

# Лабораторная работа 1. Эффективность алгоритмов сортировки

Кузин Максим Витальевич

15 декабря 2025 г.

## Содержание

<b>1 Введение</b>	<b>1</b>
1.1 Эксперимент . . . . .	1
<b>2 Алгоритмы с асимптотикой <math>O(n^2)</math></b>	<b>2</b>
2.1 Демонстрация . . . . .	2
2.2 Сравнение оптимизаций . . . . .	3
<b>3 Алгоритмы с асимптотикой <math>O(N \log N)</math></b>	<b>4</b>
3.1 Демонстрация . . . . .	4
3.2 Сравнение оптимизаций . . . . .	5
<b>4 Сравнение <math>O(N \log N)</math> и <math>O(N^2)</math></b>	<b>6</b>
<b>5 Зависимость от начального состояния данных</b>	<b>6</b>
<b>6 Вывод</b>	<b>7</b>

## 1 Введение

В рамках данной лабораторной работы проводится исследование различных алгоритмов сортировки с точки зрения их вычислительной эффективности и асимптотической сложности. Рассматривается влияние объёма входных данных, а также их начальной упорядоченности, на время выполнения алгоритмов. В ходе работы анализируются как простые методы сортировки, так и более производительные алгоритмы, применяемые на практике, а именно: пузырьковая сортировка, сортировка выбором, сортировка вставками, быстрая сортировка, сортировка кучей и сортировка слиянием.

Основной задачей лабораторной работы является закрепление теоретических знаний об асимптотической сложности алгоритмов, а также получение практического опыта в их программной реализации и экспериментальном сравнении. Проведённый анализ позволяет наглядно показать, насколько критичен правильный выбор алгоритма при обработке больших массивов данных в реальных вычислительных задачах.

### 1.1 Эксперимент

Экспериментальные измерения выполнялись на персональном компьютере, оснащённом процессором Intel i5. Температурный режим поддерживался постоянным.

## 2 Алгоритмы с асимптотикой $O(n^2)$

### 2.1 Демонстрация

В первую очередь рассмотрим алгоритмы сортировки, отличающиеся простотой реализации, но обладающие квадратичной временной сложностью.

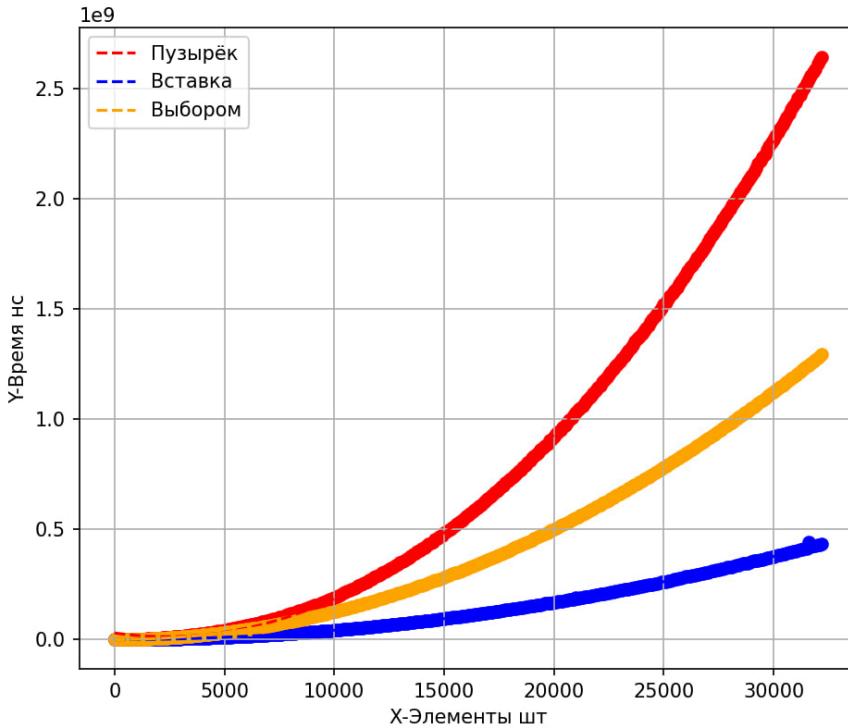


Рис. 1: Асимптотическая сложность  $O(n^2)$

Для подтверждения квадратичного характера роста времени выполнения используется логарифмическое представление данных:

$$\ln(t) = \ln(C) + 2 \ln N.$$

Графики в логарифмическом масштабе имеют линейный вид, что подтверждает квадратичную зависимость времени выполнения от размера входных данных.

