**Лабораторная работа №1   
Курс: «Языки программирования и методы программирования»**

Исходная задача: реализовать класс Sequence<T>, написать программу на С++ для сравнения различных алгоритмов сортировки.

Были написаны сортировки:

1. Bubble Sort (сортировка пузырьком)
2. Cocktail Sort (сортировка шейкерная, улучшенный пузырек)
3. Insertion Sort (сортировка вставками)
4. Count Sort (сортировка подсчетом)
5. Quick Sort (сортировка Хоара)

Тестирование сортировок проходило на двух видах контейнеров:

1. Односвязный список
2. Динамический массив

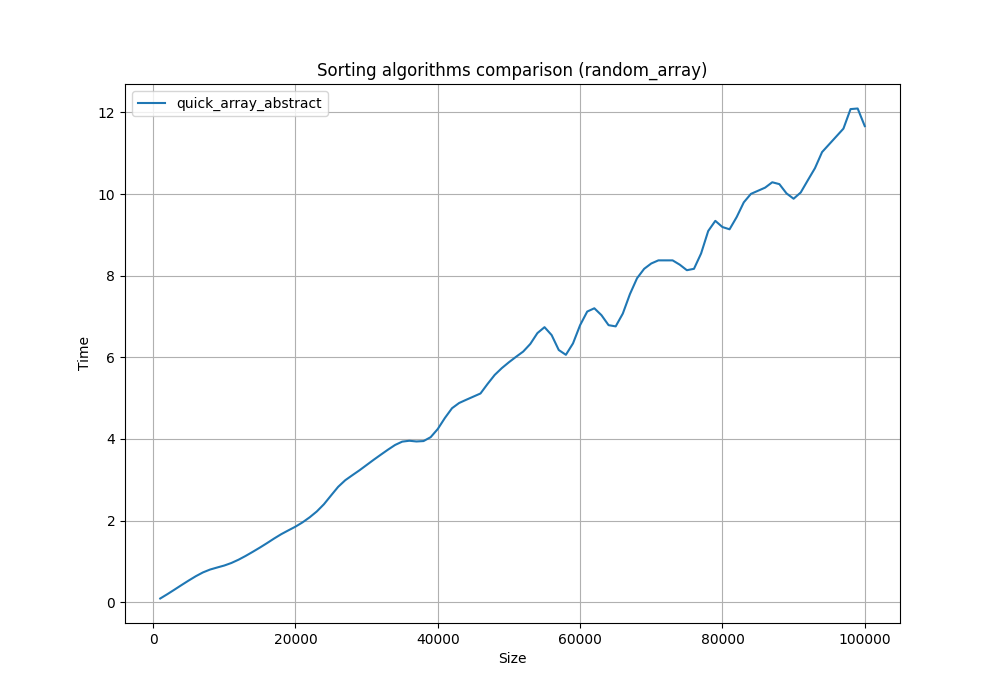
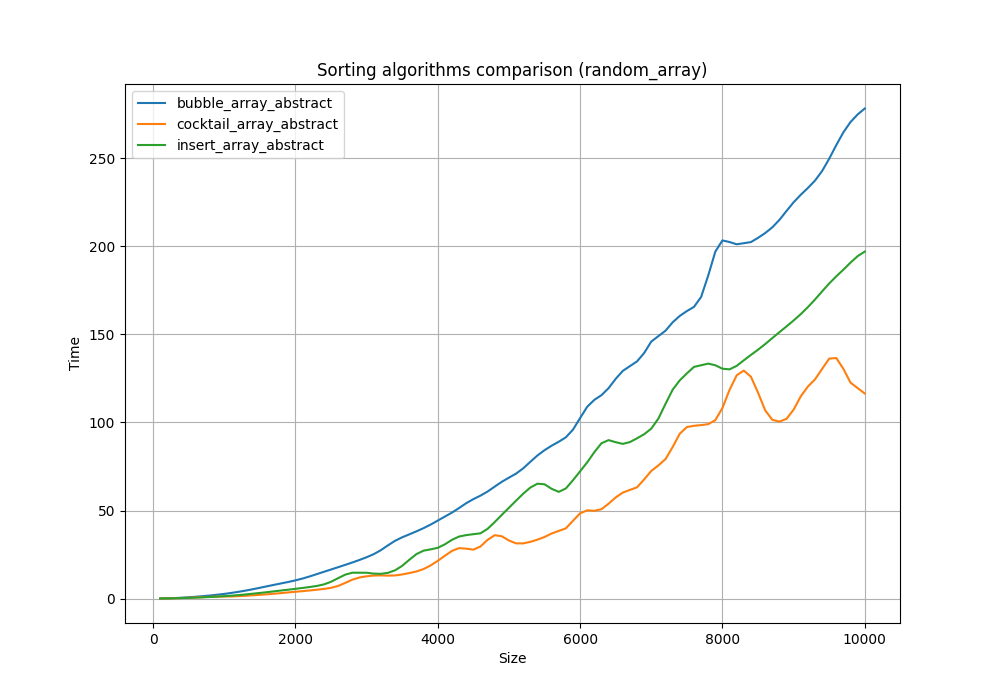
Оба контейнера – наследники класса Sequence<T>

