**Государственное бюджетное образовательное учреждение**

**высшего образования Московской области**

**«Университет «Дубна»**

**(государственный университет «Дубна»)**

Кафедра проектирования электроники для установок «мегасайенс»

**Курсовая работа**

по дисциплине «Программирование на языках высокого уровня»

**Тема: «Визуализация работы алгоритма пузырьковой сортировки»**

Выполнил студент группы: 1142

Кошкин Андрей Дмитриевич

Проверил: ст. преп. Полуян С. В.

Дубна, 2021 г.

### **Оглавление**

Введение 3

Постановка задачи 4

Глава 1. Описание используемого алгоритма сортировки 6

Глава 2. Проектирование и реализация приложения 7

2.1 Проект 7

2.2 Реализация 8

2.3 Описание меню программы 11

Заключение 12

Список использованной литературы 13

Приложение А 14

### **Введение**

Настоящая работа посвящена реализации алгоритма пузырьковой сортировки с помощью языка Си.

Си – компилируемый статически типизированный язык программирования общего назначения, разработанный в 1969—1973 годах Деннисом Ритчи. Первоначально он был разработан для реализации операционной системы UNIX, но, впоследствии, был перенесён на множество других платформ. Язык программирования Си оказал существенное влияние на развитие индустрии программного обеспечения, а его синтаксис стал основой для таких языков программирования, как C++, C#, Java и так далее. Язык программирования Си отличается минимализмом. Код Си может быть скомпилирован без каких-либо изменений почти на любой модели компьютера. Зачастую язык программирования Си называют языком среднего уровня или даже низкого уровня, учитывая то, как близко он работает к реальным устройствам.

Причина использования языка Си в настоящей работе кроется в эффективности (он позволяет наилучшем образом использовать возможности устройства), быстроте (реализация программы происходит с более высокой скоростью чем у большинства программ, написанных на других языках высокого уровня), а также удобством (язык хорошо структурирован, чтобы поддерживать хороший стиль программирования и не связывать руки программиста большим количеством ограничений).

Визуализация алгоритма происходила с помощью библиотек **libpng.**

**libpng** — официальная эталонная библиотека для работы с растровой графикой в формате PNG. Библиотека является платформо-независимой и состоит из функций, написанных на языке Си. Для работы **libpng** необходима библиотека **zlib** (свободная кроссплатформенная библиотека для сжатия данных).

Курсовая работа организована следующим образом: глава первая — описание используемого алгоритма пузырьковой сортировки, излагается принцип его работы и приводится вычислительная сложность. Вторая глава посвящена деталям проектирования приложения: представлены примеры работы приложения и взаимодействия с пользователем. В заключении приводятся результаты выполненной работы.

### **Постановка задачи**

1. **Цель**

Целью данной курсовой работы является реализация алгоритма пузырьковой сортировки одномерного массива с последующей визуализацией процесса сортировки произвольной целочисленной последовательности.

1. **Исходные данные**

Алгоритм пузырьковой сортировки одномерного массива, справочная информация библиотеки libpng.

1. **Модель**

Визуализация работы алгоритма пузырьковой сортировки

1. **Результат**

Консольное приложение, реализованное на языке Си с использованием библиотеки libpng демонстрирующее этапы работы алгоритма пузырьковой сортировки с помощью серии PNG кадров. Сгенерированные кадры затем собираются в анимированное PNG изображение независимо от приложения. Минимальный пользовательский функционал предоставляет собой краткую инструкцию, разъясняющую каким образом использовать данное приложение. На рис.1 представлено меню разработанного приложения.

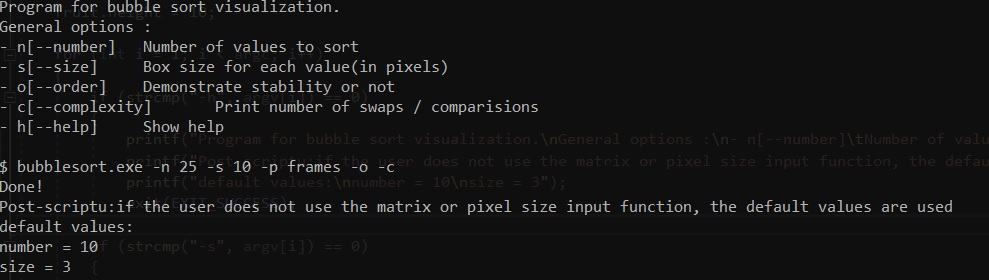


Рис.1 Пользовательский функционал

1. **Критерии оценивания**
2. Пользователь может задать размер изображения матрицы вместе с количеством сортируемых значений (в пикселях).
3. Пользователь может задать размер одного пикселя на изображении.
4. Пользователь может получить печать количества свопов/сравнений.
5. Пользователь может вызвать линию значений с одинаковой высотой и разными цветами (для проверки устойчивости).
6. Пользователь может вызвать функцию помощи.

