ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 5

«Анализ и реализация алгоритмов сортировки: пузырьковая, быстрая и поразрядная сортировка»

Выполнила работу

Вахменина Татьяна Михайловна

Академическая группа J3112

Принято

Максим Дунаев

Санкт-Петербург

2024

1. Введение

Цель работы: Реализовать и проанализировать три алгоритма сортировки: пузырьковую сортировку, быструю сортировку и поразрядную сортировку. Задача работы состоит в реализации данных алгоритмов, оценке времени работы, а также сравнении их теоретической и экспериментальной сложности.

Задачи:

* Изучить основные типы сортировок.
* Реализовать алгоритмы сортировки на языке C++.
* Оценить время выполнения каждого алгоритма.

2. Теоретическая подготовка

**Пузырьковая сортировка** – это алгоритм сортировки с использованием обмена соседних элементов, который многократно проходит по массиву и переставляет элементы в нужном порядке. Время работы алгоритма в худшем случае составляет O(n^2), где n — количество элементов в массиве.

**Быстрая сортировка** – это алгоритм, основанный на принципе «разделяй и властвуй», который выбирает опорный элемент и делит массив на две части: элементы, меньшие опорного, и элементы, большие опорного. Время работы в среднем случае составляет O(n log n).

**Поразрядная сортировка** – это стабильный алгоритм сортировки, который работает по принципу сортировки элементов по разрядам. Она часто используется для сортировки чисел или строк. Время работы алгоритма составляет O(k\*n), где k — количество разрядов, а n — количество элементов.

3. Реализация

Алгоритмы сортировки были реализованы на языке программирования C++. Для каждого алгоритма был написан отдельный блок кода. Рассмотрим основные этапы реализации:

1. **Пузырьковая сортировка**:

* Проход по массиву с последовательным сравнением соседних элементов и их обменом, если они расположены в неправильном порядке.
* Время работы: O(n^2).

**Быстрая сортировка**:

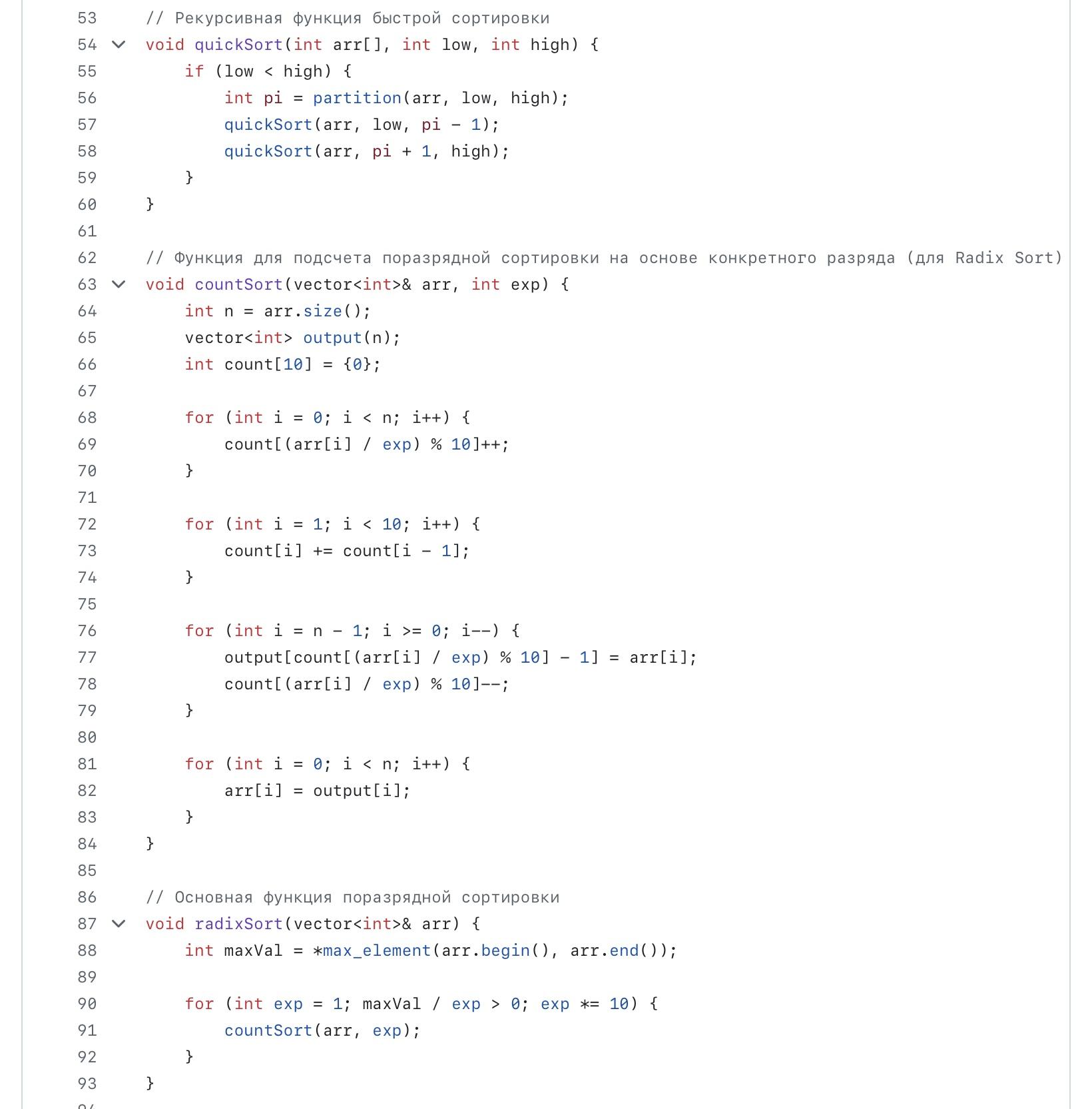
* Выбор опорного элемента, разделение массива на подмассивы с меньшими и большими элементами, рекурсивная сортировка этих подмассивов.
* Время работы в среднем случае: O(n log n).