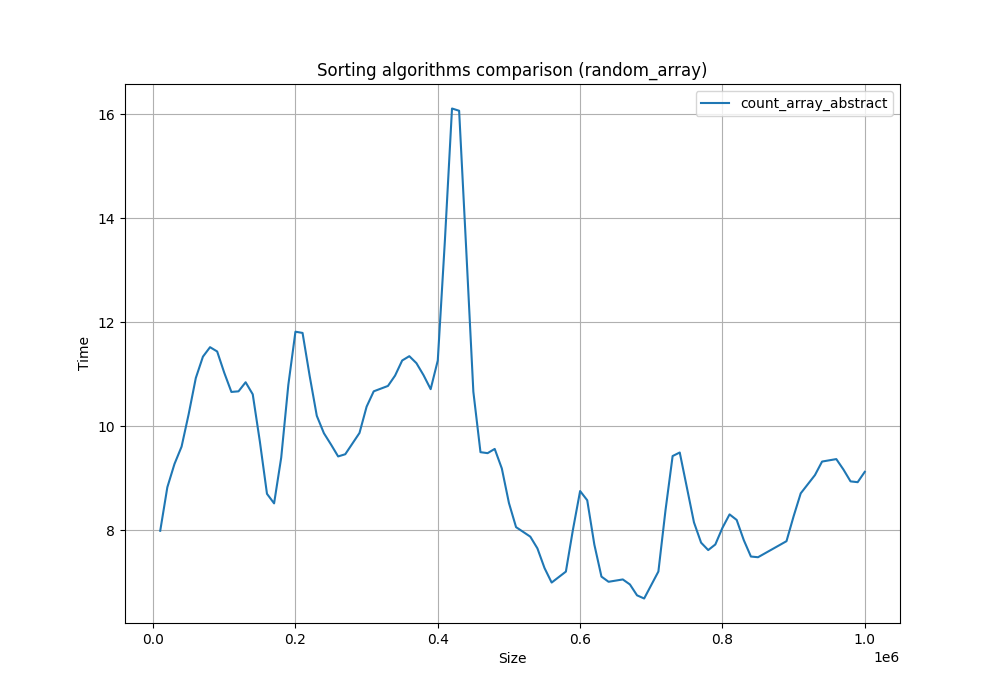
Реализован пользовательский консольный интерфейс, позволяющий выбрать юзеру, какую сортировку использовать, какую длину массива задать, выводить (или не выводить) по желанию список и массив до и после сортировки. Также добавлен выбор компаратора.

С помощью концептов и шаблонов реализована возможность сортировать любые контейнеры, у которых есть метод ”getLength()”, перегружены операторы присвоения и получения. Любые элементы с определенными операторами сравнения подходят для сортировки.