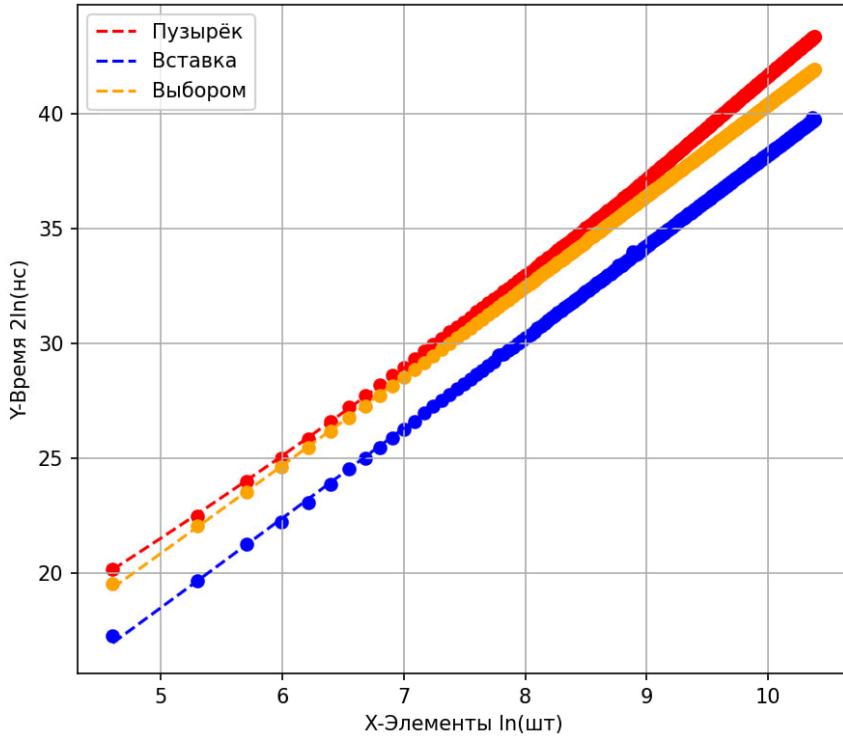


Рис. 2: Подтверждение сложности  $O(n^2)$  в логарифмическом масштабе

## 2.2 Сравнение оптимизаций

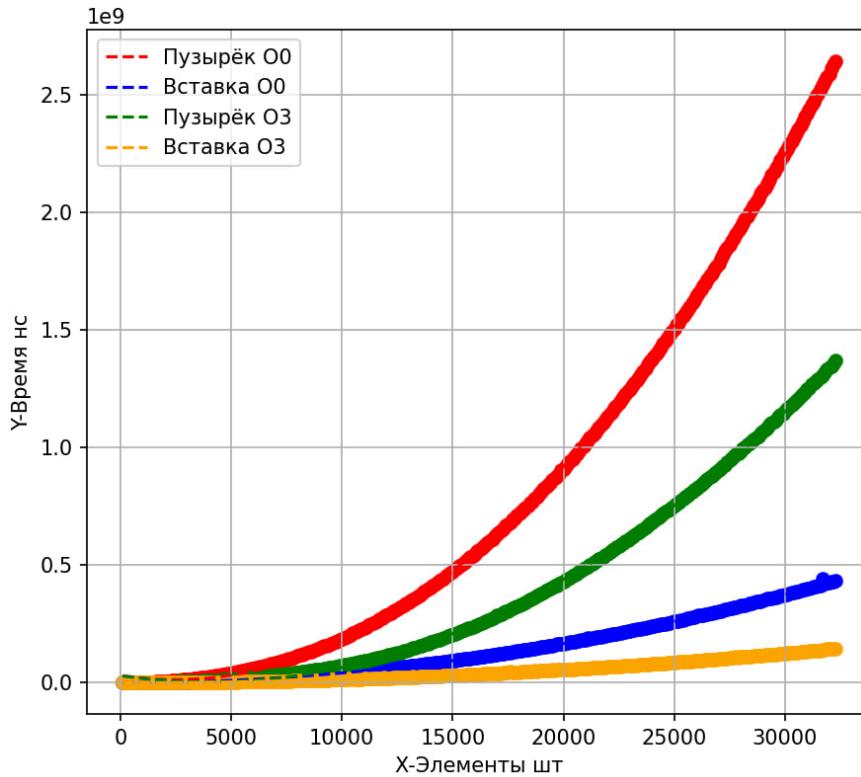


Рис. 3: Влияние оптимизаций на алгоритмы с квадратичной сложностью

Даже при применении оптимизаций алгоритмы данного класса остаются малоэффективными при увеличении размера массива.

