**Московский государственный технический**

**университет им. Н. Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Основы программирования»

Отчет по лабораторной работе № 5

«Сортировка одномерного числового массива»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-12Б |  | преподаватель каф. ИУ5 |
| Поляков Леонид |  | Правдина А. Н. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

**Постановка задачи**

Отсортировать числовой массив методами:

* метод сортировки выбором;
* методом пузырькового всплытия.

По окончании сортировки вывести:

* отсортированный массив;
* количество сделанных сравнений;
* перестановок элементов.

Сравнить быстродействие алгоритмов, которое определяется числом сравнений и перестановок, для исходного неотсортированного массива и для исходного массива, отсортированного в прямом и обратном порядке.

Исследовать зависимость быстродействия от размера массива. Возможность изменения длины массива реализуйте с помощью динамического массива, а для его инициализации используйте датчик случайных чисел. Результаты исследования выведите в виде таблицы (без рамок).

В начале выводится результат работы для случайно сгенерированного динамического массива. Количество элементов в массиве: 5. Сортировка осуществляется по возрастанию и по убыванию. Сравнение быстродействия алгоритмов:

* сначала сортировка применяется к неотсортированному массиву (<название метода> (n));
* затем сортировка применяется к отсортированному массиву (<название метода> (o)).

Затем генерируются массивы с количеством элементов 5, 50, 500. Сгенерированные массивы не выводить в консоль. Методы необходимо сравнивать на одинаковых массивах. Осуществлять сортировку по возрастанию на неотсортированном массиве.