### **Глава 1. Описание алгоритма пузырьковой сортировки**

**Сортировка пузырьком** (англ. *bubble sort*) —простейший для понимания и реализации алгоритм, но эффективен он лишь для небольших массивов. {\displaystyle (n^{2})} Алгоритм считается учебным и практически не применяется вне учебной литературы, вместо него на практике применяются более эффективные алгоритмы сортировки.

Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется перестановка элементов. Проходы по массиву повторяются {\displaystyle N-1}N - 1 раз или до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован. При каждом проходе алгоритма по внутреннему циклу, очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива. После первого завершения внутреннего цикла максимальный элемент массива всегда находится на N{\displaystyle N}NNf-ой позиции. При втором проходе, следующий по значению максимальный элемент находится на N - 1 {\displaystyle N-1}месте, и так далее. Таким образом, на каждом следующем проходе число обрабатываемых элементов уменьшается на 1 и нет необходимости «обходить» весь массив от начала до конца каждый раз. Реализация сортировки в приложении представлена на Рис.2

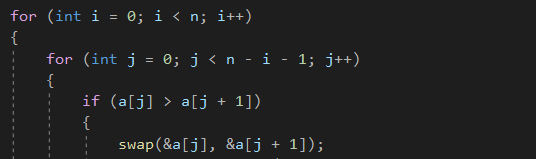


Рис.2 реализация пузырьковой сортировки

В данной сортировке число проходов по массиву в любом случае будет равняться N(N-1)/2. Так как в алгоритме могут меняться только соседние элементы, то в наихудшем случае число обменов так же – N(N-1)/2. В наилучшем случае число обменов – 0. Получается, что алгоритмическая сложность сортировки – O(n2).

### **Глава 2. Проектирование и реализация приложения**

### **2.1 Проект**

Описанный в первой главе алгоритм реализован на языке программирования

высокого уровня Cи.

В курсовой работе использовалась раннее упомянутая библиотека libpng. В проекте

использовались заголовочные файлы стандартной библиотеки языка Си, такие как:

1. stdlib.h (содержит в себе функции, занимающиеся выделением памяти, контролем

процесса выполнения программы, преобразованием типов и другие);

1. stdio.h (содержит определения макросов, константы и объявления функций и типов,

используемых для различных операций стандартного ввода и вывода);

1. string.h (содержит функции для работы с нуль-терминированными и различными

функциями работы с памятью);

1. stdint.h (содержит в себе функции для создания целочисленных типов с заданными размерами и макросы для установки размеров этих типов)

Также использовался заголовочный файл внешней библиотеки языка Си png.h.

### **2.2 Реализация**

После проделанной работы получается файл out.apng, который собирает множество кадров в один apng.

Анимация создавалась с помощью приложения APNG Assembler, которое создает анимированный APNG файл из последовательности PNG/TGA изображений.

Программа представляет из себя консольное приложение, которое мы запускаем через приложение «Командная строка». При указании исполняемого файла, мы также можем задать функции: размер матрицы (-n), размер одного пикселя (-s), поверка на устойчивость (-o), количество свопов и помощь (-с).Порядок ввода функций может быть произвольным. Если пользователь не задает функции размеров матрицы или пикселя, то используются значения по умолчанию. Для (-n) это 10, для (-s) – 3. Если пользователь не задает функции (-o) или (-с), то они не активируются. В случае, если пользователь задает функцию (-h), то происходит вывод справки для помощи по эксплуатации приложения, и на этом программа завершает свою работу. В том случае, если пользователь задал некорректные данные для функций (-n) и (-s), то программа сообщает об этой ошибке и завершает свою работу (Рис.3). В случае успешного завершения программы программа сообщает об этом выводом на консоль сообщения («Done»).

При задании каких-либо значений, никак не связанных с основными функциями, то программа их просто игнорирует (Рис.4).

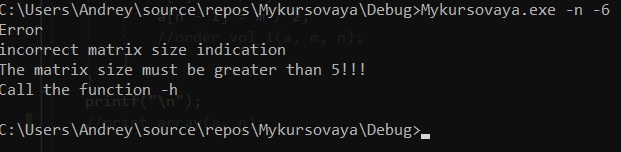


Рис.3 Работа программы в случае ввода некорректных данных

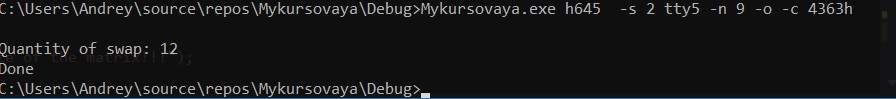


Рис.4 Пример работы программы при вводе сторонних значений

Во время работы приложения происходит создания матрицы по указанным параметрам или параметрам по умолчанию (Рис. 5). Стоит сказать, что в случае, если пользователь не задает проверку на стабильность, то все значения в матрице будут уникальными. Затем происходит сортирование матрицы (Рис.6), и в итоге мы получаем конечный результат в виде отсортированной матриц, где можно увидеть демонстрацию стабильности сортировки в случае вызова этой функции. (Рис.7).