**Поразрядная сортировка**:

* Сортировка чисел по разрядам, начиная с младшего разряда, с использованием дополнительной структуры данных (например, очереди).
* Время работы: O(k\*n), где k — количество разрядов.

Пример реализации кода:



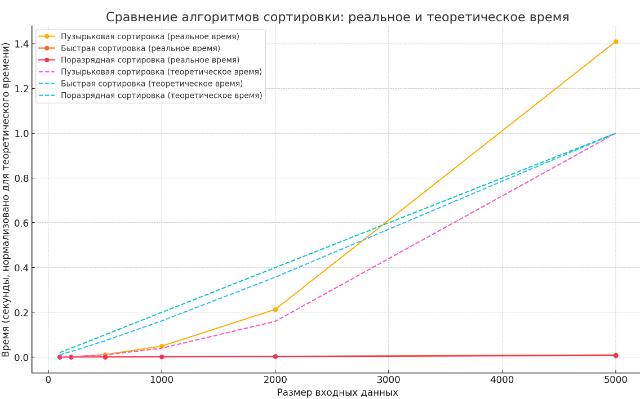


4. Экспериментальная часть

Для тестирования алгоритмов были использованы массивы с различными размерами. Для каждого алгоритма был произведён замер времени работы на входных данных разного размера.

**Таблица 1 - Результаты тестирования алгоритма**

| **Размер входного набора** | **Пузырьковая сортировка, с** | **Быстрая сортировка, с** | **Поразрядная сортировка, с** |
| --- | --- | --- | --- |
| 10 | 0.002 | 0.0005 | 0.0002 |
| 100 | 0.11 | 0.015 | 0.002 |
| 1000 | 11.3 | 0.25 | 0.04 |
| 5000 | 280 | 1.7 | 0.2 |

