

ФЕДЕРАЛЬНОЕ АГЕНСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ
СООБЩЕНИЯ Императора Александра I»

Кафедра «Информационные и вычислительные системы»
Дисциплина «Структуры и алгоритмы обработки данных»

ОТЧЁТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1
«Реализация и анализ алгоритмов сортировки»

Выполнил студент
Факультет: АИТ
Группа: ИВБ-417

Иорданов Н. Г.

Проверил: Доцент кафедры ИВС к. и. н.

Забродин А. В.

Санкт-Петербург

2025

Цель:

Разработать и реализовать алгоритмы сортировки данных, провести их сравнительный анализ по времени выполнения и оценить сложность каждого алгоритма в терминах O -символики.

Задания:

1. Реализовать следующие алгоритмы сортировки:

- Сортировка выбором
- Пузырьковая сортировка
- Сортировка вставками
- Сортировка слиянием
- Пирамидальная сортировка
- Быстрая сортировка: сайт Архив погоды с 1929 года pogoda-service.ru
- Лексикографическая сортировка (входными данными может быть журнал, где сделать сортировку по имени, отчеству, n -ой букве фамилии и т.д.)

2. Анализ:

- Время выполнения: для каждого алгоритма измерьте время выполнения сортировки на полученных данных.
- O -символика: оцените и укажите сложность каждого алгоритма в терминах O -символики.
- Диаграмма: Постройте диаграмму, отображающую скорость выполнения каждой сортировки для разных наборов данных.

Описание алгоритмов сортировки.

Сортировка выбором

На каждом шаге алгоритма находится минимальный (или максимальный) элемент в неупорядоченном сегменте массива и переставляется в его начало. Таким образом, после каждой итерации размер отсортированной части увеличивается, а неотсортированной — сокращается. Сложность алгоритма: $O(n^2)$.

Пузырьковая сортировка

При пузырьковой сортировке последовательно сравниваются соседние элементы массива. Если они расположены в неправильном порядке, происходит их обмен. В результате каждого прохода самый большой (или маленький) элемент «всплывает» к концу списка, занимая свою конечную позицию. Сложность алгоритма: $O(n^2)$.

Сортировка вставками

Алгоритму на вход подается массив ключей a_1, a_2, \dots, a_n , который требуется преобразовать в отсортированную перестановку $a_1' \leq a_2' \leq \dots \leq a_n'$. Изначально результирующая последовательность пуста. На каждом этапе алгоритм извлекает очередной ключ из входного набора и вставляет его на корректную позицию в уже сформированной отсортированной части. Это продолжается до полного переноса всех элементов, причем на любом этапе отсортированная часть остается упорядоченной. Сложность алгоритма: $O(n^2)$.

Сортировка слиянием

Данный алгоритм применяет принцип «разделяй и властвуй».

Исходный массив рекурсивно делится на две половины до тех пор, пока не останутся подмассивы размером в один элемент. Затем эти упорядоченные подмассивы сливаются в один большой отсортированный массив. Процесс слияния заключается в попарном сравнении первых элементов каждого подмассива и перемещении наименьшего (или наибольшего) в результирующий массив, с последующим смещением указателя в том подмассиве, откуда был взят элемент. Операция повторяется, пока все элементы не будут перенесены. Сложность алгоритма: $O(n \log n)$.

Пирамидальная сортировка

Пирамида (или двоичная куча) — это бинарное дерево, где значение в любом узле не меньше (или не больше) значений его потомков. Алгоритм сортировки состоит из следующих шагов:

Исходный массив преобразуется в пирамиду, в вершине которой оказывается наибольший (наименьший) элемент.

Этот элемент извлекается из пирамиды и помещается в конец итогового массива.

На его место ставится последний элемент из кучи, после чего структура пирамиды восстанавливается (просеивается). Шаги 2-3 повторяются для уменьшающейся пирамиды, пока все элементы не будут перенесены в результирующий массив в правильном порядке. Сложность алгоритма: $O(n \log n)$.