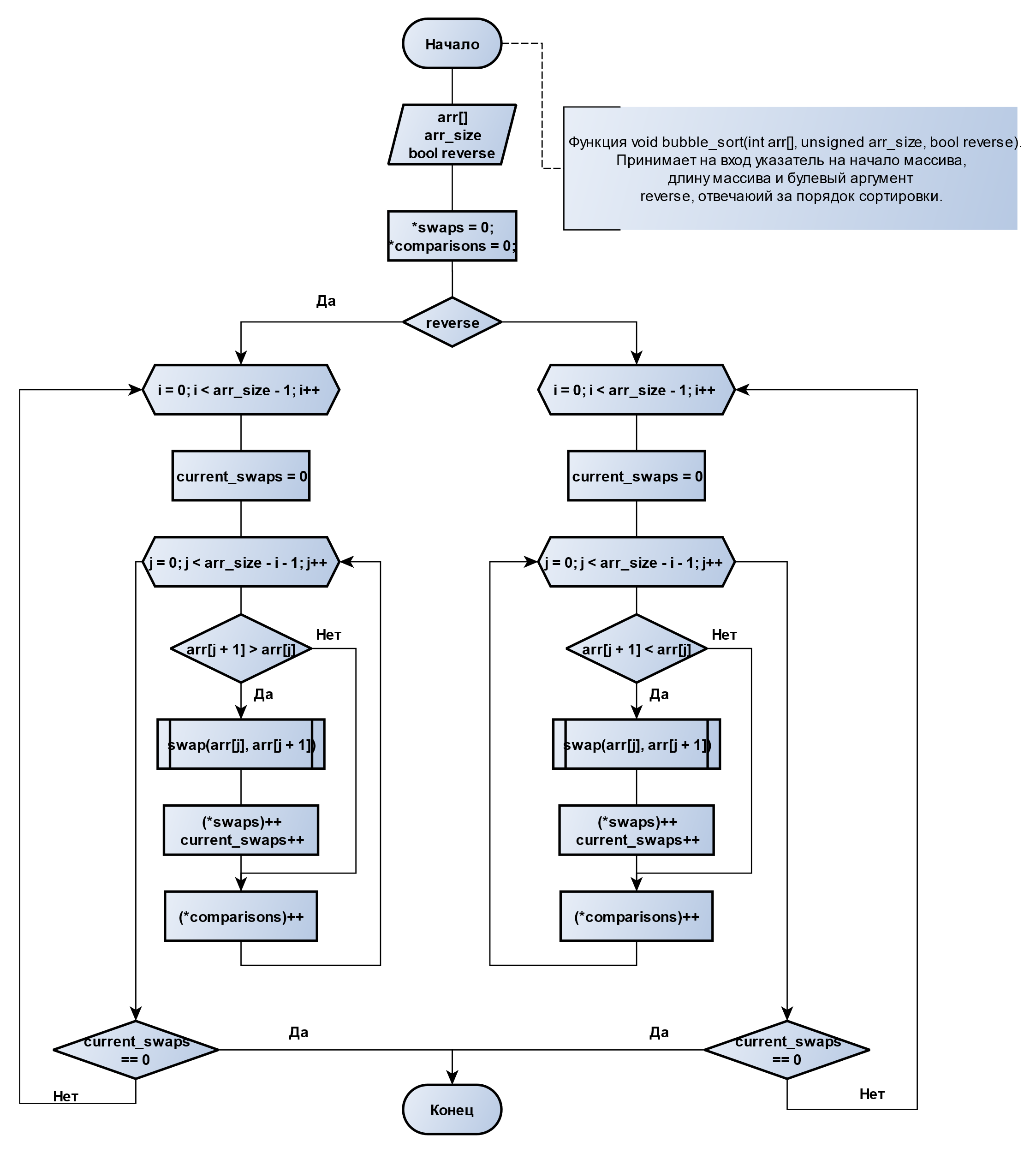
**Разработка алгоритма**

В данной лабораторной работе сортировка массива методом пузырька и сортировка массива выбора реализованы с помощью соответствующих функций, принимающих на вход указатель на динамический массив, длину этого массива и булевое значение, отвечающее за порядок сортировки (прямой или обратный). Для сортировки массива используются такие функции как void bubblesort(int [], unsigned, bool);

void selection\_sort(int[], unsigned, bool); они сортируют массив методами пузырька и выбора соответственно. Для заполнения массива случайными числами используется функция void fill\_array(int\*, unsigned, int, int, bool);

она принимает на вход указатель на динамический массив, размер этого массива, верхнюю и нижнюю границы числового промежутка, из которого нужно брать числа, а также булевое значение, отвечающее за то, будет ли функция брать отрицательные числа для заполнения. Копирование массивов производится с помощью функции void copy\_array(int[], int[], unsigned); она принимает на вход указатель на массив, который надо скопировать, указатель на массив, куда требуется скопировать элементы, и размер этих массивов. Вывод результатов на экран в виде таблицы производится с помощью функций void print\_result(int [], int [], unsigned, bool); void print\_result\_bubble\_sort(int [], int [], int []); void print\_result\_selection\_sort(int [], int [], int []); Первая из них сравнивает методы сортировки пузырьком и выбором на массивах размером 5 элементов, в отсортированном и в неотсортированном состоянии, в прямом и в обратном порядке. Другие две сортируют массивы размером 5, 50, 500 элементов (каждая своим методом), после чего выводят результат на экран в виде таблицы, где показано число сравнений и перестановок для каждого масссива.

**Функция сортировки пузырьком** void bubblesort(int [], unsigned, bool);



В данной функции указатели на переменные swaps и comparisons отвечают за подсчёт общего числа перестановок и сравнений для сортировки конкретного массива. При каждом новом вызове функции сортировки значения в них обнуляются. Параметры функции – указатель на динамический массив(arr), размер этого массива(arr\_size) и булевая переменная reverse, отвечающая за порядок сортировки (значение true, если сортировка идёт по убыванию, значение false – если по возрастанию). Переменная current\_swaps считает количество перестановок на конкретной итерации цикла, и если на момент очередного прохождения по массиву оказалось, что current\_swaps == 0, то это значит, что ни один элемент не поменял своей позиции, что говорит о том, что массив уже отсортирован, и функция сортировки завершает своё выполнение.

**Функция сортировки выбором** void selection\_sort(int[], unsigned, bool);

В данной функции указатели на переменные swaps и comparisons отвечают за подсчёт общего числа перестановок и сравнений для сортировки конкретного массива. При каждом новом вызове функции сортировки значения в них обнуляются. Параметры функции – указатель на динамический массив(arr), размер этого массива(arr\_size) и булевая переменная reverse, отвечающая за порядок сортировки (значение true, если сортировка идёт по убыванию, значение false – если по возрастанию). Переменные max\_index и min\_index хранят в себе на каждой итерации индексы максимального и минимального ещё неотсортированного элемента в массиве соответственно.