### 3 Алгоритмы с асимптотикой $O(N \log N)$

#### 3.1 Демонстрация

Далее рассмотрим более эффективные алгоритмы сортировки и сравним их между собой.

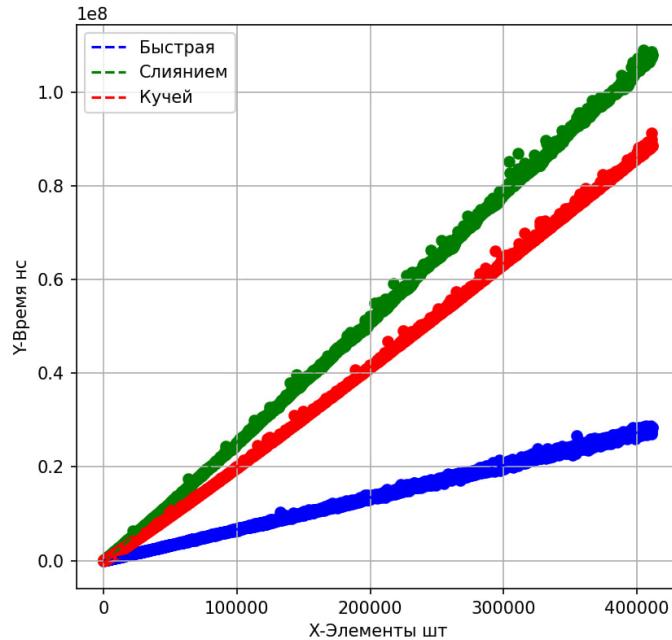


Рис. 4: Асимптотическая сложность  $O(N \log N)$

Для проверки логарифмической зависимости используется следующая нормализация:

$$t = \frac{t}{N \ln N}.$$

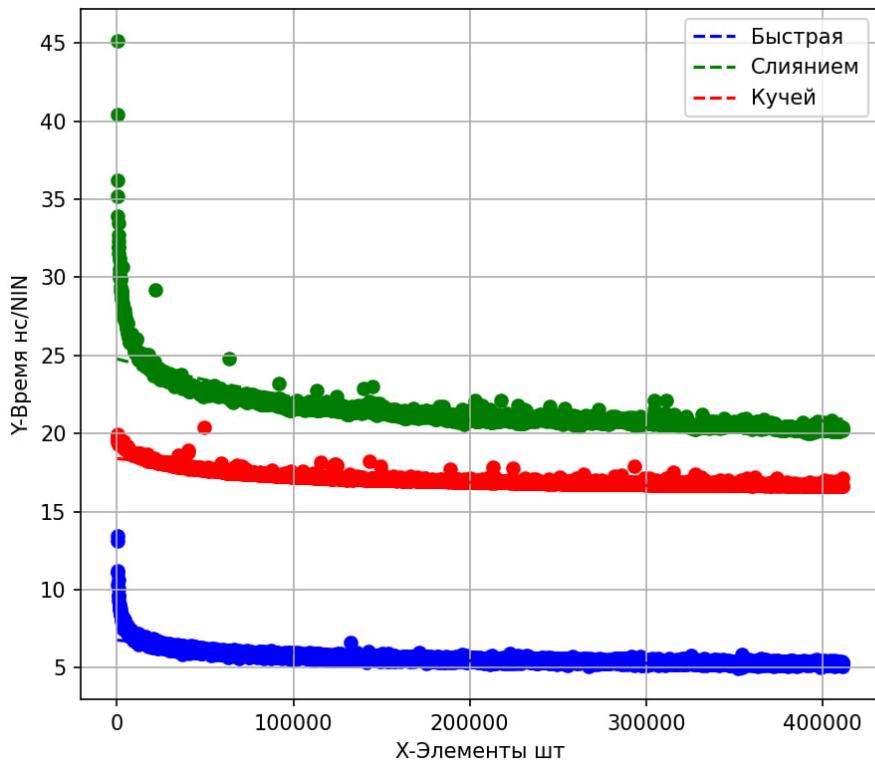


Рис. 5: Подтверждение сложности  $O(N \log N)$

Полученные графики имеют близкий к линейному характер, что подтверждает соответствие экспериментальных данных теоретической оценке сложности.

