Университет ИТМО

Кафедра прикладной математики и информатики

Алгоритмы и структуры данных

Лабораторная работа № 1

Вариант 15

Выполнил:

Сорокин Юрий

Группа Р3217

Преподаватель:

Зинчик Александр Адольфович

СПб

2016 г.

# Лабораторная работа № 1

*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.

## Варианты заданий

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.

|  |  |  |
| --- | --- | --- |
|  | Бинарное дерево по последней букве | Список с простым перебором |

## Вставка элемента

Вычисляем значение хеш-функции n=h(A) для искомого элемента А.

Сравниваем значение n со значением n0 в корне дерева. Если n>n0, переходим в правое поддерево, иначе в левое. Сравниваем n со значениями узлов, пока не найдем n=ni или пустую ветку, куда вставляем данный элемент. Каждый узел дерева содержит список, при n=ni добавляем в список очередной элемент.

## Поиск элемента

Получаем значение хеш-функции элемента n=h(A). Бежим по дереву, пока не найдем ni=n. При нахождении такого элемента, бежим по списку узла дерева, пока не найдем нужный нам элемент или пока не пройдем все значения списка. Если узла не существует, или в списке нет данного значения, то элемент не найден.

## Вывод

В ходе данной работы, была реализована простейшая хеш-функция, бинарное дерево по последней букве, с методом разрешения коллизий, список с простым перебором. Данный метод удобен скоростью поиска элемента, а также отсутствием пустых ячеек памяти.