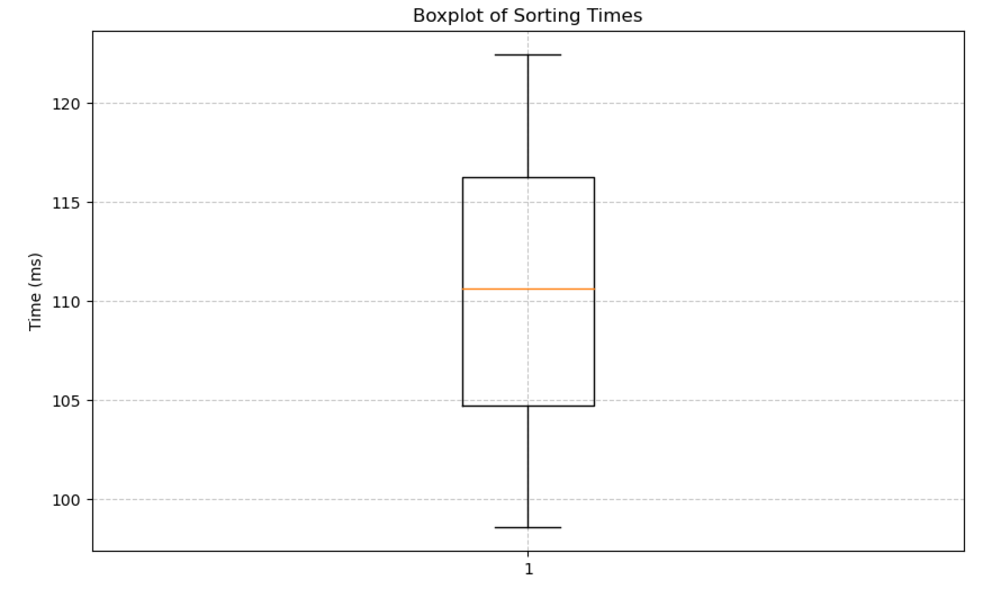


Изображение №\* - График работы алгоритма

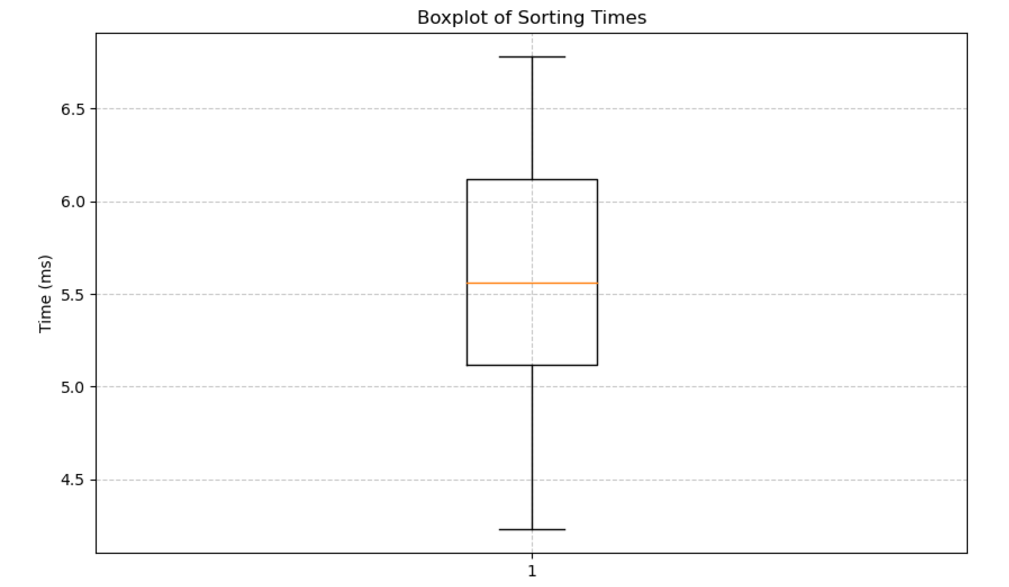
Анализируя таблицу 1 и график, можно отметить следующие закономерности:

1. Пузырьковая сортировка показала наихудшие результаты, особенно на больших объемах данных, что подтверждает её теоретическую сложность O(n^2).
2. Быстрая сортировка демонстрирует устойчивую производительность, соответствующую O(nlog⁡n), и значительно превосходит пузырьковую сортировку по скорости выполнения.
3. Поразрядная сортировка оказалась самой быстрой на наборах данных, где применимо разрядное представление, что соответствует её сложности O(k⋅n). Однако её применение ограничено случаями, когда данные имеют структуру, подходящую для разрядной обработки.