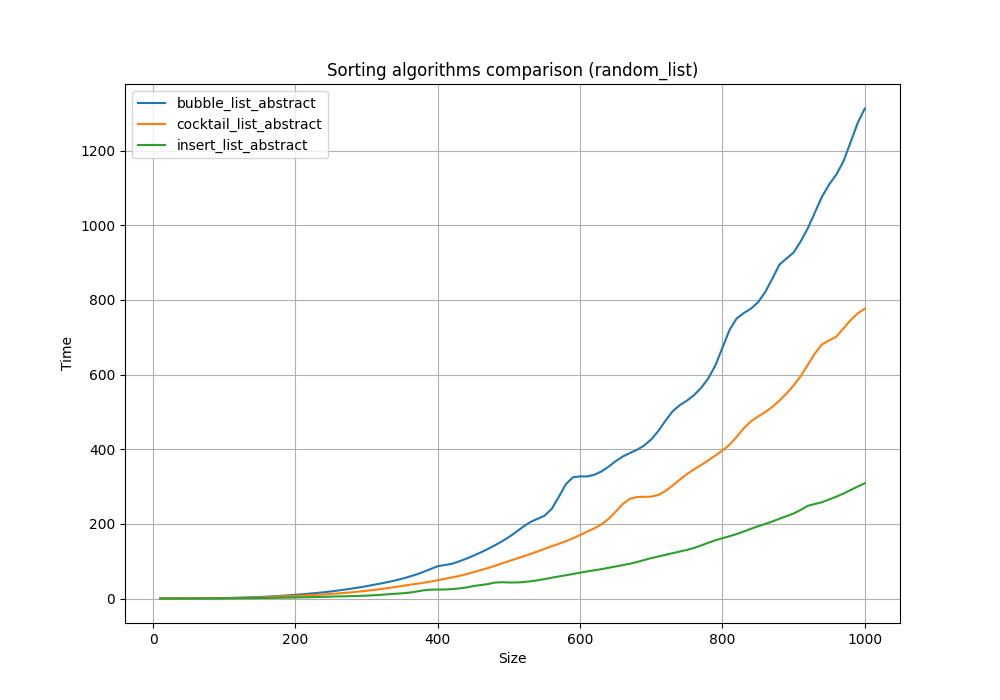
**Графики результатов сортировок для различных входных данных:**

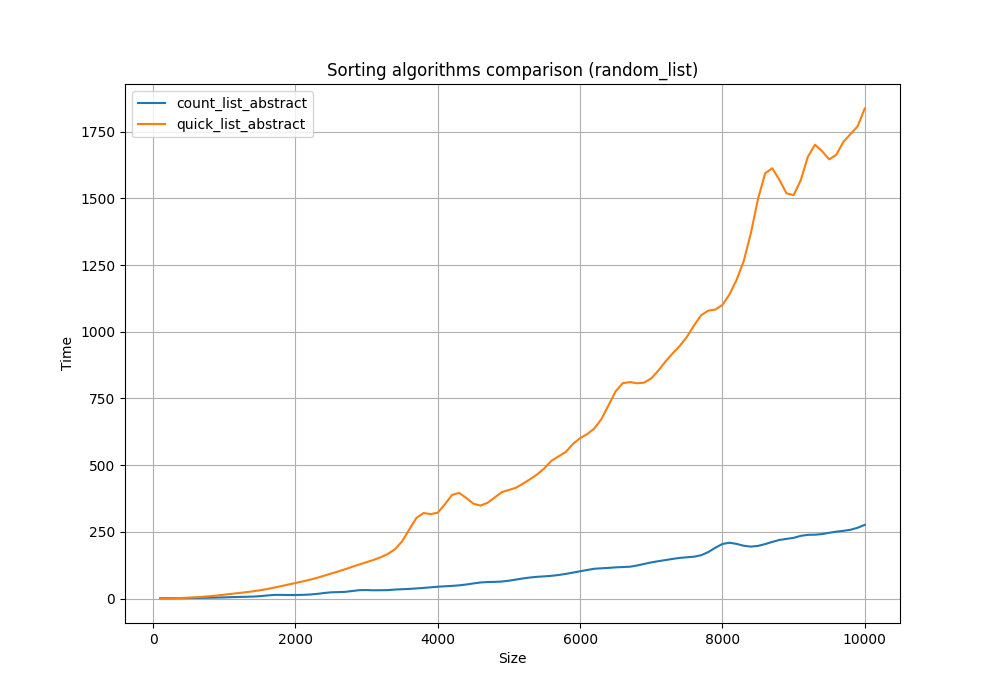
Все данные по результатам тестов выгружались в отдельный txt файл, затем в питоне обрабатывались, строились графики зависимости времени работы сортировки (в миллисекундах) от размера сортируемых данных.

