САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

*Кафедра информатики и прикладной математики*

**Лабораторная работа №1**

*«Организация простой хеш-таблицы»*

Выполнил:

студент II курса группы 2125

Припадчев Артём

Проверил:

Зинчик А.А.

Санкт-Петербург

2014

**Цель работы:** изучить основные методы организации таблиц идентификаторов,

получить представление о преимуществах и недостатках, присущих различным методам

организации таблиц символов (идентификаторов).

**Задание:** 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.

**Тип хеш-функции:** сумма кодов первой и второй букв

**Способ разрешения коллизий:** упорядоченный список с логарифмическим поиском.

**Бинарный поиск**

Идея алгоритма заключается в постепенном сжатии границ поиска. Пусть задана монотонно неубывающая функция F и начальный интервал [l, r). На каждой итерации вычислим m — середину текущего интервала, после чего сравним значение функции в этой точке с целевым значением f. Если F(m) < f, то необходимо увеличить левую границу l = m, иначе — уменьшить правую r = m. Если алгоритм работает с непрерывным интервалом, то итерации следует проводить пока длина интервала поиска больше заданной точности r-l > EPS.



**Структура программы**

****

В процессе выполнения были изучены основные методы организации таблиц идентификаторов и получены следующие **выводы:**

* На больших объемах данных хеш-таблицы значительно ускоряют поиск элемента. По хеш‑коду определяется корзина, в которой находится от 0 до нескольких элементов. Дальше в этой корзине ищется нужный элемент. И т.к. поиск проводится на совсем небольшом списке, скорость можно оценивать как О(1). Следовательно, время доступа к элементу зависит лишь от времени вычисления хеша (при отсутствии коллизии).
* Используемая в варианте хеш-функция крайне неэффективна. Вероятность коллизии очень велика. Для ее разрешения используется упорядоченный список с логарифмическим поиском.
* Максимальное время, необходимое для поиска, это время на выполнение хеш-функции, плюс O(log n) для поиска элемента в списке.