# Клещенко В. Д.

# ЛР #1: Алгоритмы сортировки

## Цель

Познакомить студента с основами анализа алгоритмов на примере операций сортировки.

**Перечень алгоритмов сортировки:**

* сортировка пузырьком,
* сортировка выбором,
* сортировка вставкой,
* сортировка подсчетом,
* цифровая сортировка (LSD, MSD) ,
* сортировка слиянием,
* шейкерная сортировка (сортировка перемешиванием),
* быстрая сортировка (сортировка Хоара),
* сортировка кучей (пирамидальная сортировка).

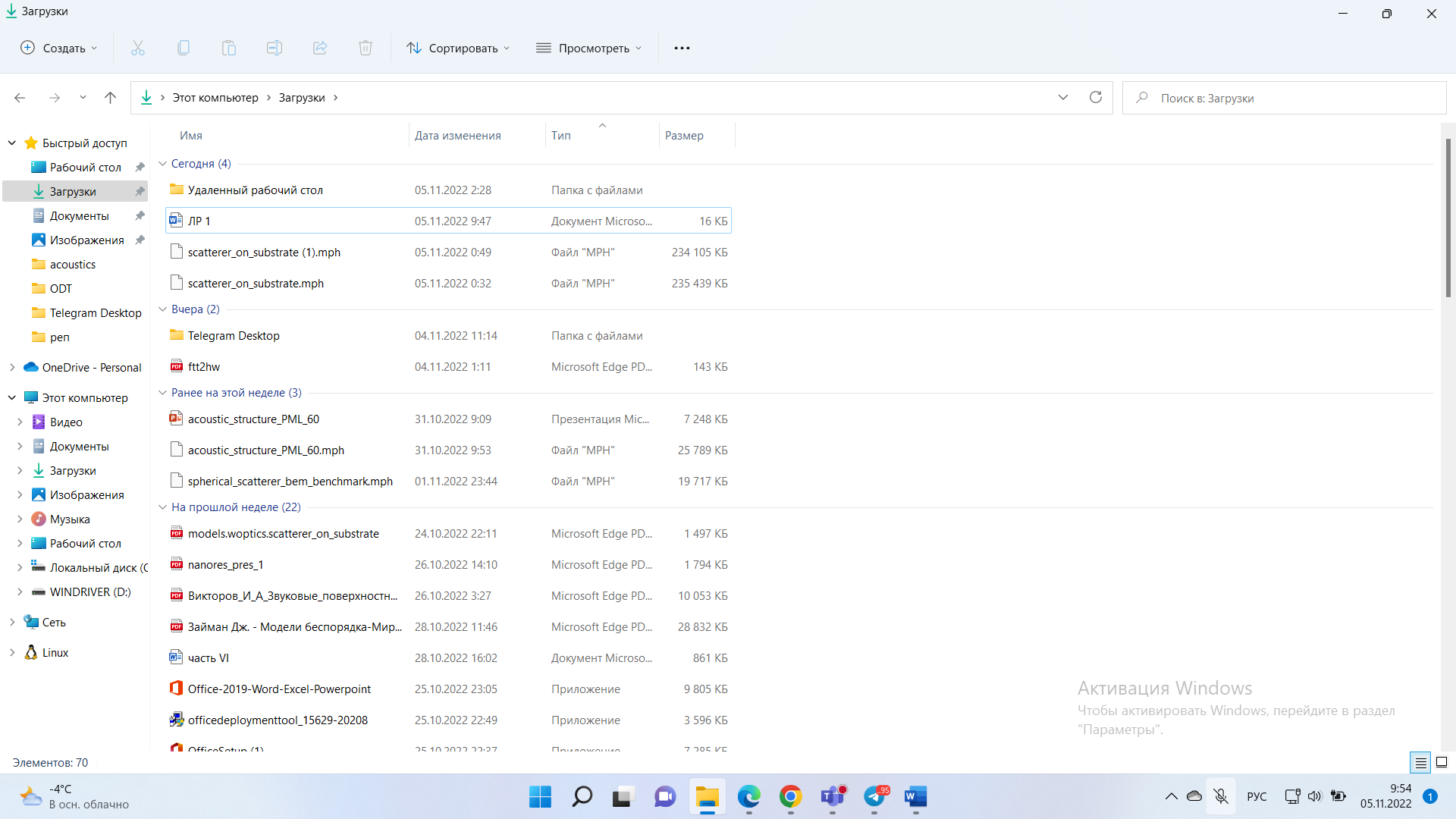
## Задача

1. **[СОРТ 1] Алгоритм сортировки 1**
   1. Сформулировать задачу, для которой требуется применение алгоритма сортировки.
   2. Выбрать алгоритм сортировки. Доказать, что выбранный алгоритм является наилучшим выбором. Привести расчет ёмкостных и вычислительных затрат.
   3. Реализовать алгоритм на языке C++.
2. **[СОРТ 2] Алгоритм сотрировки 2**
   1. Сформулировать задачу, для которой требуется применение алгоритма сортировки. Алгоритм сортировки 2 должен отличаться от алгоритма сортировки 1. Асимтотическая сложность алгоритма 2 должна быть больше, чем у алгоритма 1.
   2. Выбрать алгоритм сортировки. Доказать, что выбранный алгоритм является наилучшим выбором. Привести расчет ёмкостных и вычислительных затрат.
   3. Реализовать алгоритм на языке C++.
3. **[SAVE] Всё, что было сделано в шагах 1-2, сохранить в репозиторий (+ отчет по данной ЛР в папку doc).**

## Решение

1. **[СОРТ 1] Алгоритм сортировки 1**
   1. Сформулировать задачу, для которой требуется применение алгоритма сортировки.

Задача заключается в реализации сортировки которая бы наиболее эффективно сортировала файлы по датам / сортировала файлы по размеру, размер которых не сильно отличается. Это может быть использовано для проведения анализа кода и т.д.



* 1. Выбрать алгоритм сортировки. Доказать, что выбранный алгоритм является наилучшим выбором. Привести расчет ёмкостных и вычислительных затрат.

Алгоритмом сортировки: LSD – поразрядная сортировка с использованием сортировки подсчетом. На каждой итерации сравниваются разряды начиная с младшего у исходных чисел. Для чисел, у которых количество цифр почти одинаковое это хороший метод, так как все равно сравнивать придется практически все разряды. При сравнении разрядов использовалась сортировка подсчетом – ее выгодно использовать при небольшом разбросе значений, а цифр всего 9.

