Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Кафедра Информатики и прикладной математики

Дисциплина: Алгоритмы и структуры данных

Лабораторная работа №1

Вариант №5

Выполнил Григорьев Г.Г, гр. P3217

Преподаватель: Зинчик А.А.

Санкт-Петербург, 2018 г.

Цель работы: изучить основные методы организации таблиц

идентификаторов, получить представление о преимуществах и

недостатках, присущих различным методам организации таблиц

символов (идентификаторов).

Для выполнения лабораторной работы требуется написать программу, которая получает на входе набор идентификаторов, организует таблицу по заданному методу и позволяет осуществить **многократный** поиск идентификатора в этой таблице. Список идентификаторов считать заданным в виде текстового файла. Длина идентификаторов ограничена 32 символами.

Во всех вариантах требуется разработать программу, реализующую

комбинированный способ организации таблицы идентификаторов. Для организации таблицы используется простейшая хэш-функция, указанная в варианте задания, а при возникновении коллизий используется дополнительный метод размещения идентификаторов в памяти. Если в качестве этого метода используется дерево или список, то они должны быть связаны с элементом главной хэш-таблицы.

В каждом варианте требуется, чтобы программа сообщала среднее число коллизий и среднее количество сравнений, выполненных для поиска идентификатора.

Вариант 5

Хеш-функция – Сумма кодов первых трех букв

Способ разрешения коллизий – Рехеширование с использованием случайных чисел

Схема организации таблицы

Таблица представляет собой 2 массива, где каждый элемент

является элементом (ключом) первого списка и элементом (значением) второго списка. При помощи хеш-функции

определяется индекс обоих списков и далее

производятся операции с ними.

Описание алгоритма поиска в хеш-таблице

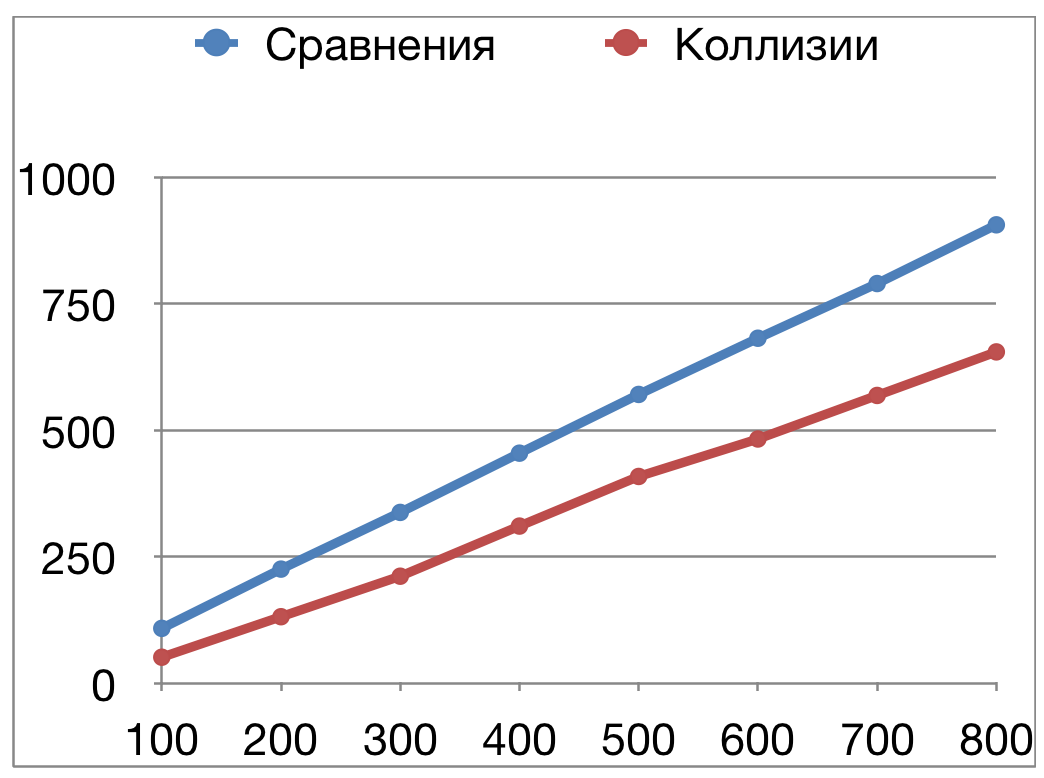
Поиск производится следующим способом: на вход подается

входящая строка, при помощи хеш-функции вычисляется ее хеш

(складываются коды первого, второго и третьего символов), далее в список ключей,

находящийся в таблице по заданному индексу производится

обращение, и проверяется наличие элемента методом путем спуска

от вершины списка.

По графикам видно, что зависимость количества сравнений

линейна, так как алгоритм поиска в списке проходит по всем элементам – O(n). Зависимость

количества коллизий от количества элементов так же линейна, т.к. Повторы встречаются равномерно в силу хеш-функции.

Вывод: хеш-таблицы хорошо представляют данные в виде пар ключ-значение и осуществляют удобный доступ к ним, однако при выборе плохой хеш-функции теряется эффективность в силу линейного возрастания количества сравнений и коллизий.