

Задание ПРО по теме «4.6 Углубленное изучение Scala, принципы функционального программирования»

В рамках задания проведено тестирование четырех алгоритмов сортировки:

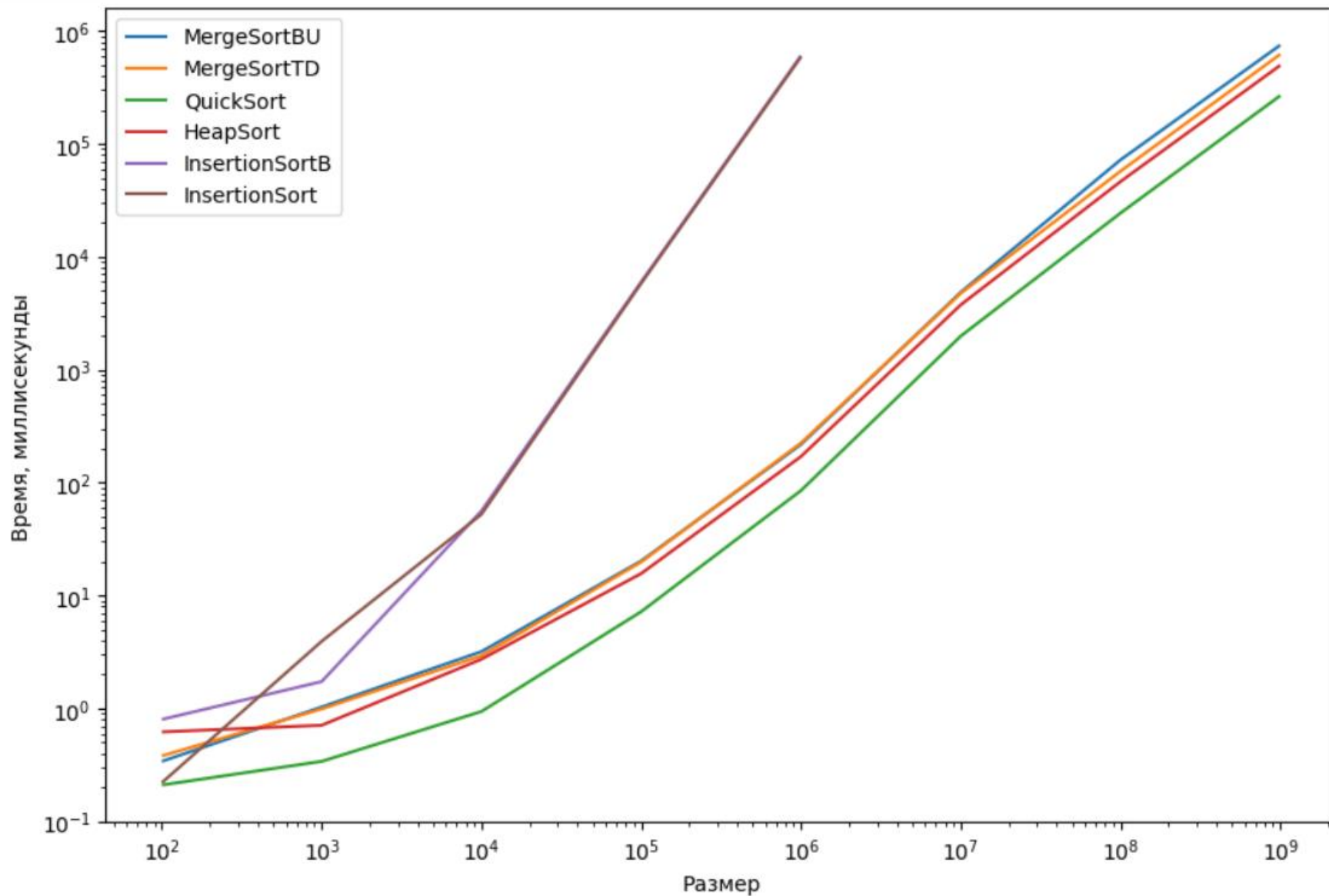
- Сортировка слиянием (merge sort) – 2 варианта:
 - Снизу вверх (Bottom-Up)
 - Сверху вниз (Top-Down)
- Быстрая сортировка (quick sort)

Существуют разные варианты реализации быстрой сортировки. В задании использовался вариант, при котором опорный элемент может присутствовать как в левой, так и в правой частях разбиения. Алгоритм на каждом шаге предполагает два рекурсивных вызова, однако второй вызов можно заменить на эмуляцию хвостовой рекурсии. (Автоматически компилятор это делать отказывается из-за наличия первого рекурсивного вызова.)
- Пирамидальная сортировка (сортировка кучей, heap sort)
- Сортировка вставками (Insertion Sort) – 2 варианта:
 - С бинарным поиском места вставки
 - Обычный вариант

Методика тестирования

На вход алгоритмам подавались массивы целых чисел (`ArrayBuffer[Int]`) размером 10^n с n от 2 до 9, сгенерированные функцией `scala.util.Random.nextInt(valRange)` с `valRange = 10` и `2_000_000_000` (чтобы протестировать алгоритмы на повторяющихся и не повторяющихся данных). Для каждого набора параметров алгоритм запускался 3 раза и результат усреднялся. Из тестирования на размерах `10_000_000`, `100_000_000` и `1_000_000_000` был исключен алгоритм сортировки вставками ввиду его квадратичной зависимости. Усредненные результаты сохранялись в csv-файлы и визуализировались в Seaborn. Ввиду большого разброса результатов использовалась логарифмическая шкала.

valRange = 10



valRange = 2_000_000_000