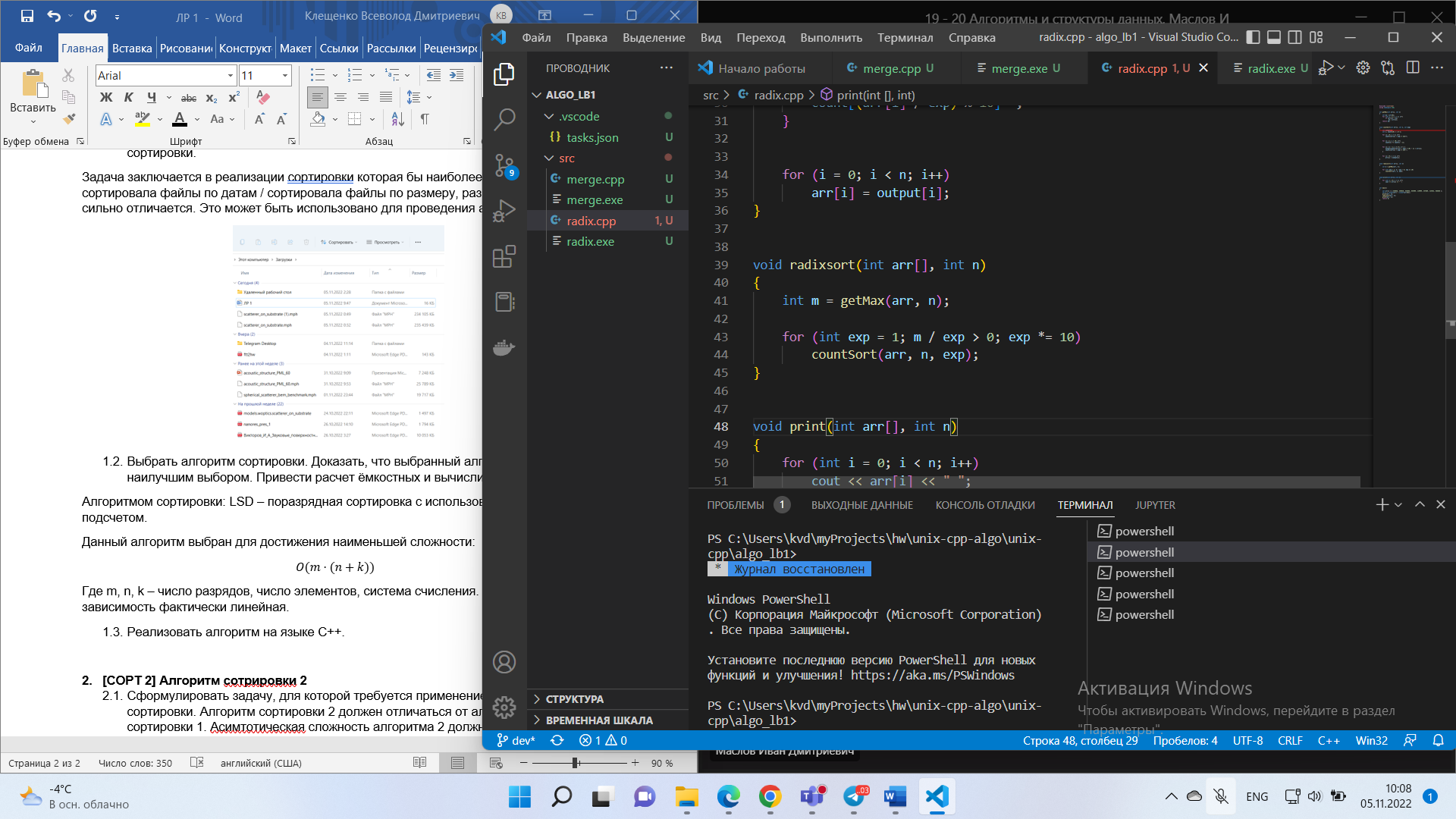
Данный алгоритм выбран для достижения наименьшей сложности:

Где m, n, k – число разрядов, число элементов, система счисления. При малых m и k – зависимость фактически линейная. Второй множитель происходит из сортировки подсчетом: нужно создать вспомогательный массив, состоящий из нулей в котором будет записываться частота каждого элемента входного массива, затем последовательно прочитать элементы входного массива. Теперь достаточно пройти по вспомогательному массиву и записать числа в отсортированном виде. Первый множитель – из поразрядной сортировки – уже реализовав сортировку подсчетом необходимо просто пробежаться по массиву один раз.

По затратам памяти:

Это следует из описанного выше, с учетом того, что дополнительные затраты по памяти используются лишь для сортировки подсчетом.

* 1. Реализовать алгоритм на языке C++.



1. **[СОРТ 2] Алгоритм сортировки 2**
   1. Сформулировать задачу, для которой требуется применение алгоритма сортировки. Алгоритм сортировки 2 должен отличаться от алгоритма сортировки 1. Асимптотическая сложность алгоритма 2 должна быть больше, чем у алгоритма 1.

Заметим, что если размер файлов сильно отличается, то бессмысленно сравнивать все разряды и лучше использовать алгоритм с большей асимптотической сложности, но он все равно будет быстрее.

* 1. Выбрать алгоритм сортировки. Доказать, что выбранный алгоритм является наилучшим выбором. Привести расчет ёмкостных и вычислительных затрат.

Алгоритм сортировки: Сортировка слиянием – Merge Sort. Это одна из самых быстрых сортировок которое всегда (лучшее, среднее и худшее время) работает одинаково – подходит в том числе и для сортировки файлов размер которых сильно отличается.