### 3.2 Сравнение оптимизаций

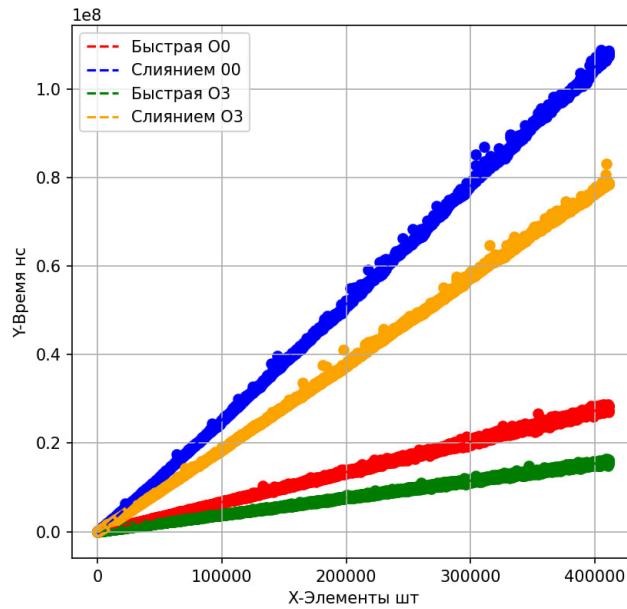


Рис. 6: Эффект оптимизаций для алгоритмов  $O(N \log N)$

Быстрая сортировка демонстрирует наилучшие показатели производительности: даже без оптимизаций она опережает остальные алгоритмы данного класса.

## 4 Сравнение $O(N \log N)$ и $O(N^2)$

Для наглядного сопоставления алгоритмов различных классов сложности были отключены все оптимизации.

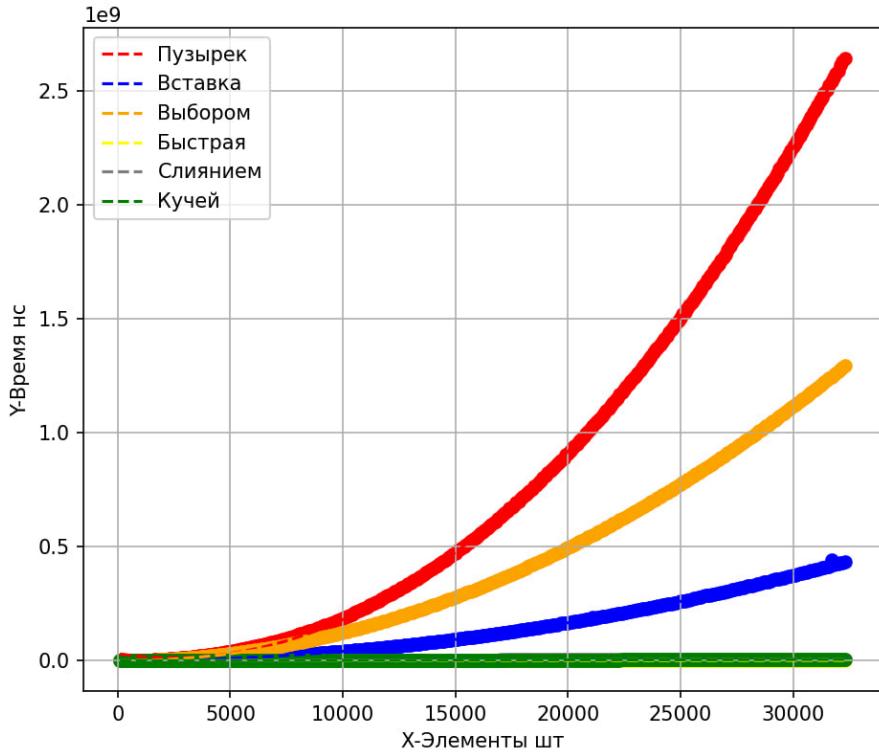
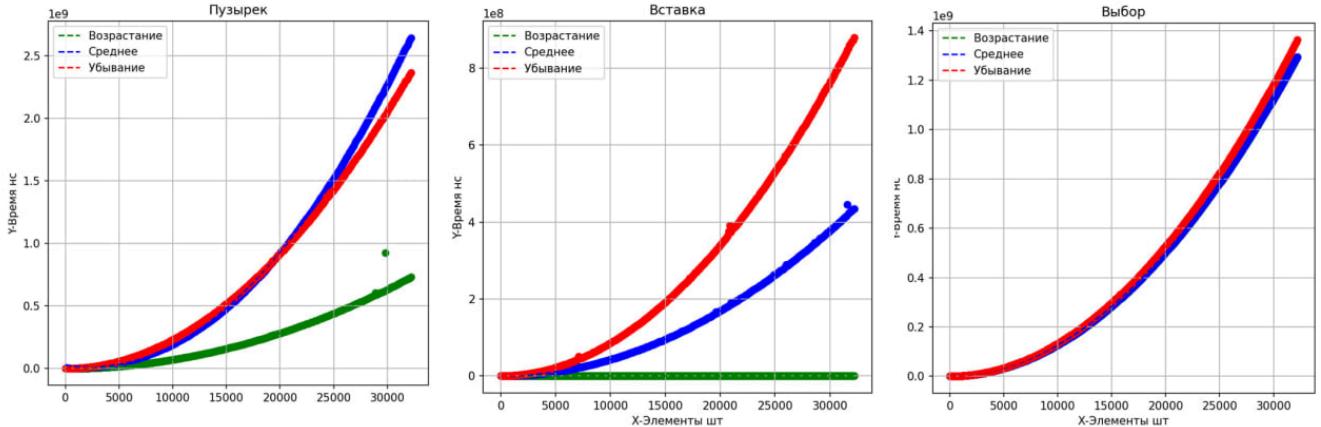