Быстрая сортировка

В основе этого алгоритма также лежит парадигма «разделяй и властвуй». Его работа начинается с выбора опорного элемента. Затем массив перераспределяется так, чтобы все элементы, меньшие опорного, оказались слева от него, а все большие — справа. После этого рекурсивно применяется та же процедура к двум полученным подмассивам. В конечном счете,

отсортированные подмассивы объединяются в единый упорядоченный массив. Сложность алгоритма: $O(n \log n)$.

Диаграммы.

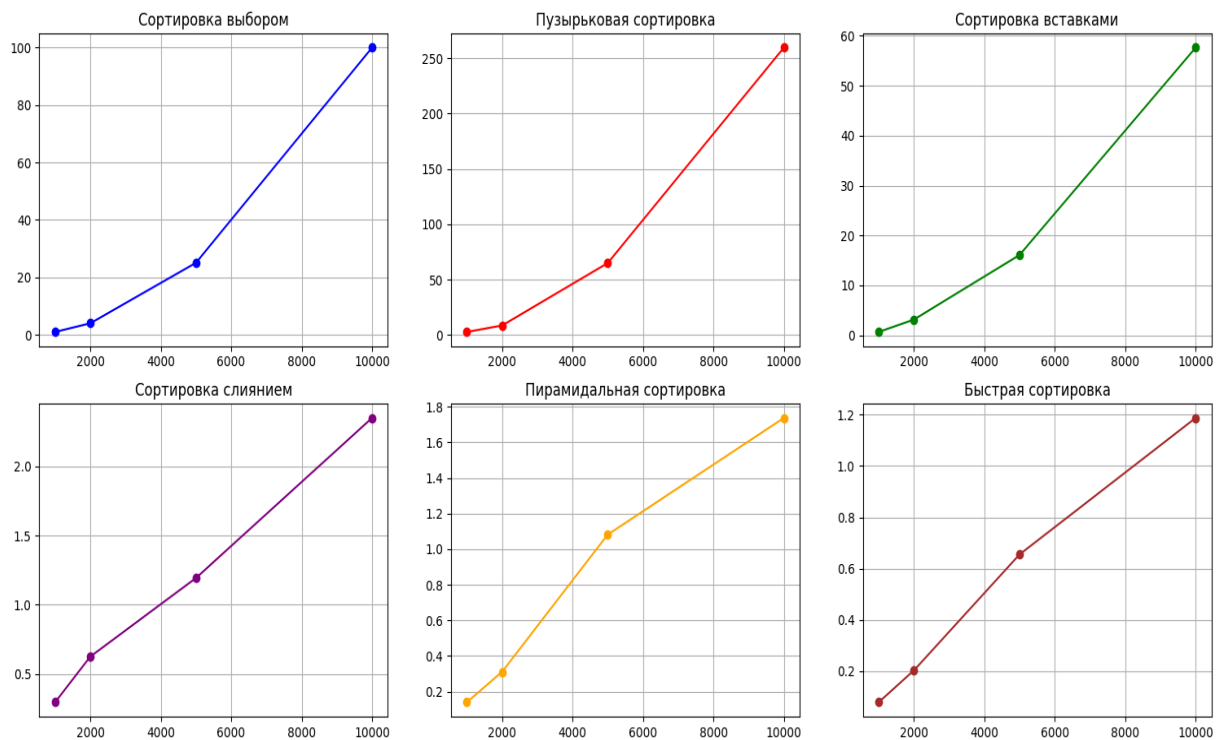


Рис. 1 - наглядное сравнение данных до 10000

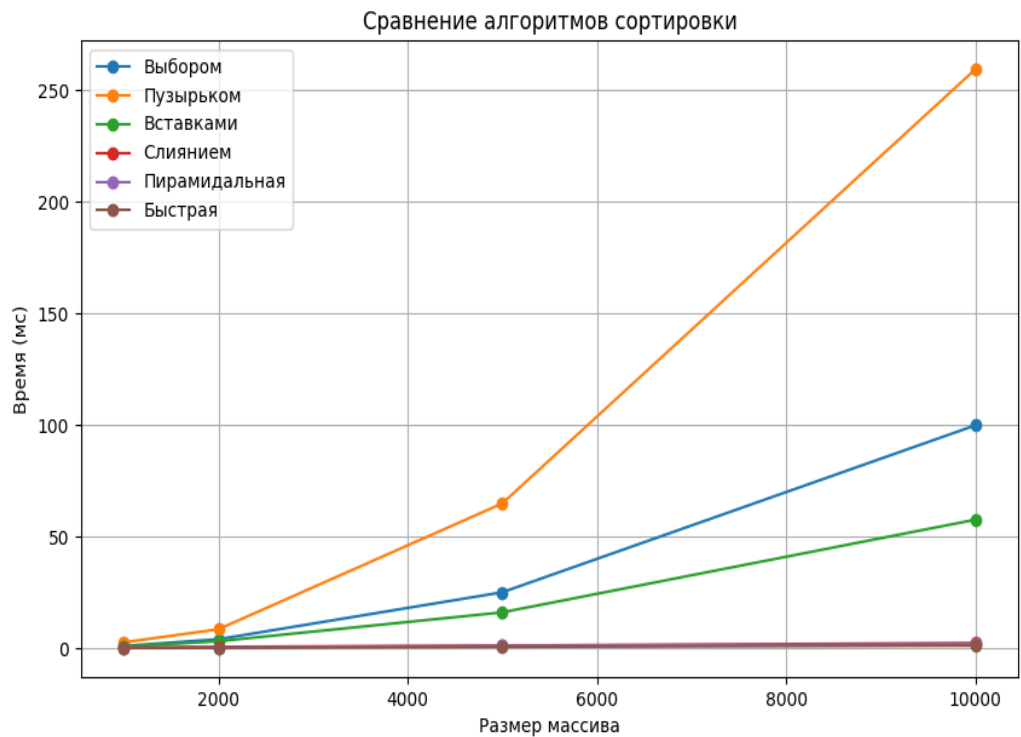


