Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1

по «Низкоуровневое программирование»

Вариант – 6(Дерево узлов с атрибутами)

Выполнил:

Студент группы P33102

Голощапов И.А.

Преподаватели:

Кореньков Юрий Дмитриевич

Санкт-Петербург

2023

Задание 1

Создать модуль, реализующий хранение в одном файле данных (выборку, размещение и гранулярное

обновление) информации общим объёмом от 10GB соответствующего варианту вида.

Порядок выполнения:

1. Спроектировать структуры данных для представления информации в оперативной памяти

a. Для порции данных, состоящий из элементов определённого рода (см форму данных),

поддержать тривиальные значения по меньшей мере следующих типов: цетырёхбайтовые

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 о 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

a. Параметры для всех операций задаются посредством формирования соответствующих структур

данных

b. Показать, что при выполнении операций, результат выполнения которых не отражает

отношения между элементами данных, потребление оперативной памяти стремится к O(1)

независимо от общего объёма фактического затрагиваемых данных

c. Показать, что операция вставки выполняется за O(1) независимо от размера данных,

представленных в файле

d. Показать, что операция выборки без учёта отношений (но с опциональными условиями)

выполняется за O(n), где n – количество представленных элементов данных выбираемого вида

e. Показать, что операции обновления и удаления элемента данных выполняются не более чем за

O(n*m) > t O(n+m), где n – количество представленных элементов данных обрабатываемого

вида, m – количество фактически затронутых элементов данных

f. Показать, что размер файла данных всегда пропорционален количеству фактически

размещённых элементов данных

g. Показать работоспособность решения под управлением ОС семейств Windows и *NIX

Исходный код

https://github.com/Ja-Vani/LLP1

Структура данных.

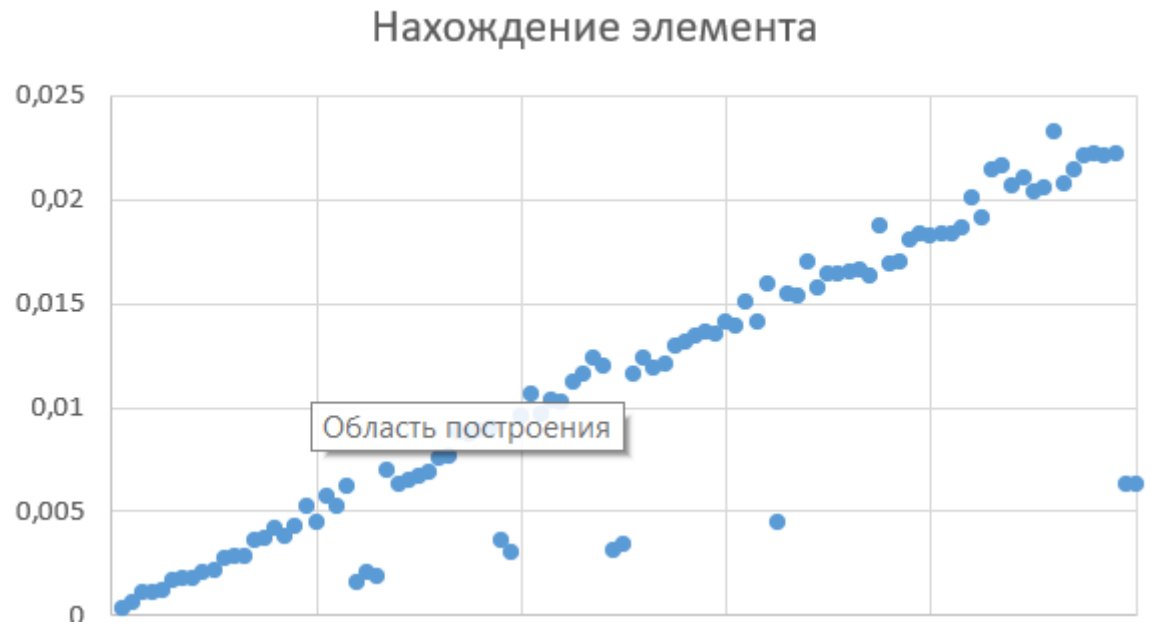
Существует структура узла. Внутри нее существует лист атрибутов и лист рёбер. Узел имеет имя(лейбл) и уникальный id. Атрибут имеет поле типа, название атрибута и поле, которое помещает в себя один из типов. Рёбра имеют имя, и id элемента, на который указывают.

Графики производительности.

