ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 5

«Реализация трех алгоритмов сортировки»

Выполнила работу

Автономова Ксения Витальевна

Академическая группа №J3113

Принято

Ходненко Иван Владимирович

Санкт-Петербург

2024

**Структура отчёта:**

1. Экспериментальная часть

1 алгоритм – Comb Sort:

Подсчёт по памяти – 117 + 8n.

Подсчёт асимптотики – O(n log n).

2 алгоритм – Tournament Sort:

Подсчёт по памяти – 164 + 4k + 4m + 4p + 8n байта.

Подсчёт асимптотики – O(n log n).

3 алгоритм – Bucket Sort:

Подсчёт по памяти – 108 + 8n + 4k байт.

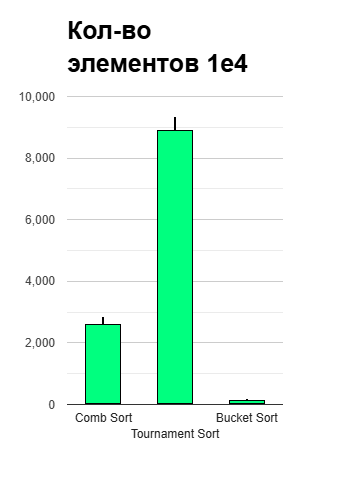
Подсчёт асимптотики – O(n + k).

Для тестирования трех алгоритмов сортировки была собрана статистика, приведенная в графике работы алгоритмов.

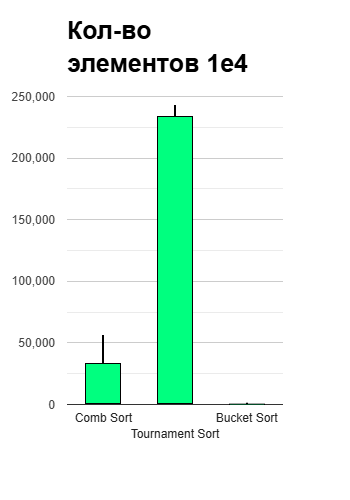
Изображение №1 - График работы алгоритмов.

Tournament Sort значительно медленнее, чем Comb Sort и Bucket Sort. Это негативно сказывается на производительности при обработке больших массивов данных. Чтобы результаты выполнения программ были более наглядными, я решила сократить диапазон значений в массиве с 1.000.000 до 400.000. Так, график лучше иллюстрирует разницу в скорости работы различных алгоритмов сортировки.

Ниже представлены box plot графики для времени работы алгоритмов с числом элементов 1e4 и 1e5 (2 разных графика). Количество запусков: 50 для каждого графика.



Изображение №2 – Box plot графики для 1e4 элементов в массиве.



Изображение №3 – Box plot графики для 1e5 элементов в массиве.

1. Заключение

Хотя теоретическая сложность указывает на логарифмическую зависимость, в практике Comb Sort быстрее по скорости, чем другой алгоритм с аналогичной асимптотикой, благодаря своей простоте и оптимизации. Подходит для ситуаций, где простота реализации важнее максимальной скорости.

Tournament Sort также имеет сложность O(n log n), но из-за дополнительных накладных расходов он работает медленнее на практике. Использование этого алгоритма довольно ограничено из-за более высокой временной сложности по сравнению с другими алгоритмами. Полезен в системах, где требуется многократное извлечение минимального или максимального элемента, но его использование ограничено из-за больших накладных расходов.

Bucket Sort показывает отличные результаты на практике благодаря своей линейной сложности, когда данные равномерно распределены. Он работает особенно хорошо для данных с известным диапазоном значений и значительно быстрее, чем другие 2 алгоритма. Идеален для сортировки больших объемов данных с известным диапазоном значений и равномерным распределением. Можно использовать для сортировки чисел или строк с фиксированной длиной.

Выбросы в производительности: Tournament Sort имеет дополнительные накладные расходы на управление структурой, что влияет на время выполнения.

1. Приложения

ПРИЛОЖЕНИЕ А

Код сортировки Comb Sort [LAB5\_1.cpp](LAB5_1/LAB5_1/LAB5_1.cpp)

Код сортировки Tournament Sort [LAB5\_2.cpp](LAB5_2/LAB5_2/LAB5_2.cpp)

Код сортировки Bucket Sort [LAB5\_3.cpp](LAB5_3/LAB5_3/LAB5_3.cpp)