

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

Кафедра информатики и прикладной математики

Лабораторная работа №1

«Хеш-таблицы»

Выполнил Борzych А.А., гр. Р3217

Преподаватель Зинчик А.А.

Санкт-Петербург, 2015

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.

## Вариант 2

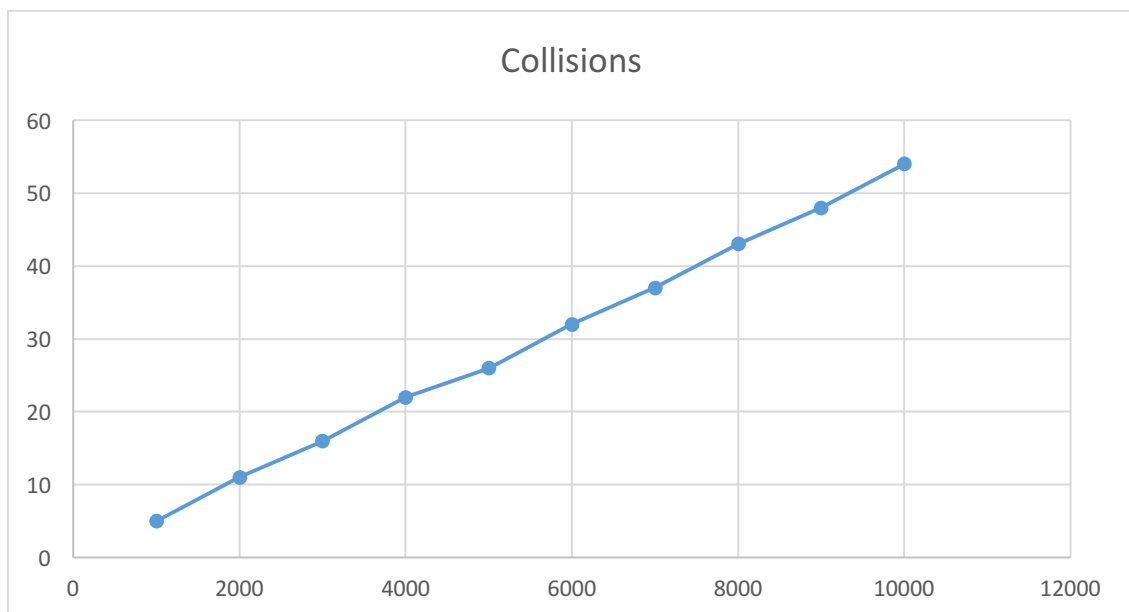
- Хеш-функция – сумма кодов первой и второй букв
- Способ разрешения коллизий – список с простым перебором

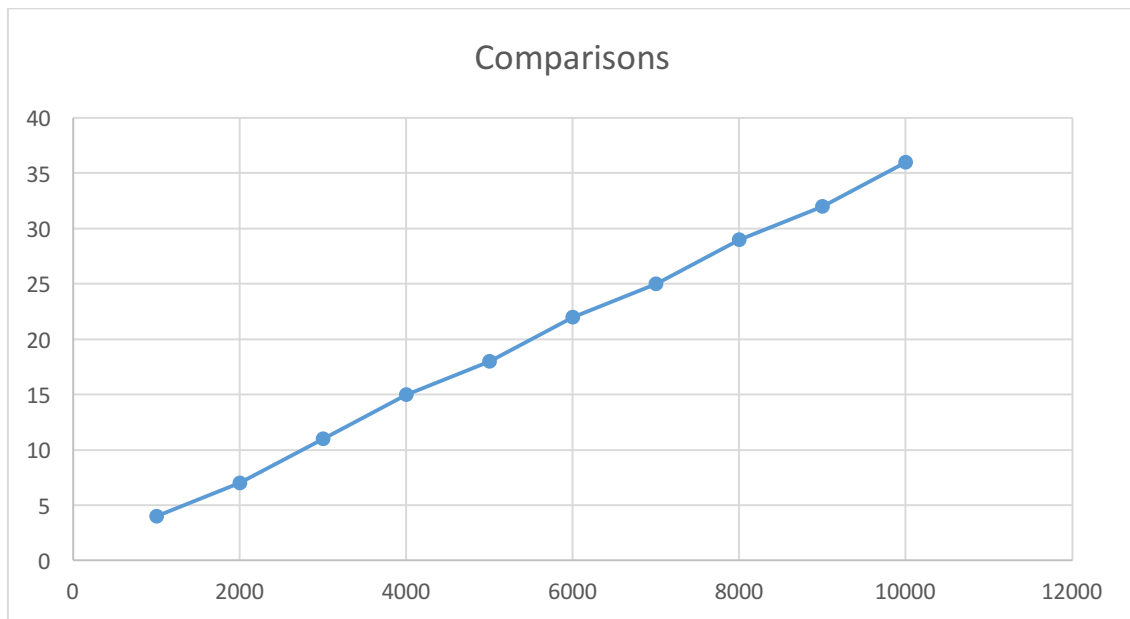
### Схема организации таблицы

Таблица представляет собой массив, где каждый элемент является односвязным списком. При помощи хеш-функции определяется индекс соответствующего списка, и далее производятся операции с ним (добавление, удаление, проверка на наличие определенного значения и так далее).

### 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 и второго символов), далее в список, 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 время поиска в списке  $O(n)$ . Зависимость количества коллизий от количества элементов линейно возрастает при большом количестве элементов, когда все ячейки таблицы содержат хотя бы по 1 значению.

Вывод: при помощи хеш-таблиц можно очень эффективно хранить уникальные данные, и быстро искать элементы. Однако преимущества могут быть легко нивелированы выбором плохой хеш-функции, которая генерирует недостаточное количество уникальных значений, в следствии чего увеличивается количество коллизий.