Федеральное государственное автономное образовательное учреждение

высшего образования

«Национальный исследовательский Нижегородский государственный университет

им. Н.И. Лобачевского» (ННГУ)

Институт Информационных технологий, математики и механики

Отчёт по лабораторной работе:

«Сравнение алгоритмов сортировки»

**Выполнил:**

студент института ИТММ

гр. 382003-2

Баринов Артём Сергеевич

**Проверил:**

преподаватель каф. МОСТ   
института ИТММ

Пирова Анна Юрьевна

Нижний Новгород

2020 г.

Содержание

1. [Введение 3](#_Toc270962758)
2. [Постановка задачи 3](#_Toc270962759)
3. [Руководство пользователя 5](#_Toc270962760)
4. [Руководство программиста 7](#_Toc270962761)
   1. [Описание структур данных 7](#_Toc270962762)
   2. [Описание алгоритмов 7](#_Toc270962763)
   3. [Описание структуры программы 7](#_Toc270962764)
5. Вычислительные эксперименты 9
6. [Заключение 9](#_Toc270962765)
7. [Литература 10](#_Toc270962766)
8. [Приложения 11](#_Toc270962767)
9. [Приложение 1 11](#_Toc270962768)
10. [Приложение 2 11](#_Toc270962768)

# Введение

Используемые в настоящее время объёмы массивов данных значительно превосходят те, что использовались десятилетие назад или ещё раньше. С ростом объёмов данных всё актуальнее становится проблема оптимизации процессов, задействующих эти данные. В том числе алгоритмы сортировки.

Главным фактором выбора того или иного метода сортировки является его скорость выполнения. Данное напрямую зависит от принципа работы и реализации алгоритма. Из-за растущих требований к скорости выполнения и расширения области применения данных алгоритмов задача сравнительного анализа алгоритмов сортировки остаётся очень актуальной.

Целью моей работы является сопоставление по скорости выполнения алгоритмов сортировки данных в массивах.

В настоящее время существует множество алгоритмов сортировки, каждый из которых подходит под решение определенных задач. В этой работе будут рассмотрены наиболее популярные из них: методом вставок, метод Шелла, методом пузырька, методом слияния, быстрая сортировка.

Языком реализации был выбран язык Си. Это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью, и переносимостью. Указанные преимущества обеспечивают качественную разработку практически любого вида программного продукта. Использование Си в качестве инструментального языка позволит создать простые реализации алгоритмов.

# Постановка задачи

Необходимо реализовать программу, которая демонстрирует работу рассматриваемых алгоритмов сортировки. Максимальный размер массива не превышает 100\_000 элементов.

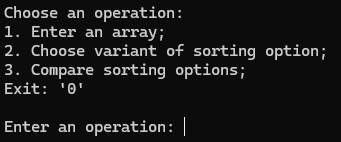
Программа должна поддерживать:

* Выбор способа ввода:
* вручную;
* автозаполнение псевдослучайными числами;
* Проверку корректности ввода данных.
* Выбор метода сортировки: для каждого метода вычисляется количество сравнений, перестановок элементов массива, общее время работы алгоритма в микросекундах.
* Возможность повторить сортировку ранее введенного массива при помощи другого метода сортировки.
* Возможность ввести новый массив.
* Сравнение работы алгоритмов сортировки по времени работы и количеству операций.

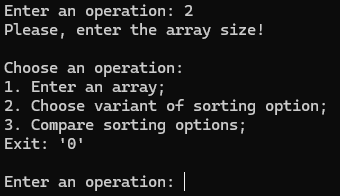
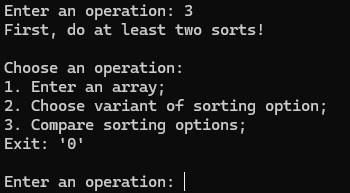
# Руководство пользователя

При запуске программы, высвечивается консоль с меню, где перечислены доступные опции.

