Федеральное агентство по образованию и науке РФ

Государственное образовательное учреждение высшего профессионального образования

«Ижевский государственный технический университет им. М.Т. Калашникова»

Кафедра «Программное обеспечение»

Работа защищена с оценкой

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к проектно-технологической практике

по дисциплине «Программирование»

на тему: «Графические API-функции»

Выполнили

студенты гр. Б20-191-3 К. Ф. Кисматов,

Д. Д. Бугаков,

Р. А. Лыков

Проверил

доцент кафедры ПО Л.Н.Левицкая

Рецензия:

степень достижения поставленной цели работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

полнота разработки темы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

уровень самостоятельной работы обучающегося \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

недостатки работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ижевск

2021

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc74000830)

[1. ЦЕЛЬ РАБОТЫ 4](#_Toc74000831)

[2. ПОСТАНОВКА ЗАДАЧИ 4](#_Toc74000832)

[3. ТЕОРИЯ 4](#_Toc74000833)

[3.1. Сортировки 4](#_Toc74000834)

[3.2. Программная реализация 5](#_Toc74000835)

[4. ПРИМЕР РАБОТЫ ПРОГРАММЫ 5](#_Toc74000836)

[ЗАКЛЮЧЕНИЕ 6](#_Toc74000837)

# ВВЕДЕНИЕ

В современном мире мы привыкли к тому, что любой процесс может быть визуализирован, что позволяет нам увидеть, как он работает, и лучше его понять. Хорошим примером может послужить визуализация алгоритма сортировки. Именно поэтому мы решили реализовать несколько алгоритмов сортировки и изобразить их работу, чтобы наглядно увидеть разницу между ними.

# ЦЕЛЬ РАБОТЫ

Научиться создавать графические объекты и анимировать их, используя API-функции.

# ПОСТАНОВКА ЗАДАЧИ

Создать анимацию сортировки массива на языке C++ в среде программирования Visual Studio 2019.

# ТЕОРИЯ

## Сортировки

1. Пузырьковая сортировка  
   Пузырьковая сортировка сравнивает каждые два соседние элемента массива, проверяя, находятся ли они в правильном порядке. Сложность алгоритма .
2. Сортировка встряской  
   Сортировка встряской похожа на пузырьковую сортировку. Вместо того, чтобы сравнивать два соседних элемента с начала, сортировка встряской сравнивает элементы в обе стороны, проходя сначала справа налево, а затем слева направо. Сложность алгоритма .
3. Сортировка вставками  
   Сортировка вставками находит в массиве максимальный элемент и устанавливает его в самый конец, затем повторяет такое действие, пропуская последний элемент. Сложность алгоритма .
4. Гномья сортировка  
   Гномья сортировка рассматривает отсортированный отрезок массива. Находя элемент, который меньше начала отрезка, сортировка ищет место, в которое необходимо вставить элемент. Сложность алгоритма .
5. Сортировка расческой  
   Сортировка расческой является улучшенной пузырьковой. Вместо того, чтобы сравнивать каждые два соседние элемента, сортировка расческой сначала сравнивает элементы на большом расстоянии, постепенно уменьшая это расстояние. Сложность алгоритма .
6. Сортировка слиянием  
   Сортировка слиянием является одной из самых быстрых сортировок. Предположим, что наш массив состоит из двух отсортированных массивов. Тогда сложность сортировки . Разделим массив на два, отсортируем их и сольем в один. Сложность алгоритма .
7. Обезьянья сортировка  
   Будем случайным образом переставлять элементы массива, пока не отсортируем массив. Сложность алгоритма .

## Программная реализация

Мы использовали заголовочный файл Graphics.h, в котором есть необходимые нам графические функции (Set\_pen(), Set\_brush(), Rectangle()). Также мы написали свой класс Stolb, в котором описаны методы для удобной визуализации числа. Массив, который мы сортируем представлен как vector<Stolb>.

Для анимирования сортировки были написаны функции, которые плавно изменяют размер столба до необходимого.

