Министерство цифрового развития,

связи и массовых коммуникаций

Федеральное государственное

образовательное бюджетное учреждение

высшего профессионального образования

«Сибирский государственный университет   
телекоммуникаций и информатики»

(СибГУТИ)

**Кафедра ПМиК**

**Курсовая работа по дисциплине**

**«‎Структуры и алгоритмы обработки данных»**

**Вариант 32**

**Выполнил студент**

**группы ИП-212:**

Жеребцов Дмитрий

Новосибирск 2023

**Введение и постановка задачи**

Общая постановка задачи

Хранящуюся в файле testBase4.dat базу данных (4000 записей) загрузить динамически в оперативную память компьютера в виде массива или списка (в зависимости от варианта), вывести на экран по 20 записей (строк) на странице с возможностью отказа от просмотра.

В качестве ключа для упоpядочения нужно взять всего по нескольку (обычно не менее тpех) байт из соответствующих полей. Файл базы данных загpужается в динамическую память в виде списка, который соpтиpуется цифpовым методом. Затем, для проведения быстрого поиска, по отсортированному списку строится индексный массив. Вывести на экран.

Предусмотреть возможность быстрого поиска по ключу (первые три буквы названия улицы) в упорядоченной базе, в результате которого из записей с одинаковым ключом формируется очередь, содержимое очереди выводится на экран.

Из записей очереди построить АВЛ-дерево поиска по ключу, отличному от ключа сортировки, вывести на экран содержимое дерева и предусмотреть возможность поиска в дереве по запросу.

Закодировать файл базы данных кодом Гилберта-Мура, предварительно оценив вероятности всех встречающихся в ней символов. Построенный код вывести на экран, вычислить среднюю длину кодового слова и сравнить ее с энтропией исходного файла.

Дополнительное задание. Упакованную базу данных записать в файл, вычислить коэффициент сжатия данных.

**Основная идея**

***Цифровая сортировка:***

Для сортировки базы данных используется Цифровая сортировка, благодаря тому, что мы можем взять лишь несколько бит из полей для сортировки, достигается быстрая скорость упорядочивания данных. Сортировка происходит последовательно, сначала по номеру квартиры, потом по номеру дома, а следом по улице. Сортировки происходят поочередно, это достигается при помощи цифровой сортировки.

***Списки и индексный массив:***

Для хранения обрабатываемой информации используются списки и индексный массив указателей на элементы списка для удобства обработки данных. Списки тут будут удобны тем, что нам не придётся знать конечное количество элементов, то-есть нам не придётся перевыделять память.

Индексный массив предлагает более простой способ управления данными, которые находятся в списке, так как доступ к списку может быть только последовательным, то индексный массив решает данную проблему. Так же благодаря индексному массиву мы не обязаны изменять хранимые данные, можем изменять только индексный массив.

***АВЛ-дерево:***

АВЛ-дерево используется для быстрого поиска*,* мы берем данные из очереди, чтобы добавить их в дерево по другому ключу и провести поиск, так как ключи бывают одинаковые, то записываем их в динамический массив (vector), при поиск по ключу выводим массив данных.

***Бинарный поиск и поиск всех ключей:***

Бинарный поиск используется для быстрого нахождения ключа в отсортированном массиве за O(log2(n)). Мой вариант бинарного поиска ищет самое правое вхождение ключа, это создано для реализации поиска всех возможных ключей в базе данных. Поиск всех ключей работает при помощи бинарного поиска, мы берем ключ из первого элемента, далее ищем его конечный индекс, перемещаемся на следующий элемент записываем данные в массив типа Point (содержит ключ и начальный и конечный индексы ключа), поиск всех ключей происходит до конца данных.

***Код Гилберта-Мура:***

Используем статический код, чтобы закодировать базу данных и произвести сжатие данных.