**Графики для сортировок динамического массива на случайных данных**:

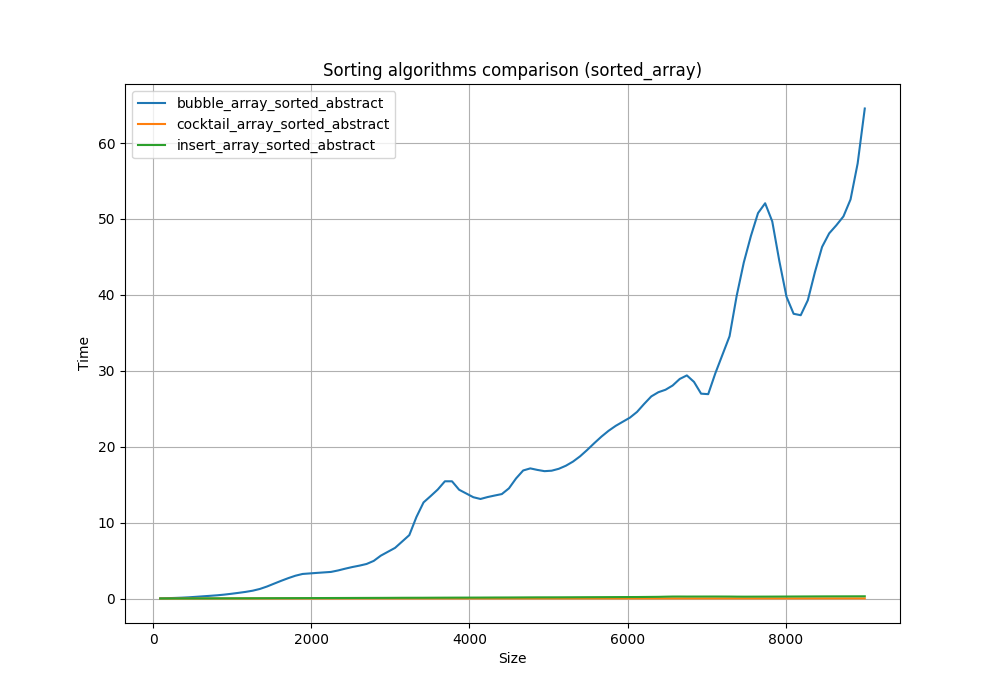


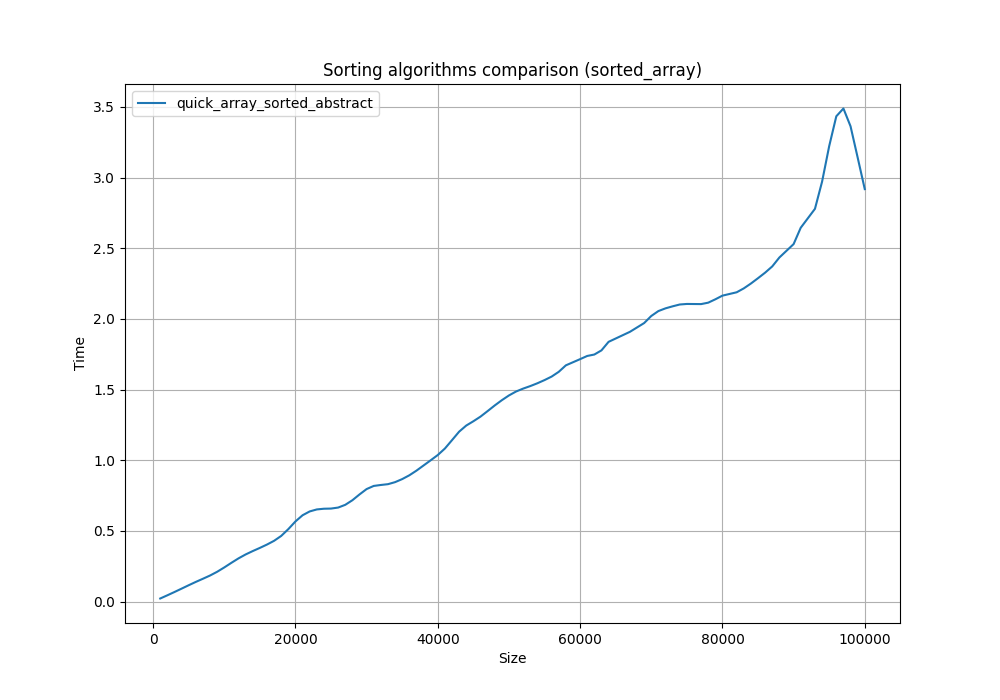
**Графики для сортировок связного списка на случайных данных:**



