Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский химико-технологический университет имени Д.И. Менделеева»

Кафедра информационных компьютерных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

Выполнил студент группы КС-33 (Вагенлейтнер Никита Сергеевич)

Ссылка на репозиторий: (https://github.com/OutumnRay/Algorithms\_2025.git)

Приняли: Пысин Максим Дмитриевич

Краснов Дмитрий Олегович

Лобанов Алексей Владимирович

Крашенинников Роман Сергеевич

Дата сдачи: (19.02.2025)

Оглавление

[Описание задачи. 1](#_Toc1992041850)

[Описание метода/модели. 1](#_Toc1472445925)

[Выполнение задачи. 1](#_Toc1220626840)

[Заключение. 1](#_Toc783145269)

# Описание задачи.

В лабораторной работе предлагается изучить способ анализа алгоритма связанный со временем. Рассмотреть для выбранного алгоритма сортировки наилучшие, наихудшее и среднее время и соотнести его с известным для алгоритма показателем эффективности O-большое.

# Описание метода/модели.

**Сортировка вставками**

Задача заключается в следующем: есть часть массива, которая уже отсортирована, и требуется вставить остальные элементы массива в отсортированную часть, сохранив при этом упорядоченность. Для этого на каждом шаге алгоритма мы выбираем один из элементов входных данных и вставляем его на нужную позицию в уже отсортированной части массива, до тех пор пока весь набор входных данных не будет отсортирован. Метод выбора очередного элемента из исходного массива произволен, однако обычно (и с целью получения устойчивого алгоритма сортировки), элементы вставляются по порядку их появления во входном массиве Так как в процессе работы алгоритма могут меняться местами только соседние элементы, каждый обмен уменьшает число инверсий на единицу. Следовательно, количество обменов равно количеству инверсий в исходном массиве вне зависимости от реализации сортировки. Максимальное количество инверсий содержится в массиве, элементы которого отсортированы по невозрастанию. Число инверсий в таком массиве n(n-1)/2

Алгоритм работает за О(1+k), где к число обменов элементов входного массива, равное числу инверсий. В среднем и в худшем случае за О(12). Минимальные оценки встречаются в случае уже упорядоченной исходной последовательности элементов, наихудшие когда они расположены в обратном порядке.