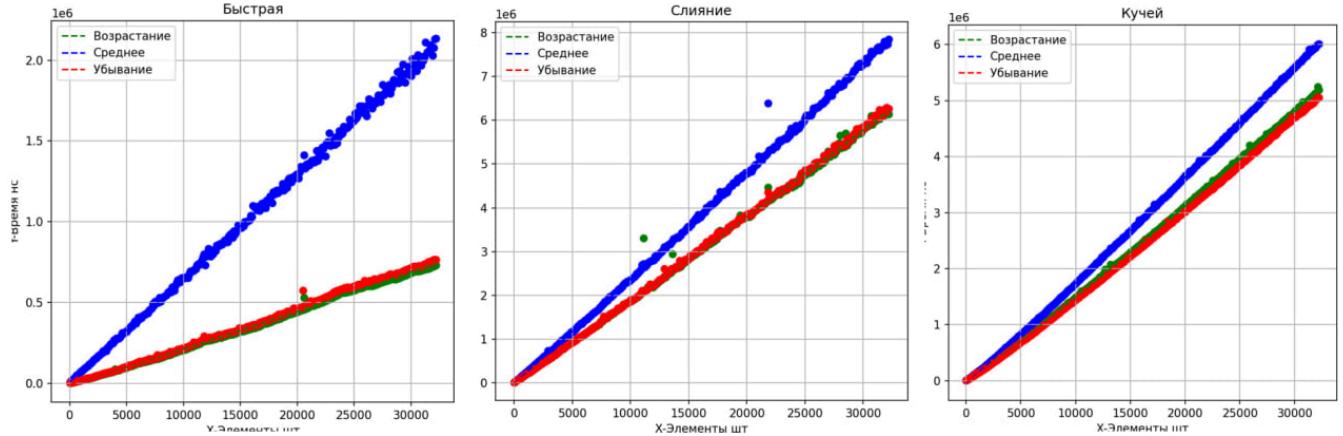
Рис. 7: Сравнение всех исследуемых алгоритмов сортировки

Различие в скорости работы алгоритмов становится особенно заметным при росте размера входных данных.

## 5 Зависимость от начального состояния данных

Для анализа влияния структуры входных данных использовались три типа массивов: случайно упорядоченные, отсортированные по возрастанию и отсортированные по убыванию.





Результаты показывают, что начальное расположение элементов может существенно влиять на время работы отдельных алгоритмов.

## 6 Вывод

В процессе выполнения лабораторной работы была проведена экспериментальная и теоретическая оценка эффективности различных алгоритмов сортировки, включая пузырьковую сортировку, сортировку выбором, сортировку вставками, быструю сортировку, сортировку кучей и сортировку слиянием.

Было исследовано время выполнения алгоритмов на массивах различной структуры: случайных, упорядоченных и обратных. Эксперимент подтвердил, что временная сложность существенно зависит как от размера данных, так и от их начального состояния.

Алгоритмы с квадратичной сложностью  $O(n^2)$ , такие как пузырьковая сортировка и сортировка выбором, показали крайне низкую эффективность при работе с большими массивами и практически непригодны для реальных задач.

Алгоритмы с асимптотикой  $O(N \log N)$  продемонстрировали значительно более высокую производительность. Среди них быстрая сортировка оказалась наиболее эффективной во всех проведённых тестах.

Полученные результаты наглядно показывают важность анализа алгоритмической сложности и обоснованного выбора алгоритмов при разработке программного обеспечения и оптимизации вычислительных процессов.