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. 3823Б1ПМ2

Кротов Роман Романович

**Проверил**:

ст. преп. кафедры ВВиСП    
института ИТММ

Пирова А. Ю.

Нижний Новгород

2024 г.

**Содержание**

[Введение 3](#_Toc161747297)

[1 Постановка задачи 4](#_Toc161747298)

[2 Руководство пользователя 5](#_Toc161747299)

[3 Руководство программиста. 7](#_Toc161747300)

[3.1 Описание алгоритмов. 7](#_Toc161747301)

[3.2 Описание модульной структуры программы 9](#_Toc161747302)

[3.3 Описание структуры программы 9](#_Toc161747303)

[4 Вычислительные эксперименты 10](#_Toc161747304)

[Заключение 11](#_Toc161747305)

[Литература 12](#_Toc161747306)

# Введение

Сортировка представляет собой упорядочивание элементов набора данных в определенной последовательности, либо по возрастанию, либо по убыванию ключей. Ее основная цель - обеспечить более эффективную обработку данных в случае больших объемов информации и представить данные в удобной для анализа форме. Время сортировки играет важную роль, поскольку оно определяет скорость работы алгоритма. Поэтому с течением времени появилось множество различных методов сортировки, обеспечивающих оптимизацию процесса.

В рамках моего исследования, я рассмотрю шесть различных видов сортировок и проанализирую их производительность. Для реализации выбран язык программирования C.

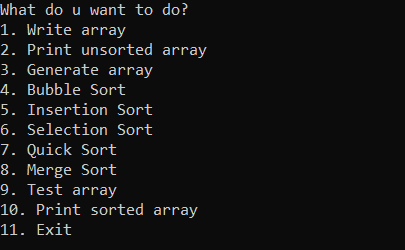
# Постановка задачи

Необходимо реализовать программу, демонстрирующую работу различных алгоритмов сортировки на массиве целых чисел. Для этого программа должна выполнять следующие функции:

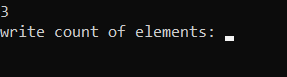
* Cлучайную генерацию элементов массива с заданным пользователем количеством элементов в массиве.
* Возможность ввода массива.
* Выбор метода сортировки.
* Для каждой сортировки необходимо выводить время ее работы.
* Возможность повторения работы на том же массиве с применением другого алгоритма сортировки.
* Возможность проверить правильность отсортированного массива.
* Возможность вывести отсортированный массив.

# Руководство пользователя

После запуска программы появляется меню с перечнем опций.



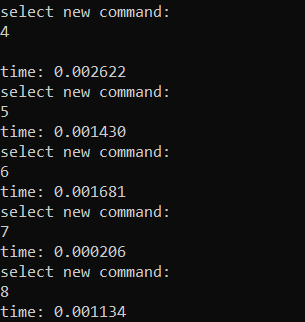
Для того что бы сгенерировать массив, пользователю необходимо выбрать «3» и указать количество элементов в массиве.

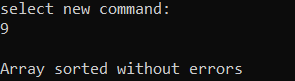


После чего программа спросит, что мы хотим сделать далее.



Если пользователь выберет «4», то сгенерированный массив отсортируется методом пузырька и покажет время выполнения в секундах. И так на каждый метод сортировки.



Для проверки корректности сортировки необходимо нажать клавишу "9". Если сгенерированный массив отсортирован верно, программа выведет сообщение "Array sorted without errors". В противном случае будет выведено сообщение "Array sorted with errors!", что позволит проверить правильность каждого метода сортировки после его применения.

При выборе пользователем опции "11", программа завершит свою работу.

# Руководство программиста.

## Описание алгоритмов.

Алгоритм сортировки методом пузырька.

1. Начинаем сравнивать первый и второй элементы списка. Если второй элемент меньше первого, меняем их местами.

2. Переходим ко второму и третьему элементам и снова сравниваем их. Если третий элемент меньше второго, меняем их местами.

3. Продолжаем этот процесс до конца списка, сравнивая и обменивая соседние элементы.

4. После первого прохода наибольший элемент окажется на последнем месте. Затем повторяем все шаги, исключая последний элемент.

5. Продолжаем проходы по списку до тех пор, пока все элементы не будут упорядочены по возрастанию.

Алгоритм сортировки методом вставок.

1. С начала второго элемента в массиве выбирается элемент и сравнивается со всеми предшествующими элементами.

2. В случае, если текущий элемент оказывается меньше любого из предыдущих, он перемещается влево, пока не обретет свое истинное местоположение.

3. Этот алгоритм повторяется для каждого элемента в массиве до тех пор, пока не будет достигнута полная упорядоченность всего массива.