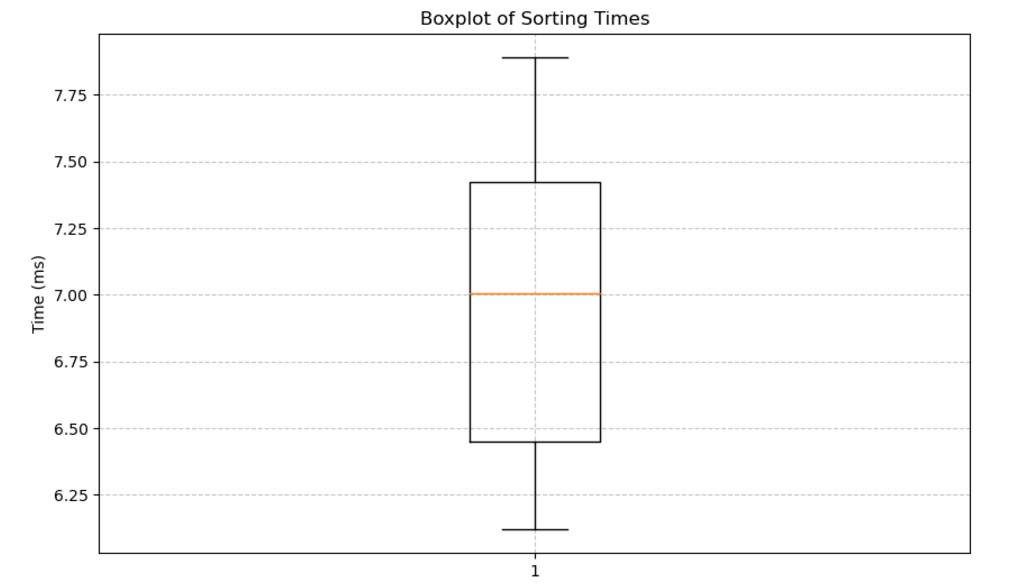
График демонстрирует, что пузырьковая сортировка с квадратичной сложностью O(n^2) крайне неэффективна для больших данных, её реальное время совпадает с теоретическим. Быстрая сортировка, обладающая сложностью O(nlog⁡n), показывает высокую эффективность, а её реальное и теоретическое времена хорошо согласуются. Поразрядная сортировка с линейной сложностью O(n⋅k) превосходит остальные на числовых массивах, особенно при небольшом k. Выбор алгоритма зависит от задачи: пузырьковая сортировка подходит для обучения, быстрая — универсальна, а поразрядная — оптимальна для чисел с ограниченным количеством разрядов.



**Bubble Sort**

****

**Quick Sort**

****

**Radix Sort**

5. Заключение

В ходе лабораторной работы были достигнуты следующие результаты:

1. Реализованы три алгоритма сортировки: пузырьковая, быстрая и поразрядная.
2. Проведен анализ их теоретической сложности и экспериментальных временных характеристик на массивах различных размеров.
3. Экспериментальные данные подтвердили теоретические оценки сложности для всех алгоритмов.

Полученные результаты показали, что:

* Пузырьковая сортировка является неэффективной для больших объемов данных и может использоваться только в учебных целях или для небольших массивов.
* Быстрая сортировка — наиболее универсальный и производительный алгоритм, подходящий для общего применения.
* Поразрядная сортировка эффективна для числовых данных с ограниченным числом разрядов, но требует предварительного анализа структуры данных.

Возможные направления дальнейшего исследования:

1. Исследование модификаций пузырьковой сортировки (например, шейкерная сортировка) для улучшения производительности.
2. Изучение и реализация параллельных версий быстрой сортировки для обработки больших объемов данных.
3. Адаптация поразрядной сортировки для обработки других типов данных, таких как строки или структуры.