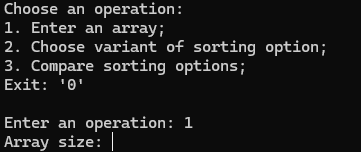
От пользователя ожидается ввод номера действия:



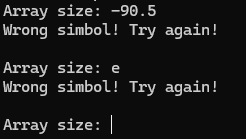
В случае, если пользователь пытается выбрать способ сортировки или сравнение сортировок до того, как был введен массив, то программа выведет предупреждение и предложит ввести номер операции повторно:

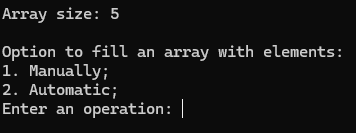
При выборе первого пункта меню – «1. Enter an array;» - программа предложит ввести размер массива:



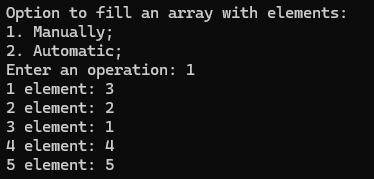
При вводе числа, не являющегося натуральным, или вводе нечисла программа выдаст ошибку и попросит повторить ввод:



После корректного ввода данных программа выведет меню, где предложит два способа заполнения массива: вручную («1. Manually;») и автоматически («2. Automatic;»):

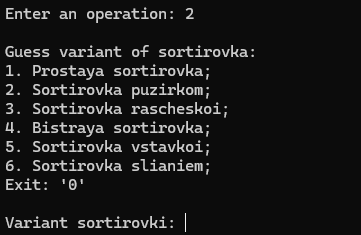


При выборе первого пункта (вручную), пользователь должен ввести каждый элемент массива:



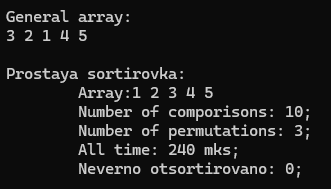
Автоматическое заполнение (пункт 2) осуществляется путем считывания элементов из файла, с записанными псевдослучайными числами.

Если пользователь выберет пункта «2. Choose variant of sorting option;», то будет выведено меню выбора метода сортировки:

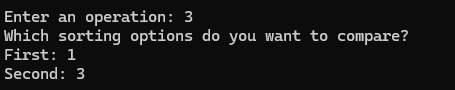


После этого будет создана копия массива, что даёт возможность сравнить несколько сортировок между собой, так как массив-копия перед сортировкой обновляется, чтобы соответствовать исходному.

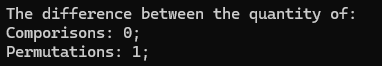
При выборе любого метода сортировки выведется исходный массив, а также следующая информация: отсортированный массив, количество сравнений, количество перестановок элементов и время выполнения в микросекундах:



Как только пользователь отсортировал массив хотя бы двумя разными методами, появится возможность сравнить сортировки между собой. Для этого пользователю необходимо выбрать в начальном меню третий пункт «3. Compare sorting options;». После чего указать первую сортировку и вторую, чтобы сравнить их данные (для примера взяты Простая сортировка и сортировка Шелла [Расчёска]). Для выхода из меню сортировок необходимо ввести “0” (ноль).



Далее будут выведены разница в количестве сравнений и перестановок:



При вводе “0” (нуля) программа завершит выполнение.

# Руководство программиста

***4.1 Описание алгоритмов***

1. **Алгоритм сортировки методом пузырька**

Данный алгоритм сравнивает между собой два соседних элемента, и в случае, когда их расположение не соответствует правильному (отсортированному) порядку, меняет их местами. Т.е. сначала сравниваются 1-й и 2-й элементы, потом 2-й и 3-й, затем 3-й и 4-й и т.д. Для полной сортировки приходится совершать несколько проходов по всему массиву и проверять одни и те же элементы.

Привести среднее значение по сортировкам в микросекундах