# Выполнение задачи.

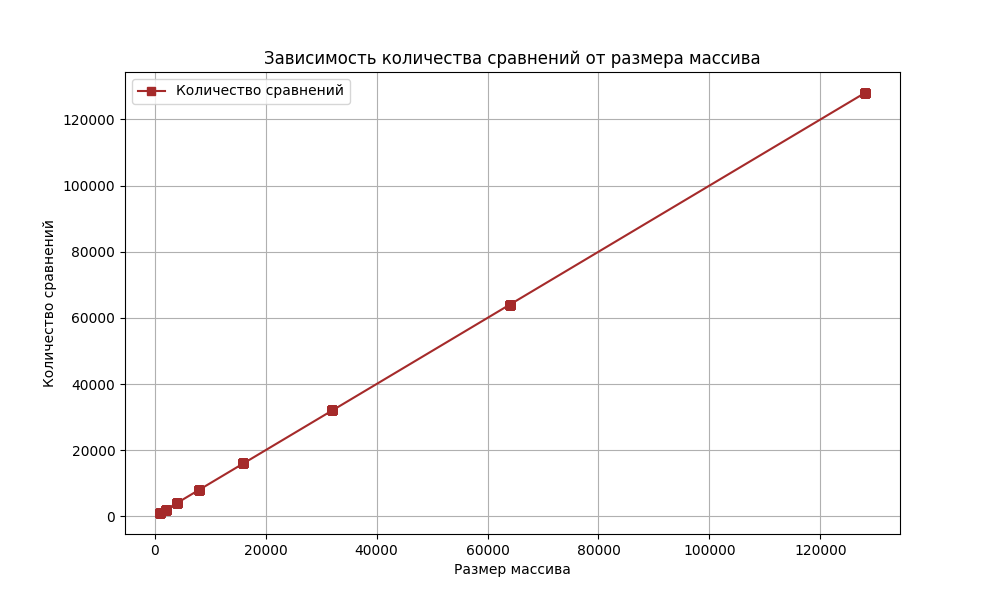
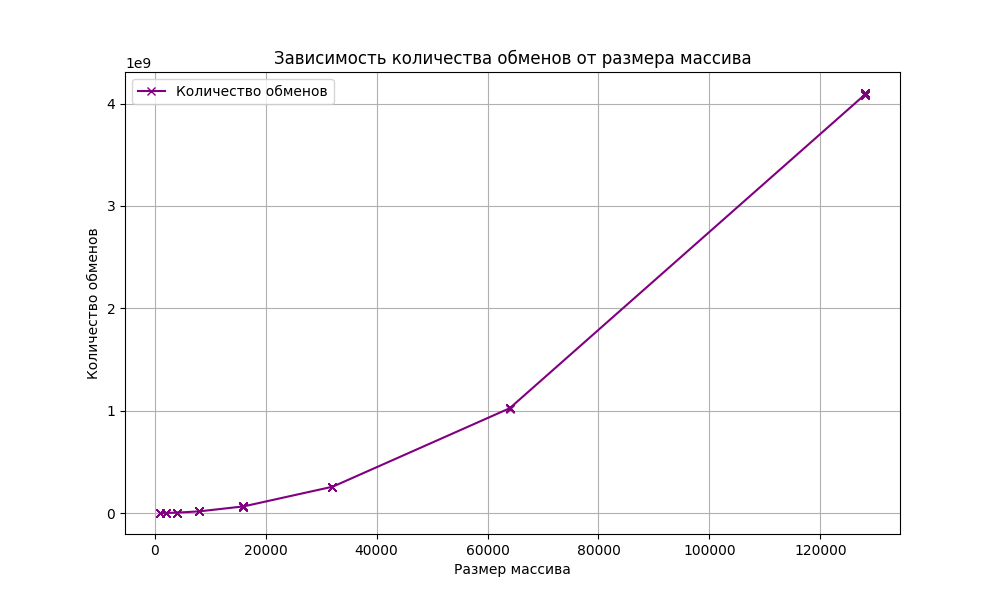
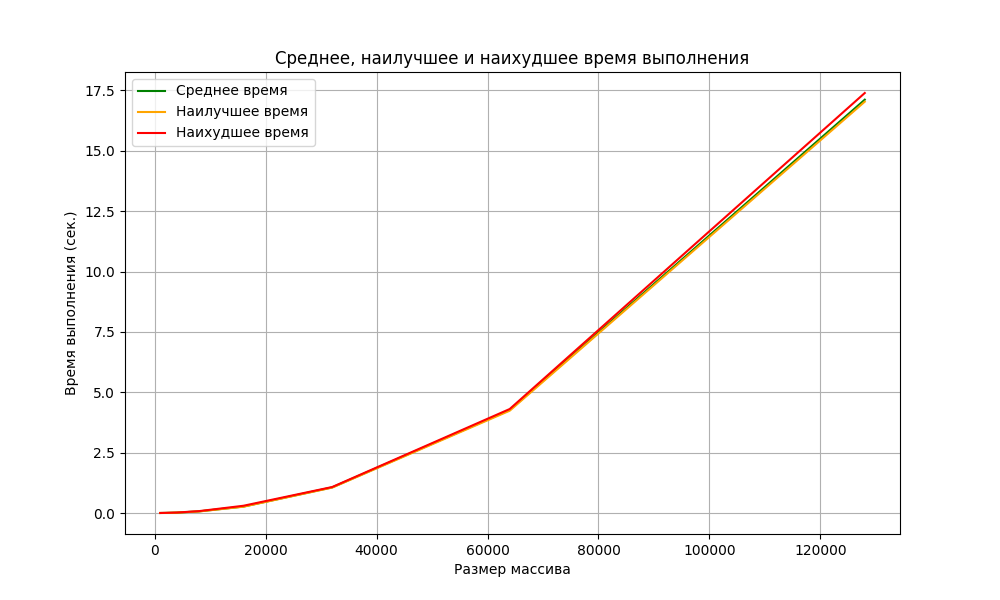
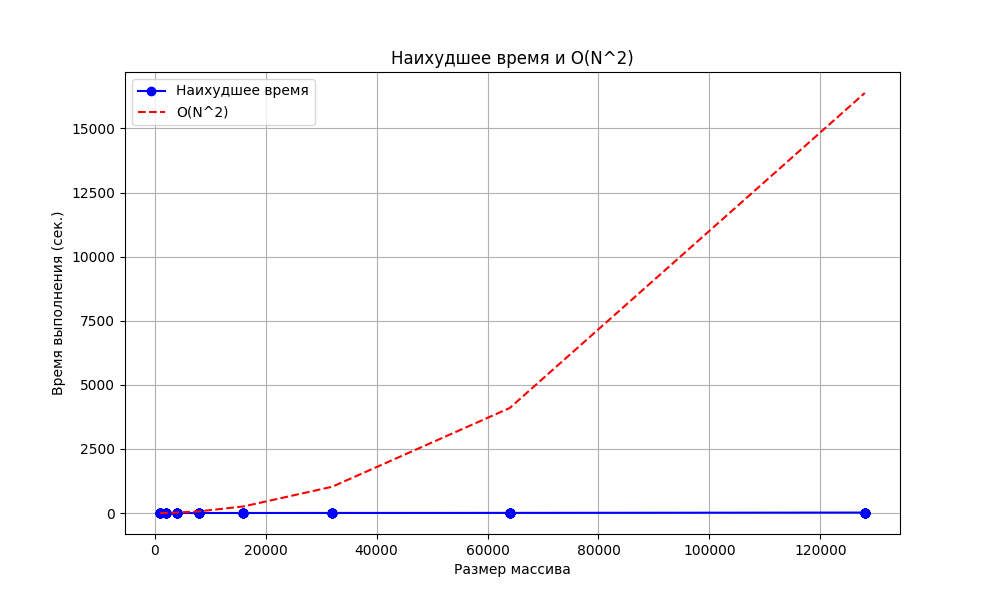
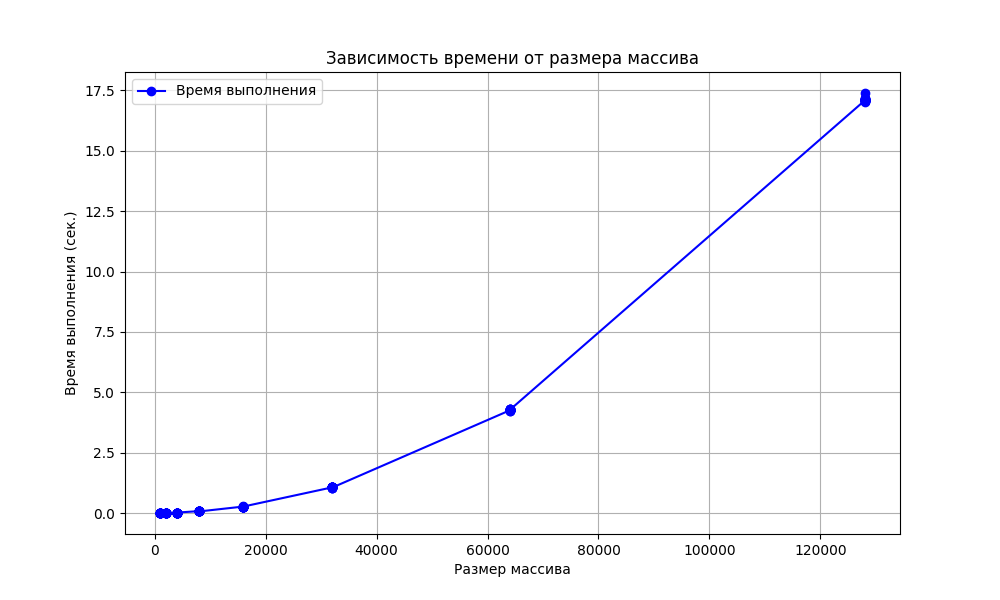
Для реализации и выполнения данного алгоритма было задействовано два языка C и Python. На C была реализована и протестирована сортировка вставками, а на Python был реализован аналитический блок полученных результатов работы алгоритма.