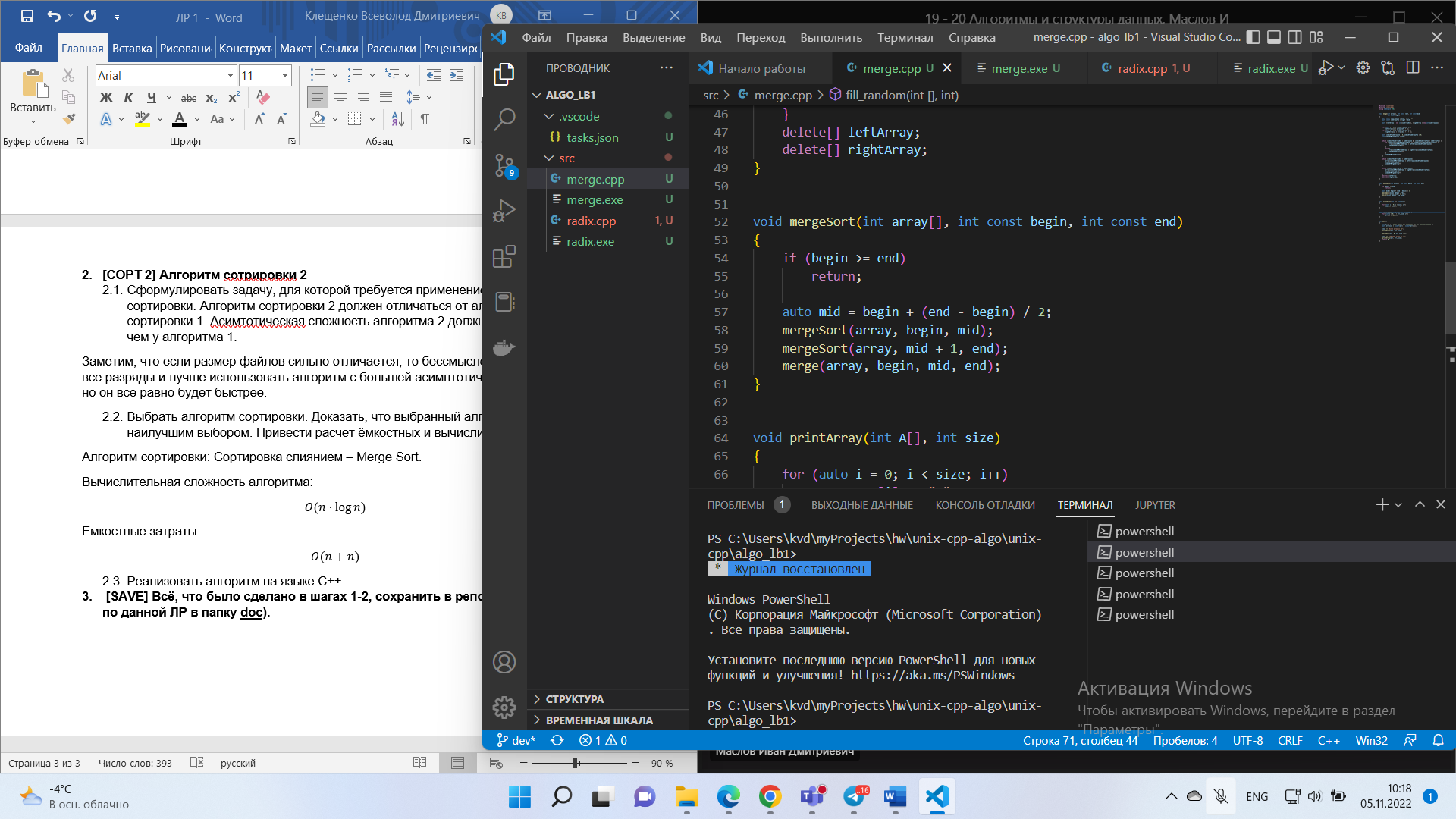
Вычислительная сложность алгоритма:

Это следует из того, что сортировка слиянием происходит с поэтапным деление исходного массива пополам и сравнением этих частей, начиная с последней. Логарифм получается из количества таких делений пополам (количество уровней), а n – из сравнения каждых двух частей (их размеры отличаются, но асимптотически – n)

Емкостные затраты:

Это дополнительные затраты по памяти, они получаются из того, что необходимо хранить результаты слияния, при сравнении частей.

* 1. Реализовать алгоритм на языке C++.



## Заключение

В результате лабораторной работы был произведен анализ алгоритмов на примере операций сортировки. Было исследовано, что в некоторых случаях необязательно сравнивать только асимптотическую сложность, а также реализованы алгоритмы сортировки включающие в себя несколько типов сортировок.