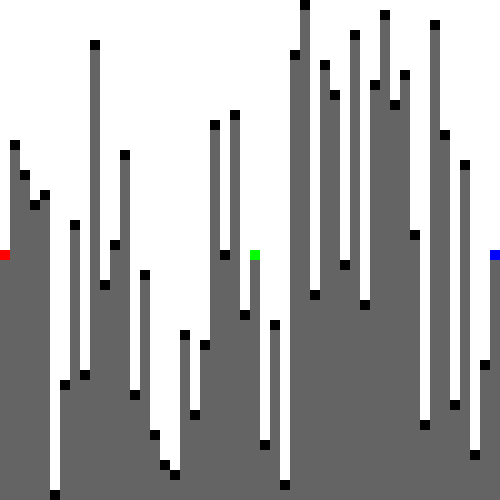


Рис.5 Создание матрицы

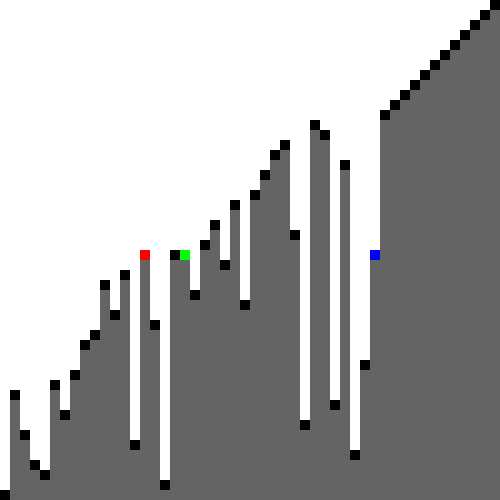


Рис.6 Сортирование матрицы

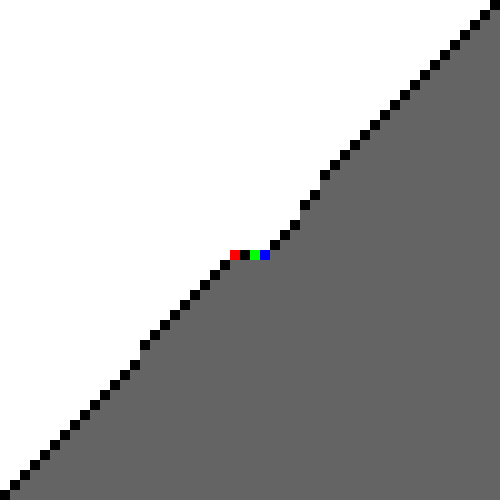
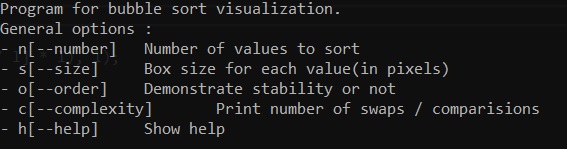


Рис.8 Конечный результат с демонстрацией стабильности

### **2.3 Описание меню программы**



1. – Размер матрицы
2. – Размер пикселя
3. – Демонстрация стабильности
4. – Печать количества обменов
5. – Показ справки

### **Заключение**

В результате выполненной работы спроектировано и разработано консольное приложение, которое реализовано на языке Cи с использованием библиотеки libpng. Приложение демонстрирует этапы работы алгоритма пузырьковой сортировки с помощью серии PNG кадров. В работе приводятся и анализируются ключевые особенности алгоритма сортировки пузырьком. Демонстрируются основные этапы создания приложения. Также демонстрируется пример работы самого приложения.

В процессе работы я приобрел более глубокие знания по построению программы на языке Си и научился работать со встроенной библиотекой libpng. А также создавать apng файлы.

**Список используемой литературы**

1. Сортировка пузырьком [электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/Сортировка_пузырьком> (дата обр. 04.05.2021)
2. Библиотека libpng [электронный ресурс]. -URl: <https://sourceforge.net/projects/libpng/files/> (дата обр. 12.04.2021)
3. Репозиторий GitHub с исходным кодом приложения [электронный ресурс]. – URL: <https://github.com/Andrew-Ko-D/pr1/blob/main/main.c>

### **Приложение А**