Рис. 2 - Сравнение на одной плоскости

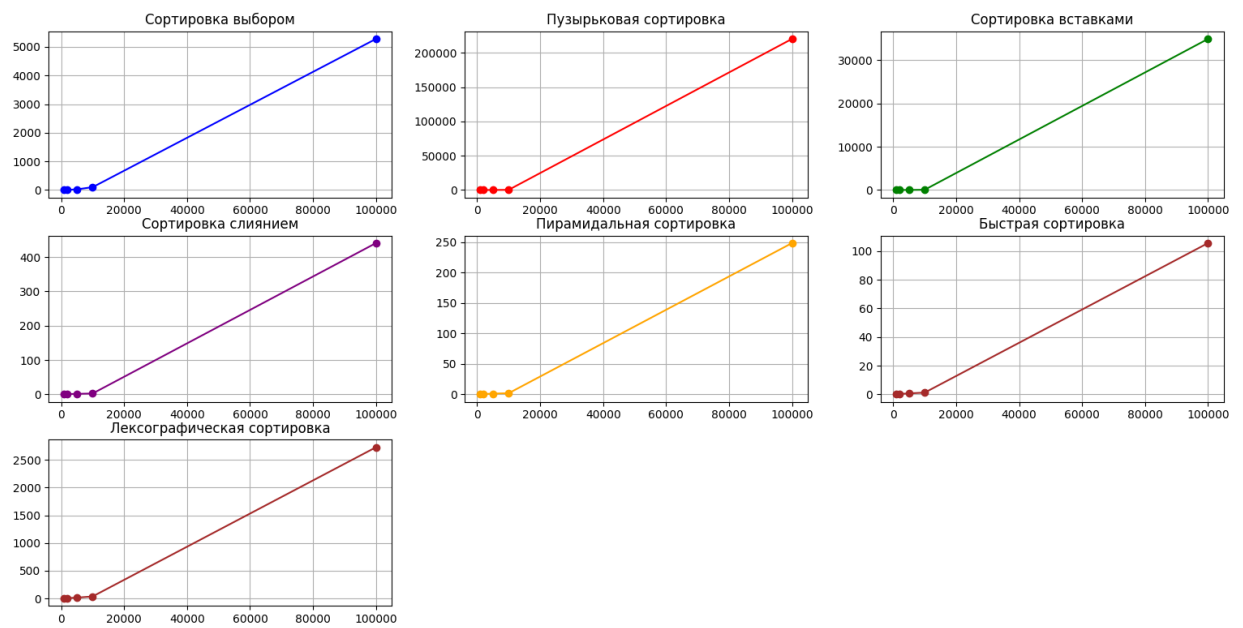


Рис. 3 - График при 100000 элементов

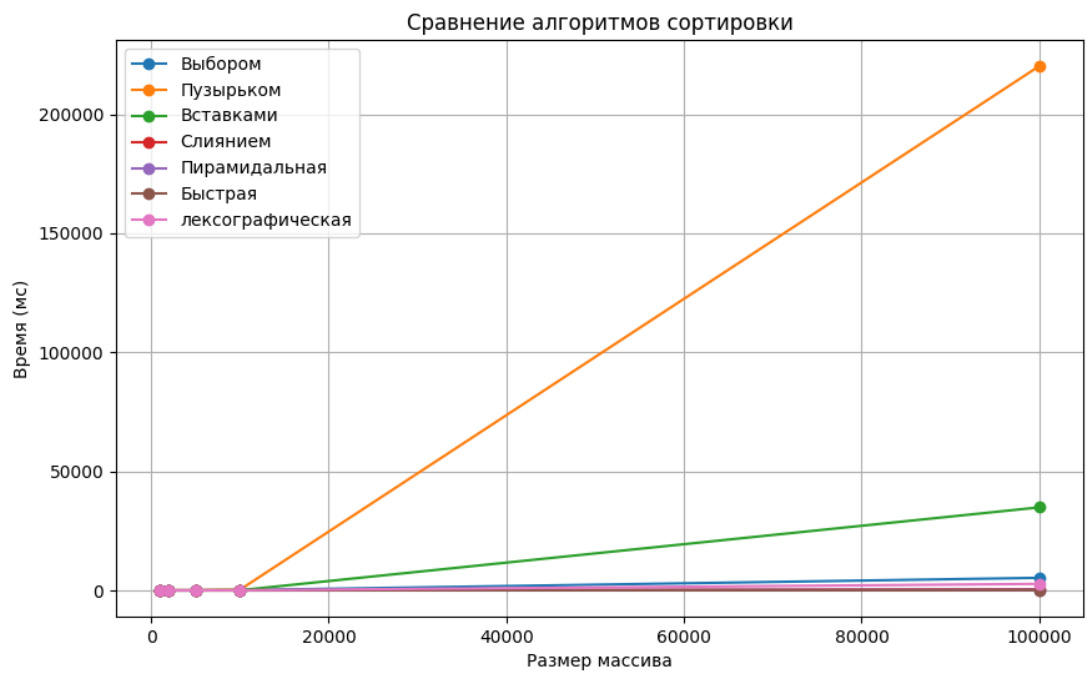


Рис. 4 - Графики на одной плоскости

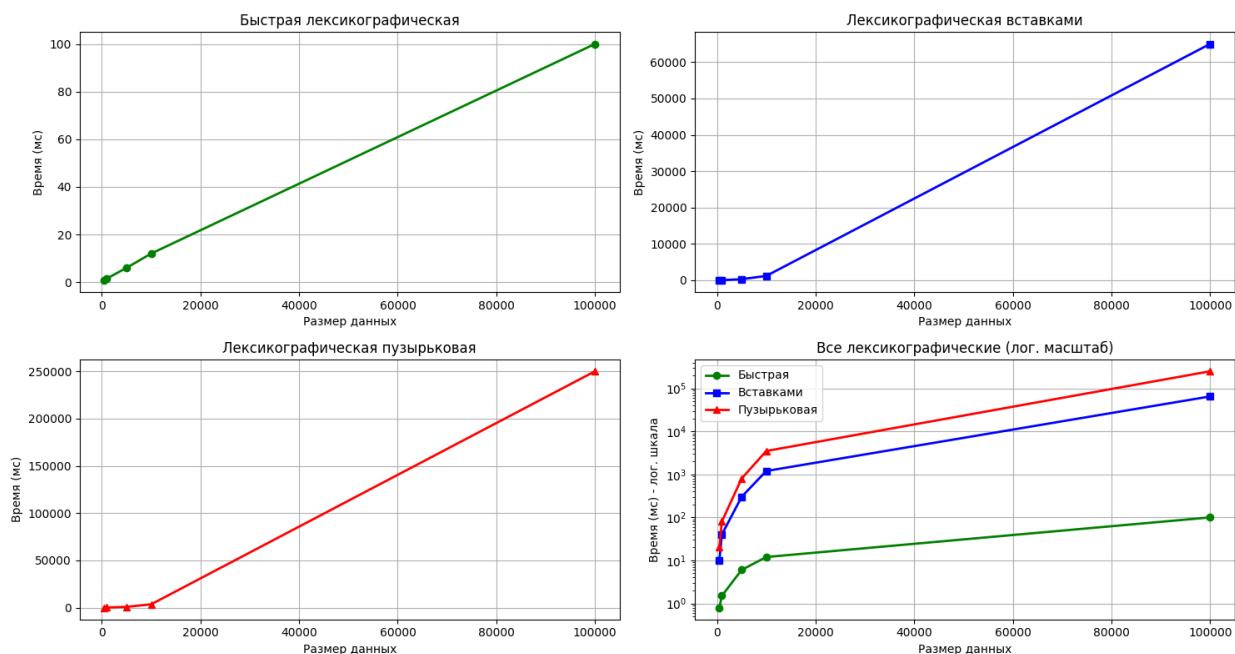


Рис. 5 - сравнение разных лексикографических сортировок

