Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1 по Информатике

Вариант 5

Юнусов Роман Ильдарович
Группа P33102

Преподаватель Юрий Кореньков

Г.Санкт-Петербург

2023 г.

Задание

Реализовать модуль хранящий в одном файле множество таблиц, произвольного размера. Внутри ячейки таблицы может хранится целочисленный тип, числа с плавающей точкой, строка произвольного размера.

Необходимо уметь создать, удалять таблицы.

Также нужно уметь делать операции insert - O(1), select - O(N), delete - O(N), update - O(N) по отношению к записям в таблицах

Детали реализации

Поделим нам файл на list фиксированного размера.

У каждого листа есть в конце элемент со ссылкой на следующий лист.

Первый лист файла и следующие по цепочке хранят данные таблицы – номер таблицы, количество значений в одной записи, количество записей, последний и первый list принадлежащий странице.
Второй лист файла и другие в цепи хранят информацию о свободных листах.

Если лист файла не является частью цепочек первого и второго, то он хранит записи, принадлежащие, какой-то из таблиц.

Запись представляет собой последовательность ячеек cell, одна ячейка хранит либо число, либо строку длиной до 32 символов.

```
struct cell {
	enum type type_of;
	union {
		int int_data;
		double double_data;
		char string_data[string_data_size_in_cell];
	};
	enum cell_flag flag;
};
```

Соответственно комбинация ячеек, которая начинается со специальной(которая имеет flag `RAW_NUM`) – это запись в таблице.

Для реализации операций используют функции addRaw, select, delete, update.

Начнём с addRaw за константу.

Логика проста, для каждой таблицы поддерживаем последнюю страницу, которая ей принадлежит, и всегда стараемся добавлять в неё, если не хватает места, то просто запросим новую пустую страницу(она может быть либо освобожденной, либо просто создадим новую увеличив файл).

Для условий на операции заведем структуру

```
struct queryCondition {
	int stolbec_num;
	char sign;
	enum type type_of;
	union {
		int int_data;
		double double_data;
	};
};
```

Массив таких структур задаёт условия, по которым можно понять подходит запись условиям. Также если stolbec_num равен 0, то мы считаем, что речь идёт об абстрактных номерах записей, т.е. порядке их добавления.

Сами select и delete выполнены просто.
Идём по содержимому таблицы начиная с её первой страницы и выполняем операции в цикле

1) Взять raw
2) Проверить raw
3) Если raw подходит то исполнить операцию
4) Найти адрес следующего raw

Также конкретно для delete после удаления у нас могут быть множества пустых страниц, значит после удаления нужно пробежаться по всем страницам таблицы, и зарегистрировать пустые как свободные.

Графики.

График времени добавления

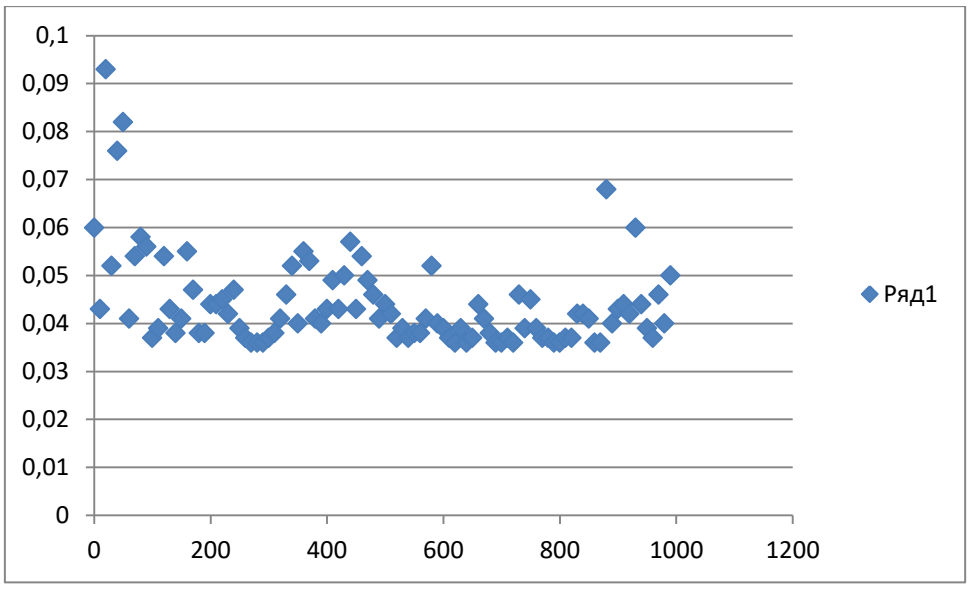

График времени удаления

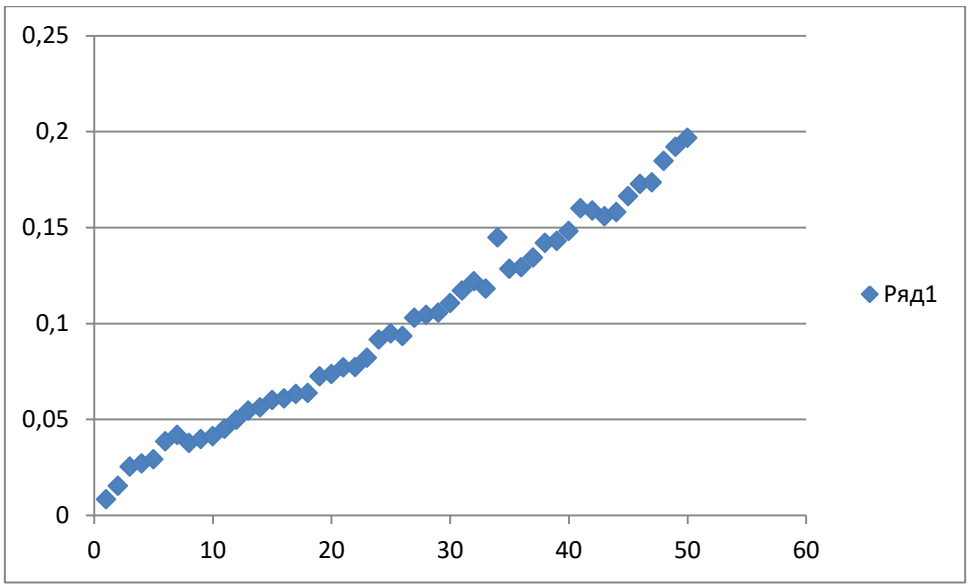

График select