Этот метод считается эффективным для небольших массивов и для массивов, которые уже почти упорядочены, однако он проявляет себя менее эффективным в случае больших списков из-за его квадратичной временной сложности.

Алгоритм сортировки выбором.

1. На каждом этапе алгоритма находится минимальный элемент в оставшейся неотсортированной части массива, который затем обменивается местами с первым неотсортированным элементом. Этот процесс повторяется до тех пор, пока весь массив не будет упорядочен.

2. Мы проходим по всем элементам массива.

3. Находим минимальный элемент в оставшейся части массива.

4. Обмениваем найденный минимальный элемент с первым неотсортированным элементом.

5. Увеличиваем индекс неотсортированной части массива на 1.

6. Повторяем шаги 3-5 до завершения сортировки всего массива.

Этот алгоритм отличается простотой в реализации и эффективен при сортировке небольших массивов. Однако на больших массивах он не является оптимальным из-за значительного количества сравнений, которые требуются для нахождения минимального элемента на каждом шаге.

Алгоритм быстрой сортировки.

1. Выбирается опорный элемент из массива. Обычно это средний элемент.

2. Все элементы меньше опорного помещаются слева от него, а все элементы больше - справа.

3. Рекурсивно применяется быстрая сортировка к левой и правой частям массива.

4. Процесс продолжается, пока каждый подмассив не будет отсортирован.

5. В итоге все подмассивы объединяются в один упорядоченный массив.

Алгоритм быстрой сортировки обладает временной сложностью O(n log n) в среднем случае, что делает его одним из самых быстрых методов сортировки.

Алгоритм сортировки слиянием.

Алгоритм сортировки слиянием представляет собой эффективный метод сортировки, основанный на принципе "разделяй и властвуй".

1. Разделение: Исходный массив разделяется на две равные (или практически равные) части.

2. Рекурсивное разделение: Каждая из полученных частей сортируется отдельно путем рекурсивного применения алгоритма сортировки слиянием.

3. Слияние: Отсортированные подмассивы сливаются в один упорядоченный массив.

Основное преимущество сортировки слиянием заключается в том, что она гарантированно работает за время O(n log n) в худшем, лучшем и среднем случаях, что делает ее одним из

## Описание модульной структуры программы

void BubbleSort (int mass[], int N)

void InsertionSort (int mass[], int N)

void SelectionSort (int mass[], int N)

void QuickSort (int mass[], int N)

void MergeSort (int mass[], int N)

Подпрограмма сортировки пузырьком, вставкой, выбором, быстрая, слиянием . На вход принимается массив, длина массива. Результатом подпрограммы является отсортированный массив, но не выведенный на экран.

void Generate(int mass[], int N)

Подпрограмма, генерирующая массив длины N, которую указал пользователь. На вход принимается незаполненный массив, длина массива.

void Test(int mass[], int N)

Подпрограмма проверки правильности массива после сортировки. На вход принимается массив, длина массива. В результате подпрограмма выводит: "Сортировка выполнена правильно» или "Сортировка выполнена неправильно».

void Print (int mass[], int N)

подпрограмма, которая выводит массив. На вход принимается массив, длина массива.

## Описание структуры программы

Программа разделена на 3 файла

В одном файле находятся названия заголовков, во втором все реализации, третий файл содержит главную программу.

# Вычислительные эксперименты

Для того чтобы сравнить методы сортировки по времени работы, необходимо отсортировать один и тот же массив разными методами. Для вычислительных экспериментов было решено выбрать массивы размеров 5000, 10000, 50000, для наглядности полученных результатов.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Размер массива | Сортировка пузырьком | Сортировка вставками | Сортировка выбором | Быстрая сортировка | Сортировка слиянием |
| 50000 | 4.850299 | 1.821013 | 3.001263 | 0.005266 | 0.034413 |
| 100000 | 19.961352 | 12.865060 | 11.996770 | 0.009268 | 0.066780 |

Видно, что наилучший результат - у сортировки методом слиянием. Сортировка слиянием требует дополнительного пространства для хранения временных массивов в процессе слияния. Это может быть проблематично при работе с очень большими массивами или при ограниченности доступной оперативной памяти.

# Заключение

В результате выполнения данной работы была написана программа, анализирующая сортировку методом пузырька, вставки, выбором, быстрой, слиянием. Применяя к одному и тому же массиву разные виды сортировок, можно увидеть и сравнить скорость работы каждого метода. Наихудший результат - у сортировки методом пузырька, наилучший - у сортировки методом слияния.

# Литература

1. Керниган Б.В, Ричи Д.М. Си для профессионалов. М.: Энергия, 1996.– 213 с.
2. Либерман М. Алгоритмы сортировки массивов. М.: Наука, 1997. – 43-81с.