**Функция копирования массива**

void copy\_array(int arr\_to\_be\_copied[], int arr\_where\_to\_copy[], unsigned arr\_size)

****

**Текст программы**

**constants.h**

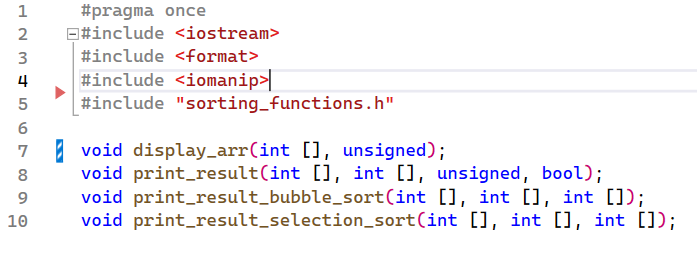
**Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание**

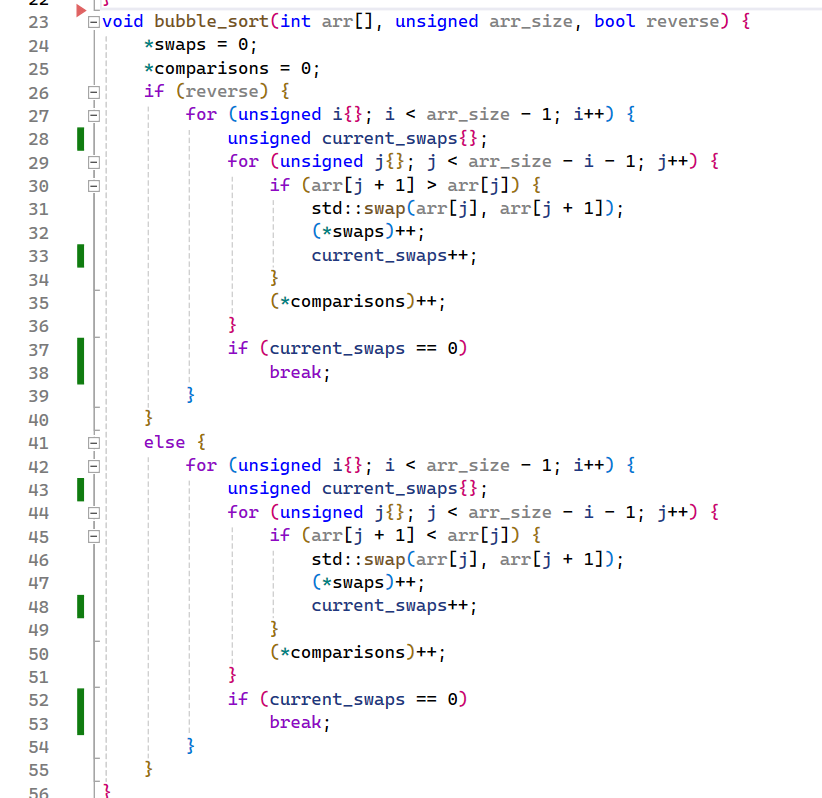
**sorting\_functions.h**

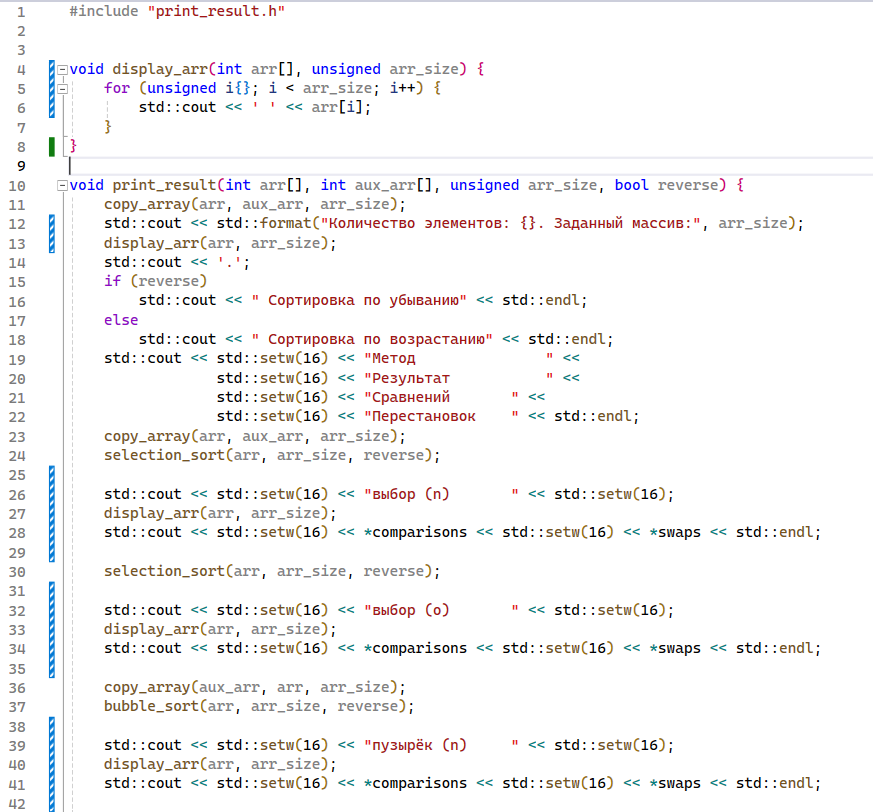
*Изображение выглядит как текст, Шрифт, число, снимок экрана