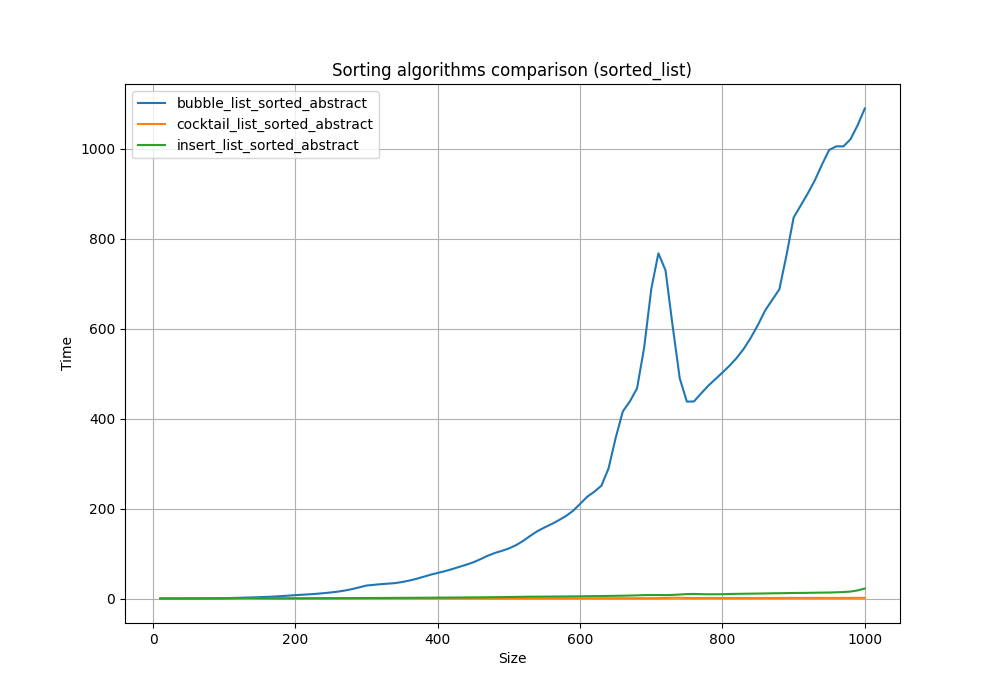


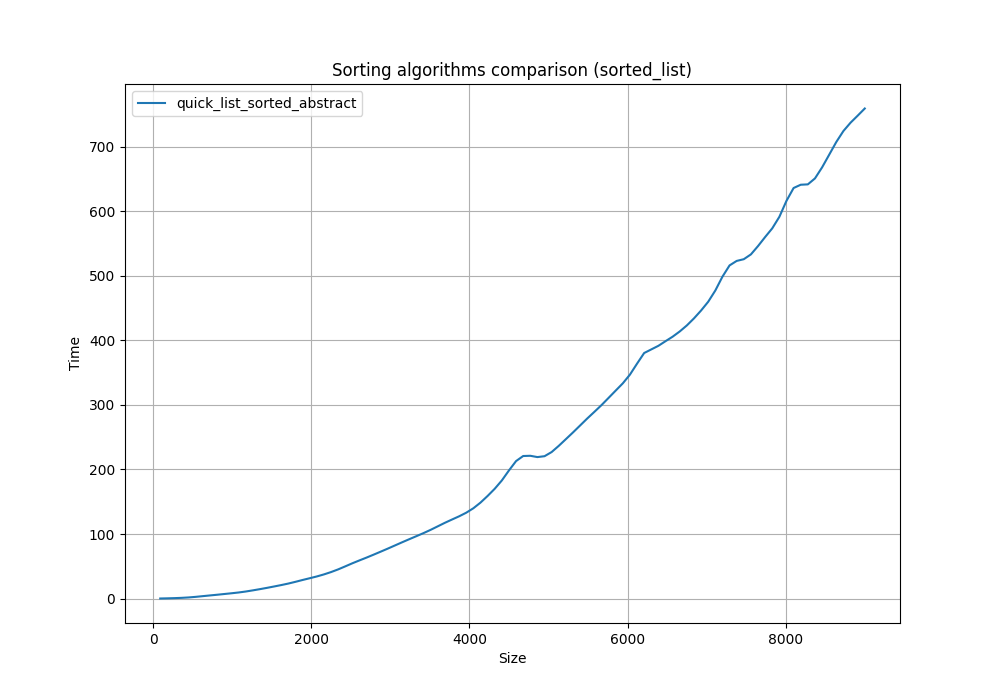
**Графики для сортировок динамического массива на отсортированных данных:**