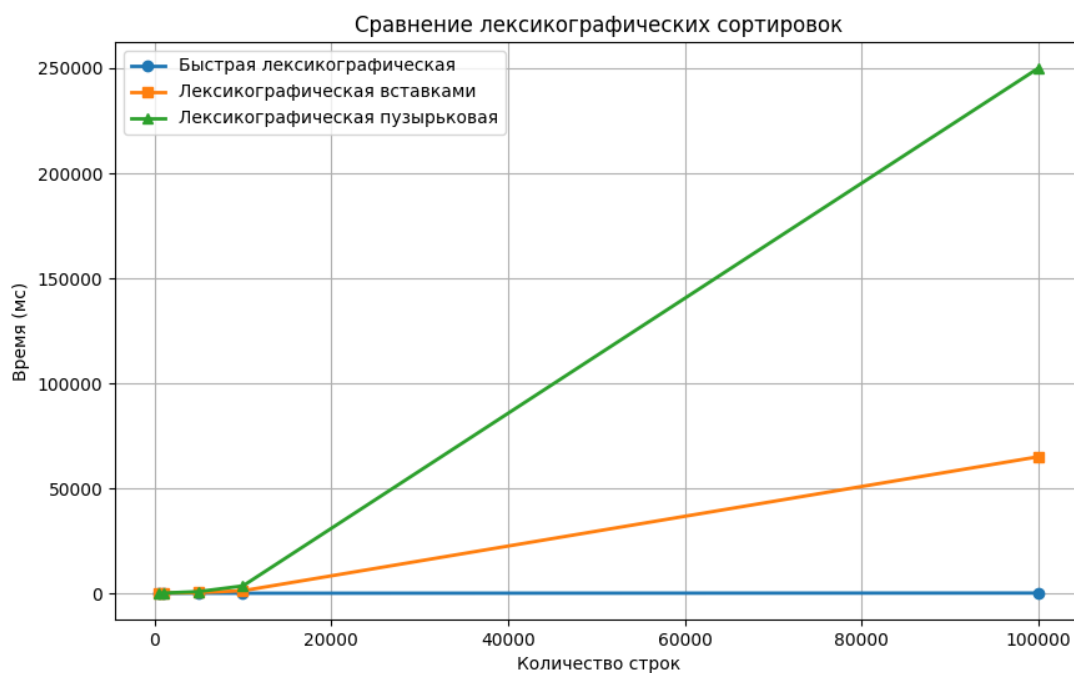


Рис. 6 – лексикографические сортировки на одной диаграмме

Заключение.

В процессе выполнения лабораторной работы были реализованы алгоритмы сортировки данных, проведен их сравнительный анализ по времени выполнения и оценена вычислительная сложность каждого алгоритма в рамках O -нотации.