**А1. Анализ строковых сортировок**

**Задание.**

*Вам необходимо провести практическое исследование, в ходе которого нужно сравнить работу обычных и модифицированных алгоритмов сортировки для массивов строк. Сортировка должна выполняться в прямом лексикографическом порядке.*

Ниже приведены графики времени работы алгоритмов и количество посимвольных сравнений. Слева графики зависимости времени работы алгоритмов от количества элементов массива, справа графики зависимости количества посимвольных сравнений от количества элементов массива. Первая строка графиков – демонстрация экспериментов на случайном массиве, вторая строка – отсортированный массив в обратном порядке, третья строка – почти полностью отсортированный массив.

По графикам зависимости времени работы алгоритмов от количества элементов массива видно, что стандартный алгоритм Quick Sort и алгоритм MSD Radix Sort работают заметно хуже других алгоритмов, примерно за . Алгоритмы String Quick Sort и алгоритм MSD Radix Sort with String Quick Sort работают примерно в 2 раза быстрее, чем алгоритмы MergeSort и String MergeSort.

По графикам зависимости количества посимвольных сравнений от количества элементов массива видно, что стандартный алгоритм Quick Sort требует заметно больше посимвольных сравнение, чем другие алгоритмы, примерно в 5 раз. Алгоритмы типа Radix Sort требуют меньше всех посимвольных сравнений, но требуют выделения большего объема памяти для работы алгоритма.

ID посылок:

A1m: [320226071](https://dsahse.contest.codeforces.com/group/NOflOR1Qt0/contest/608956/submission/320226071)

A1q: [320226278](https://dsahse.contest.codeforces.com/group/NOflOR1Qt0/contest/608956/submission/320226278)

A1r: [320226409](https://dsahse.contest.codeforces.com/group/NOflOR1Qt0/contest/608956/submission/320226409)

A1rq: [320226645](https://dsahse.contest.codeforces.com/group/NOflOR1Qt0/contest/608956/submission/320226645)

Исходный код, данные, картинки лежат в каталоге code.





