

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №1
по дисциплине
«Низкоуровневое программирование»
Вариант №1

Студент:

Лунева Арина Алексеевна

Группа Р33312

Преподаватель:

Кореньков Юрий Дмитриевич

Санкт-Петербург, 2023

Цель работы: создать модуль, реализующий хранение в одном файле данных (выборку, размещение и гранулярное обновление) информации общим объёмом от 10GB соответствующего варианту вида.

Задачи:

1. Спроектировать структуры данных для представления информации в оперативной памяти
 - a. Для порции данных, состоящий из элементов определённого рода (см форму данных), поддерживать тривиальные значения по меньшей мере следующих типов: четырёхбайтовые целые числа и числа с плавающей точкой, текстовые строки произвольной длины, булевские значения
 - b. Для информации о запросе
2. Спроектировать представление данных с учетом схемы для файла данных и реализовать базовые операции для работы с ним:
 - a. Операции над схемой данных (создание и удаление элементов схемы)
 - b. Базовые операции над элементами данных в соответствии с текущим состоянием схемы (над узлами или записями заданного вида)
 - i. Вставка элемента данных
 - ii. Перечисление элементов данных
 - iii. Обновление элемента данных
 - iv. Удаление элемента данных
3. Используя в сигнатурах только структуры данных из п.1, реализовать публичный интерфейс со следующими операциями над файлом данных:
 - a. Добавление, удаление и получение информации об элементах схемы данных, размещаемых в файле данных, на уровне, соответствующем виду узлов или записей
 - b. Добавление нового элемента данных определённого вида
 - c. Выборка набора элементов данных с учётом заданных условий и отношений со смежными элементами данных (по свойствам/полями/атрибутам и логическим связям соответственно)
 - d. Обновление элементов данных, соответствующих заданным условиям
 - e. Удаление элементов данных, соответствующих заданным условиям
4. Реализовать тестовую программу для демонстрации работоспособности решения

Аспекты реализации:

Описание программы:

Программа включает в себя несколько модулей:

commands – содержит функции пользовательских команд

utils – содержит вспомогательные функции для работы с файлами (*file_utils*), структурами (*structs_utils*) и строками (*string_utils*)