**Блок кода отвечающий за реализацию сортировки на C:**

Результаты работы программы выводится в data.txt файл в формате {размерность массива: время затраченное на сортировку: количество повторных заходов в массив: количество перестановок}. Пример приведён ниже для сортировки массива размерностью 1000 элементов.

**1000 Time taken: 0.001000 seconds, Entries: 999, Exchanges: 248145**

Полученный фал с данными после 20 итераций каждой из серий размерностей (1000, 2000, 4000, 8000, 16000, 32000, 64000, 128000) импортируется в Py скрипт для анализа и построения графиков зависимостей полученных результатов. Графики приведены ниже.



# Заключение.

По проделанной работе можно понять, что работа сортировки вставками сильно зависит от размерности массива. Чем больше размерность, тем больше времени на сортировку требуется (наихудший исход O(n^2)) и соответственно задействуется больше ресурсов рабочей машины. Так же стоит понимать, что стоит подбирать вид сортировки и ее организацию в программном коде строго под специфику исполняемой задачи, так как для больших размерностей данных и их последующей сортировки оптимальнее всего использовать подходящие под подобные задачи сортировки, например, можно усовершенствовать сортировку вставками бинарным поиском, что заметно ускорит работу программы, а так же уменьшит нагрузку на вычислительные машины.