# ПРИМЕР РАБОТЫ ПРОГРАММЫ

Изображение выглядит как текст, музыка, синтезатор

Автоматически созданное описание

Рис. 1. Пример визуализации сгенерированного массива

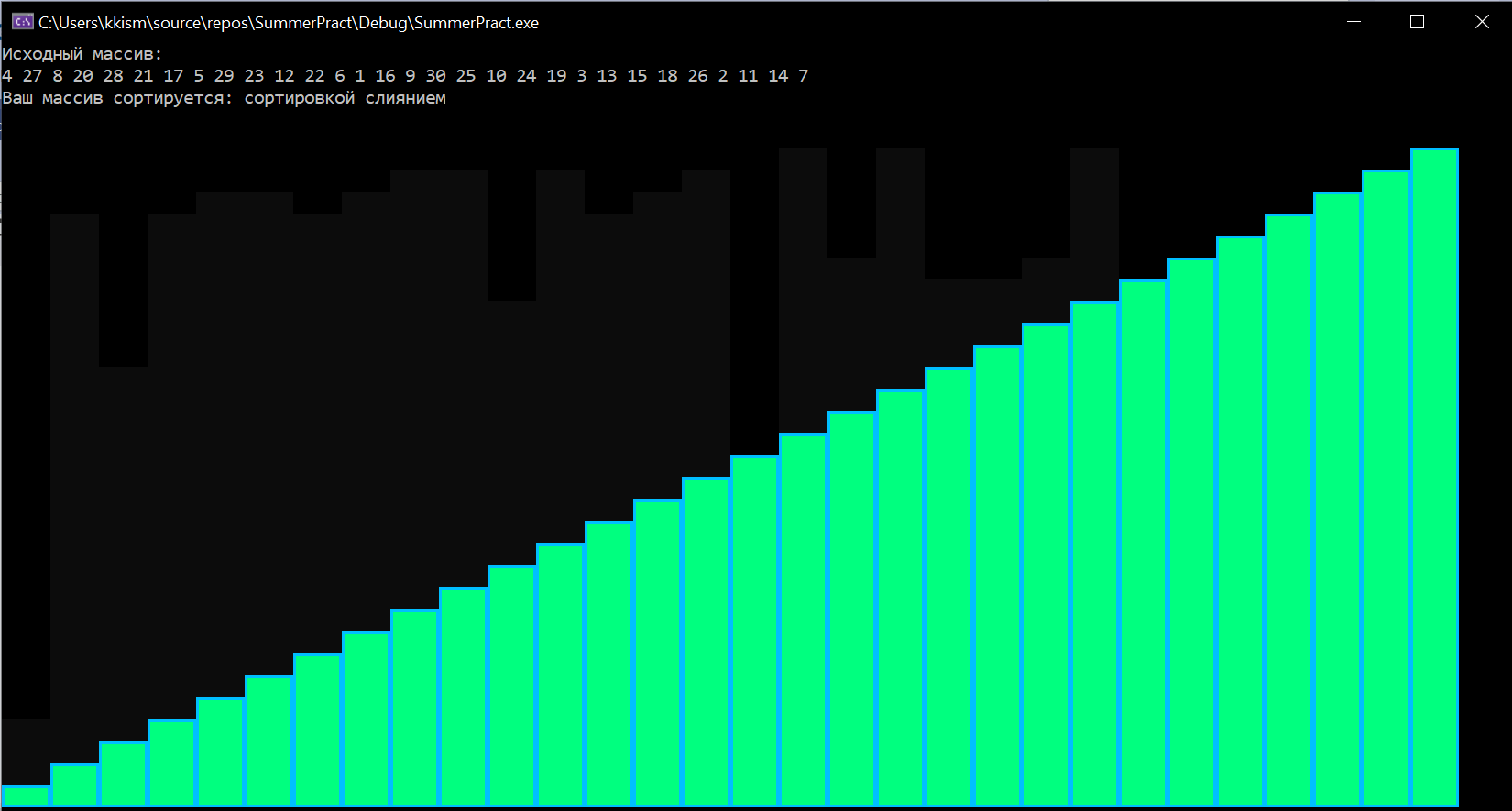


Рис. 2. Результат работы программы (отсортированный массив)

# ЗАКЛЮЧЕНИЕ

В ходе практической работы была написана программа, визуализирующая различные алгоритмы сортировки массива с помощью API-функций.