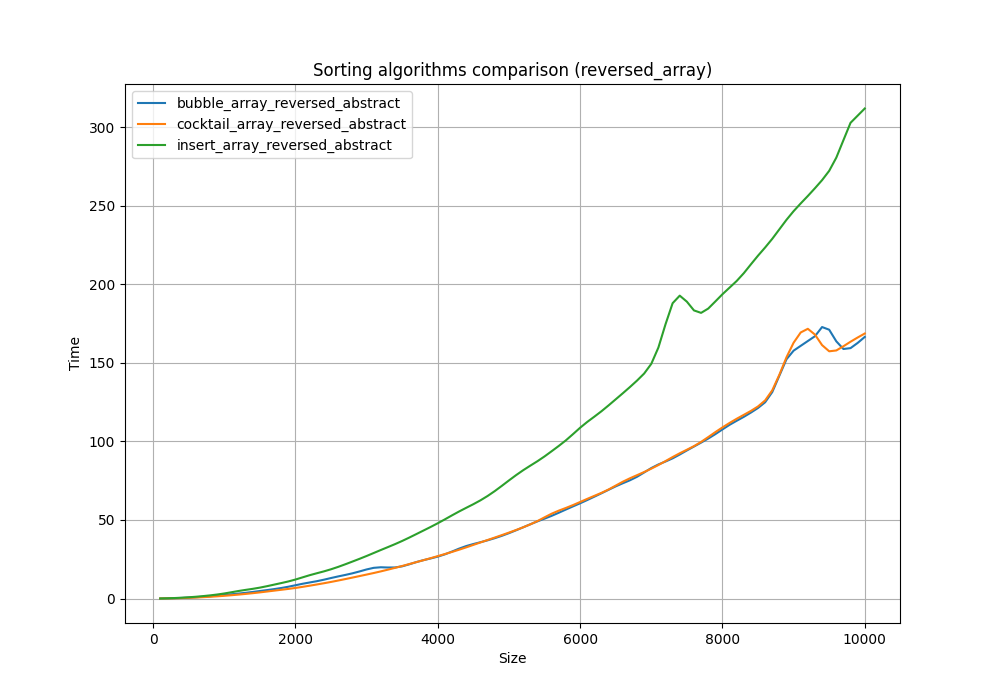


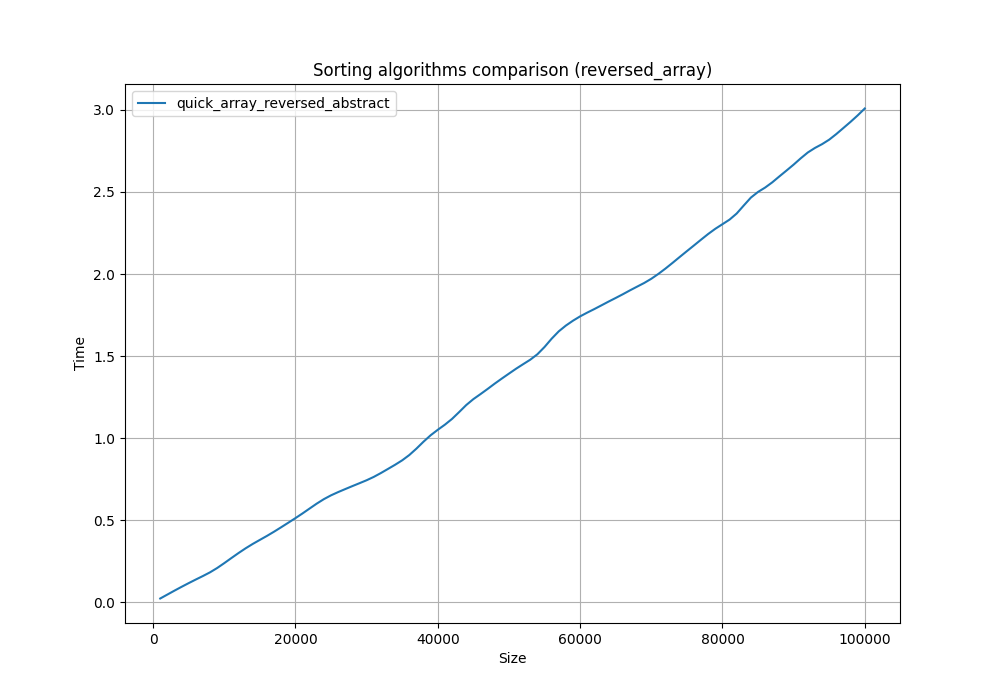
**Графики для сортировок связного списка на отсортированных данных:**