Автоматически созданное описание*

**print\_result.h**

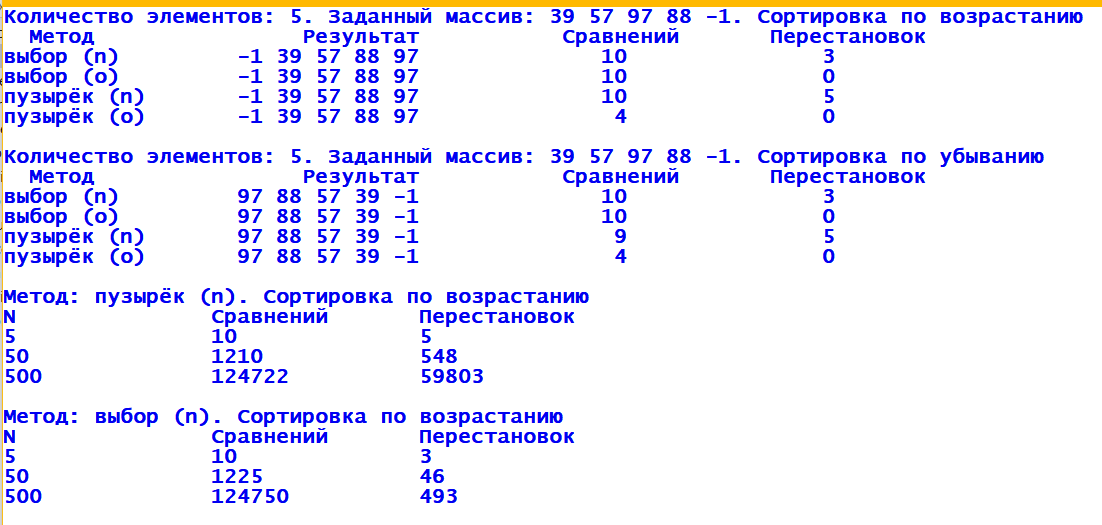
**sorting\_functions.cppИзображение выглядит как текст, Шрифт, программное обеспечение, число

Автоматически созданное описание**

**print\_result.cpp**

**main.cppИзображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание**

**Анализ Результатов**

**Результат работы программы соответствует ожиданиям и полностью удовлетворяет требованиям ТЗ**

**Вывод**

На небольших массивах (в данном случае размером 5 элементов) оба метода сортировки справляются одинаково хорошо. Но когда размер массива увеличивается на несколько порядков, то чётко начинает прослеживаться преимущество сортировки методом выбора перед сортировкой пузырьком, так как последний метод совершает примерно в 121 раз больше перестановок при сортировке массива размером 500 элементов.

**Использованные источники**

* **Веб-сайт кафедры ИУ5, курс «основы программирования»**

[**https://cpp1.wiki.iu5edu.ru**](https://cpp1.wiki.iu5edu.ru)

* **Веб-сайт кафедры ИУ5 «автоматизированные системы обработки информации и управления»**

**https://e-learning.bmstu.ru/iu5/mod/folder/view.php?id=277**