crud – содержит функции манипуляции с данными

generator – содержит функции для генерации пустого файла

Описание структур:

```
struct tree_subheader {  
    uint64_t ASCII_signature;
```

```

        uint64_t root_offset;
        uint64_t first_seq;
        uint64_t second_seq;
        uint64_t current_id;
        uint64_t pattern_size;
};

struct key_header {
    uint32_t size;
    uint32_t type;};
struct key {
    struct key_header* header;
    char* key_value;
};

struct tree_header {
    struct tree_subheader* subheader;
    struct key** pattern;
    uint64_t* sequence_id;
};

union tuple_header {
    struct {
        uint64_t parent;
        uint64_t alloc;
    };
    struct {
        uint64_t previous;
        uint64_t next;
    };
};

struct tuple {
    union tuple_header header;
    uint64_t* data;
};

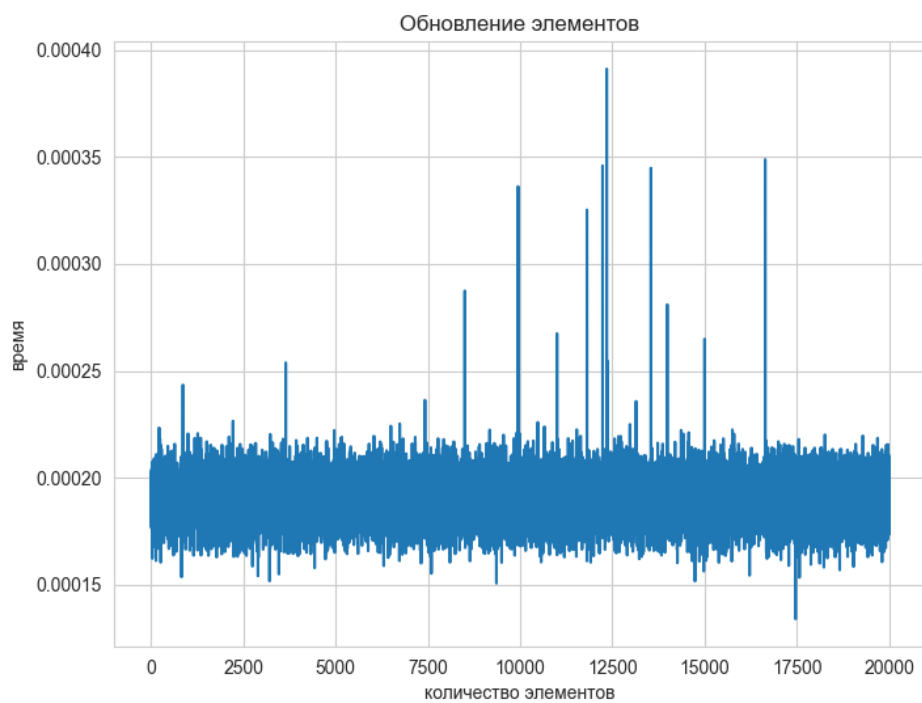
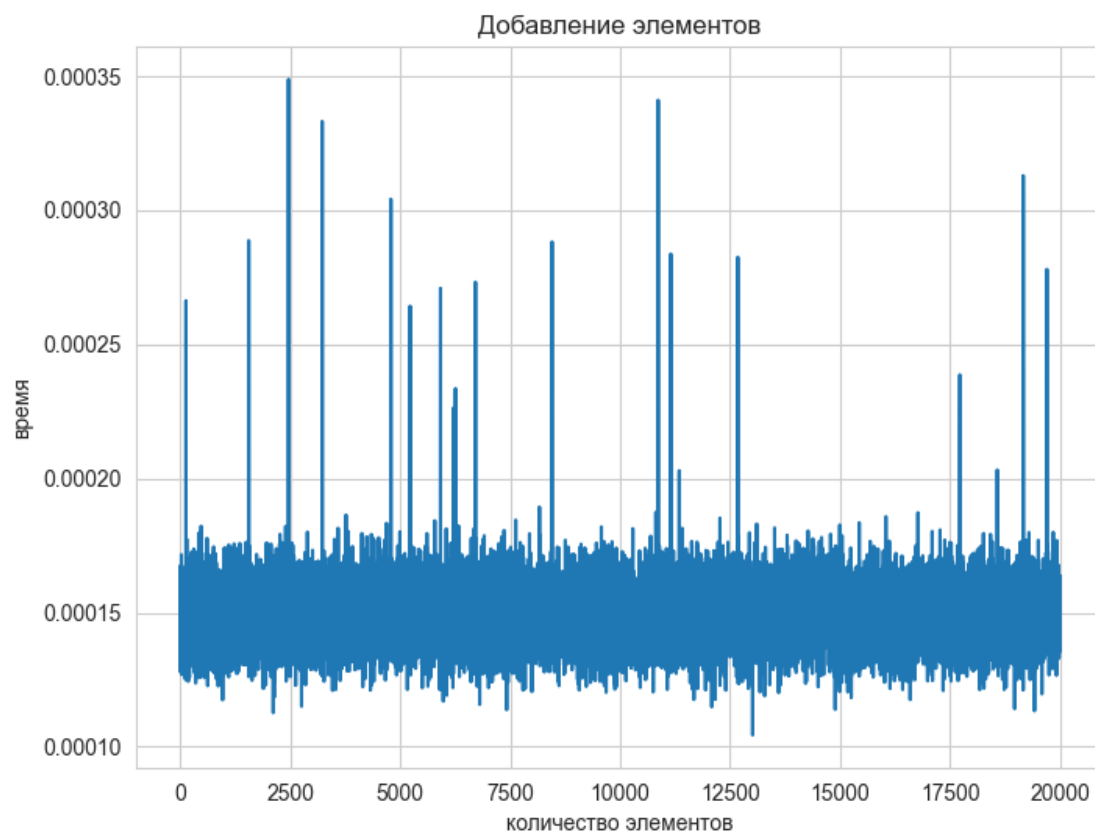
struct tuple_result_list {
    struct tuple* value;
    uint64_t tuple_id;
    struct tuple_result_list* previous;
};

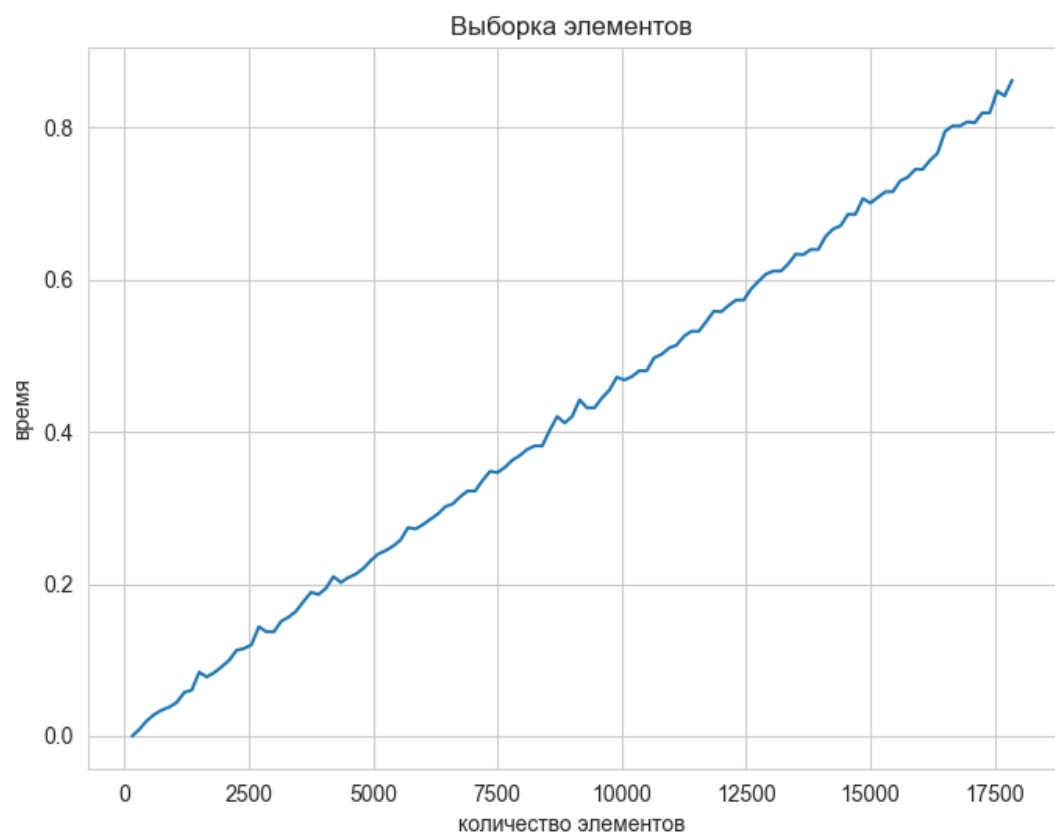
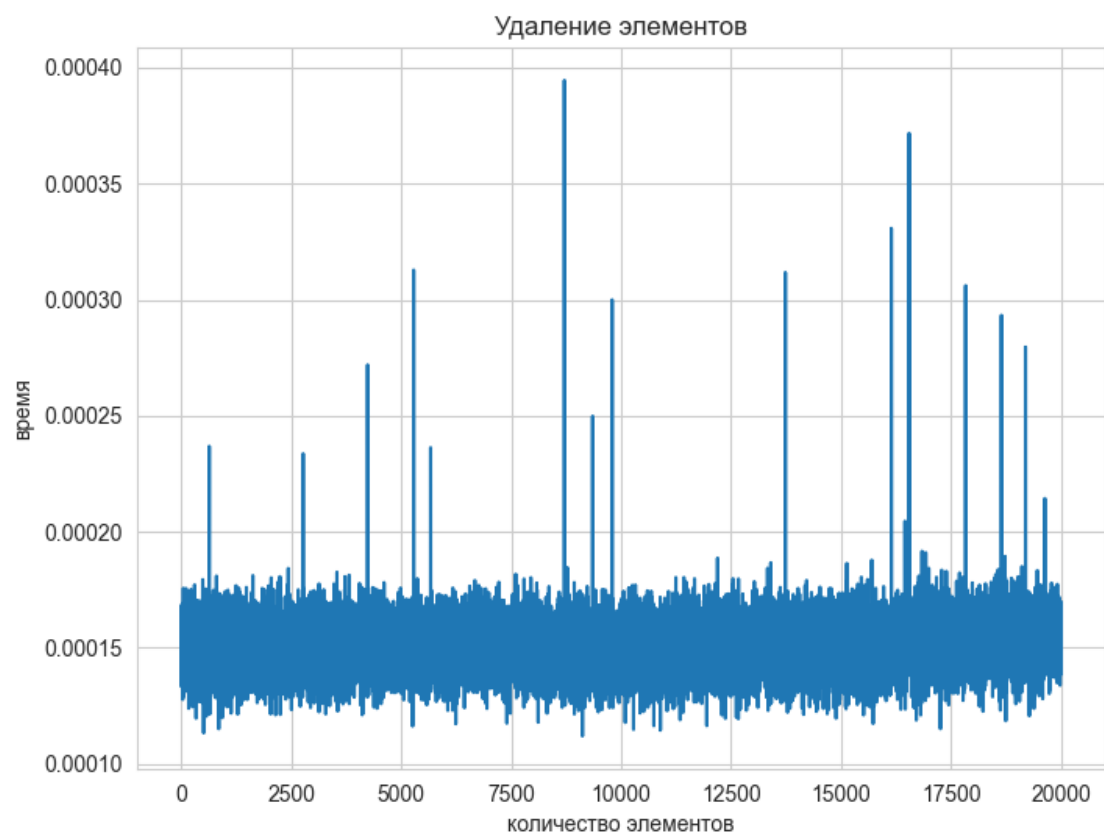
```

Особенности реализации:

Поиск элементов возможен по id, родителю или условию, для пользователя эти команды были разделены (он может ввести одну из команд: find_all, find_by_parent, find_by_id, find_by_condition).

Графики:





Вывод: в результате выполнения лабораторной работы был разработан модуль, реализующий хранение в одном файле данных в виде дерева с атрибутами, объем которых может достигать 10GB.