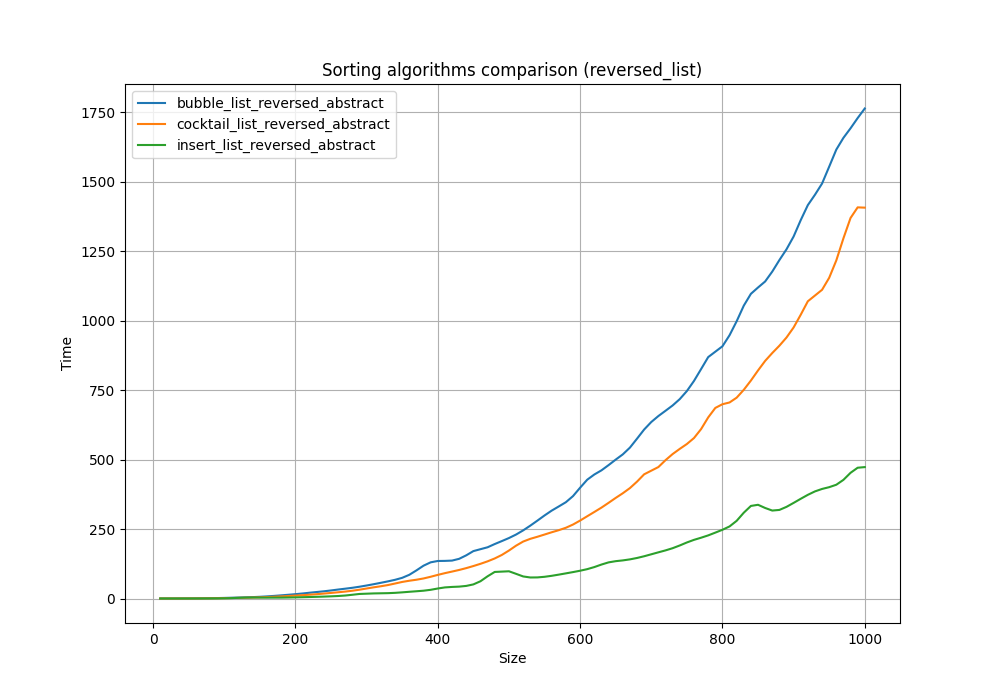


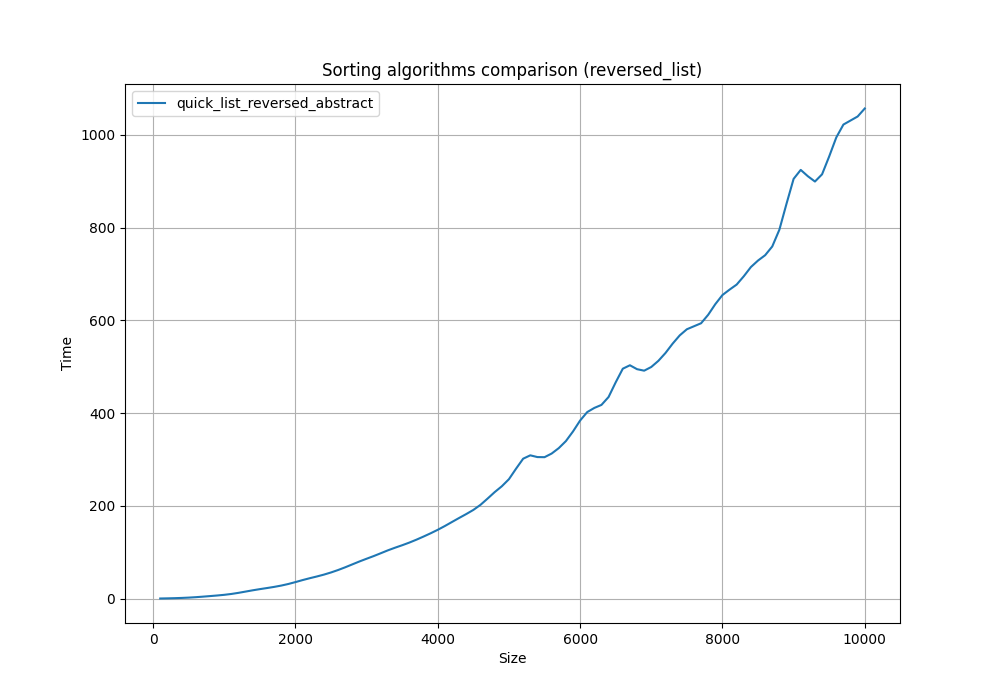
**Графики для сортировок динамического массива на сортированных развернутых данных:**





**Графики для сортировок связного списка на развернутых отсортированных данных:**





Для count sort не имело смысла проводить тест на различных типах входных данных, так как скорость работы этого алгоритма не зависит от расположения чисел в массиве.