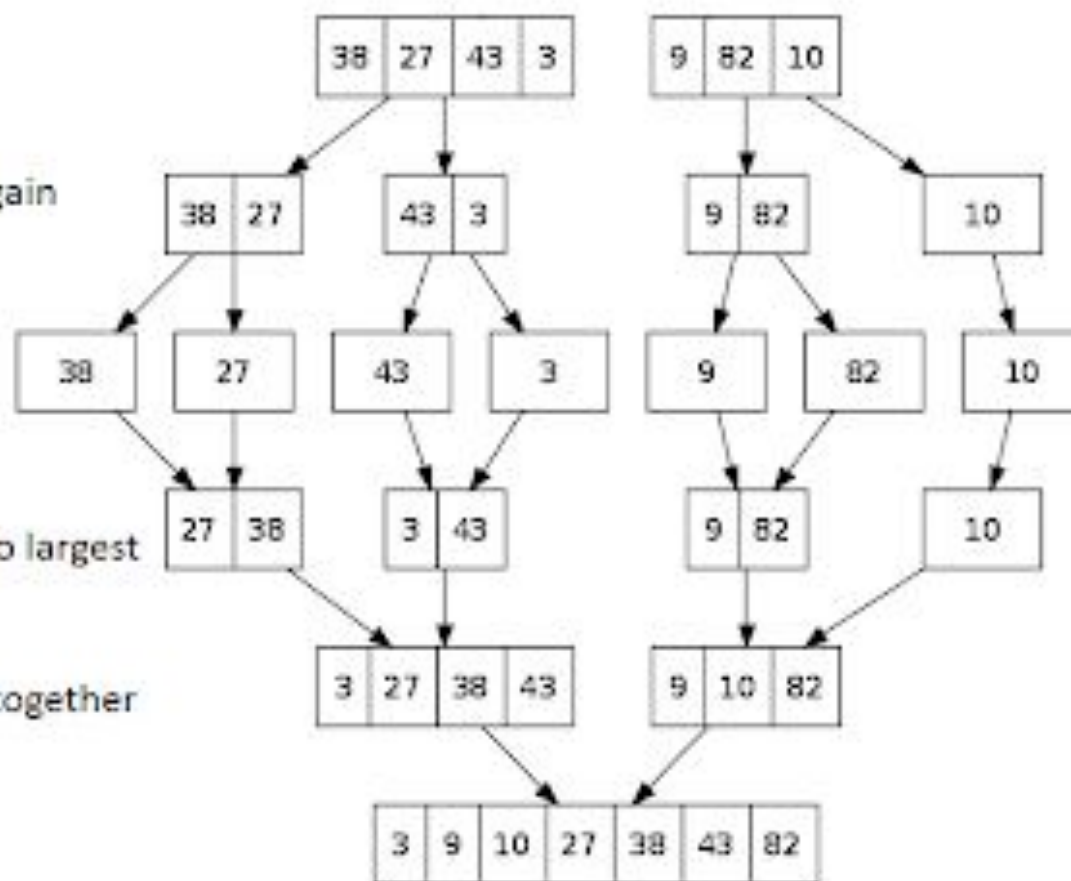
2. Divide the array into two parts again

3. Break each element into single parts

4. Sort the elements from smallest to largest

5. Merge the divided sorted arrays together

6. The array has been sorted



## Merge sort (быстрая сортировка)

**mergeSort**(array[], leftIndex, rightIndex)

если leftIndex < rightIndex

1. Находим среднюю точку, чтобы разделить массив на две половины:  
middle = (leftIndex + rightIndex)/2
2. Вызов mergeSort для первой половины:  
Вызов mergeSort(array, leftIndex, middle)
3. Вызов mergeSort для второй половины:  
Вызов mergeSort(array, middle + 1, rightIndex)
4. Объедините две половины, отсортированные на шаге 2 и 3:  
Вызов merge(array, leftIndex, middle, rightIndex)

// Объединим два отсортированных массива

**merge**(array[], leftIndex, middle, rightIndex)

left\_i = 1, right\_i = middle

while(left\_i < leftIndex AND right\_i < rightIndex)

if(element at left\_i > element at right\_i)

left\_i ++

else if(element at left\_i < element at right\_i)

left\_i ++ and right\_i ++

## Merge sort VS Quick sort

Quick sort	Merge sort
Не обязательно на половину	Всегда на половину
$O(n^2)$	$O(n \log n)$
Меньший массив	Любой массив
Быстрее, чем другие алгоритмы для небольшого набора	Постоянная скорость
Требует меньше места	Требует больше места
Не эффективен для больших массивов	Эффективен для больших массивов