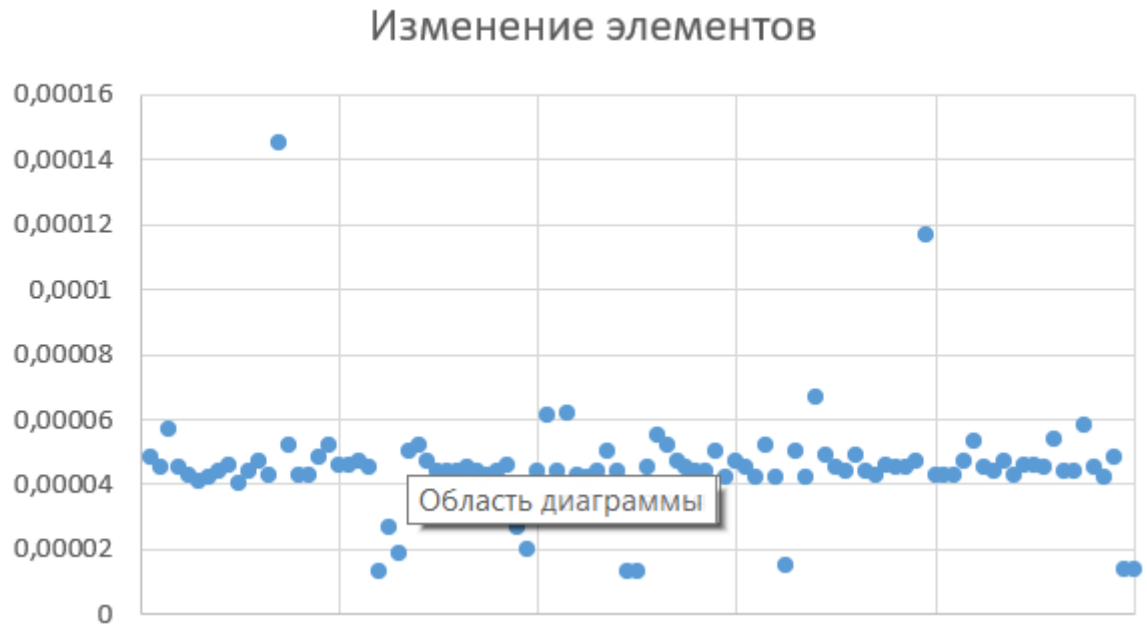
Вставка

Удаление

Поиска

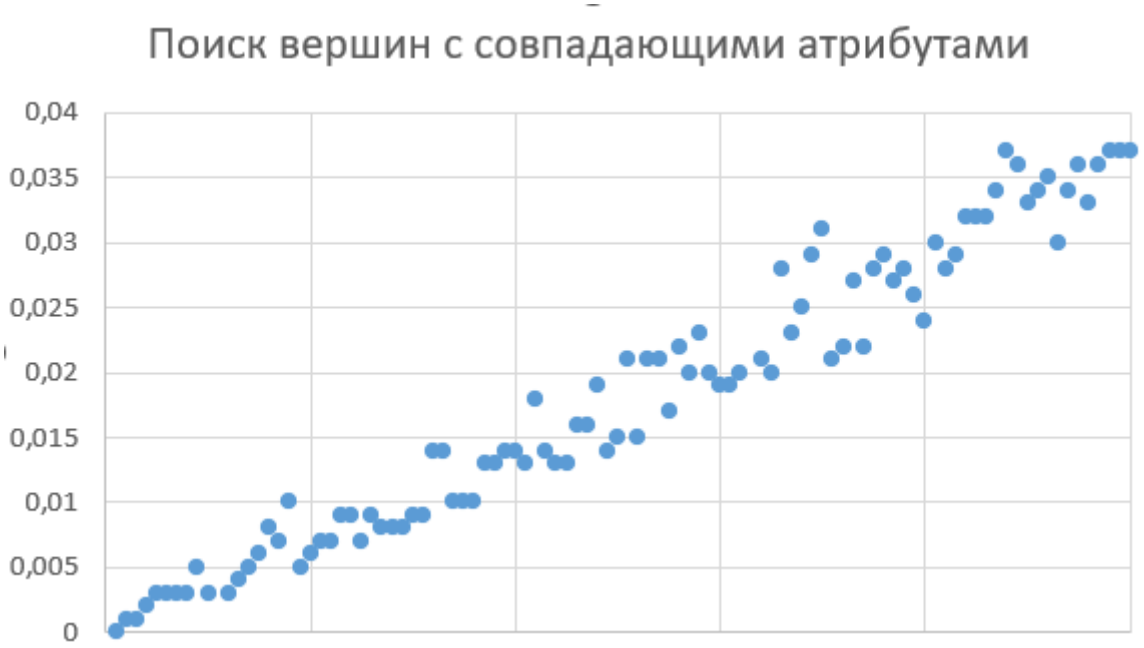


Изменение



Переход по ребру

Совпадение атрибутов



Вывод: в ходе выполнения работы была создана программа умеющая в файле хранить древо узлов с атрибутами, а так же совершать действия над этими узлами, не загружая все элементы из файла.