МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«Дальневосточный федеральный университет»**

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра прикладной математики, механики, управления и программного обеспечения**

**О Т Ч Е Т**

по лабораторной работе №4 по дисциплине “Структуры и алгоритмы компьютерной обработки данных”

|  |
| --- |
| Выполнил студент гр. Б8204  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. А. Федоренко |
| Проверил к.т.н. доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Н. Остроухова |

г. Владивосток

2018

**Неформальная постановка задачи**

Разработать класс Hash-таблица.

**Спецификация класса**

struct Hash {

int hash\_key = -1; // Значение hash-функции у записи

unsigned short int status = 0; // статус каждой записи (0 – свободна, 1-занята,

2 - занята)

string FIO = "----------", // ФИО преподавателя

name\_of\_lesson = "--------", // Название предмета

class\_number = "----", // Номер аудитории

lesson\_number = "-"; // Номер пары

};

class HashTable

{

public:

HashTable(int n = 10, float k = 0.8); // Конструктор

int addRecord (Hash hash); // Функция добавления записи в таблицу

int searchRecord (Hash hash); // Функция поиска записи в таблице

int deleteRecord (Hash hash); // Функция удаления записи из таблицы

void print(); // Функция печати таблицы

~HashTable(); // Деструктор

private:

Hash\* hash\_table;

int SIZE; // Переменная, хранящая размер таблицы

int count = 0; // Счётчик записей в таблице

float RATIO; // Переменная, хранящая коэффициент заполненности

таблицы, при превышении которого, таблица должна

увеличиваться

int hash1 (string, string); // 1-я функция хеширования

int hash2 (int); // 2-я функция хеширования

void add(Hash, int); // вспомогательная функция добавления записи в

таблицу

int compare(Hash, int); // функция сравнения двух записей

void resizeTable(); // Функция увеличения размера таблицы

};

**Спецификация функций**

int **addRecord** (Hash **hash**) - Функция добавления записи в таблицу.

На вход принимает запись **hash** и возвращает целочисленное значение:

* -1, если запись **hash** уже содержится в таблице
* -2, если запись с таким же ключом, как и у **hash**, уже содержится в таблице
* 0, если добавление записи **hash** прошло успешно

int **searchRecord** (Hash **hash**) - Функция поиска записи в таблице.

На вход принимает запись **hash** и возвращает целочисленное значение:

* -1, если записи **hash** нет в таблице, либо она удалена
* Значение поля hash\_key в записи, которую нашла функция

int **deleteRecord** (Hash **hash**) - Функция удаления записи из таблицы.

На вход принимает запись **hash** и возвращает целочисленное значение:

* 1, если запись **hash** не содержится в таблице, либо она ужу удалена
* 0, если удаление записи **hash** из таблицы прошло успешно

void **print**() - Функция печати таблицы на экран.

На вход ничего не принимает и ничего не возвращает.

int **hash1** (string **s1**, string **s2**) – 1-я функция хеширования.

На вход принимает ключи **s1** и **s2** и возвращает целочисленное значение по правилу: коды символов **s1** и **s2** складываются, и это число берётся по модулю размера таблицы(SIZE)

int **hash2** (int **i**) – 2-я функция хеширования.

На вход принимает целочисленное значение **i**, полученное в результате хеширования, которое привело у коллизии,и возвращает целочисленное значение по формуле: (i+(SIZE-1))%SIZE

void **resizeTable**() - Функция увеличения размера таблицы.

Функция увеличивает размер таблицы, если заполненность таблицы превысила значение в переменной RATIO.

На вход ничего не принимается и ничего не возвращается.

**Тесты**

**addRecord** (Hash **hash**)**:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ситуация | Входные данные | Таблица до | Таблица после | Значение функции |
| Запись **hash** попадает в таблицу при первом хешировании | 2 F738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  - - ------ ------- --- -1 0  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1 | 0 |
| Запись **hash** попадает в таблицу при хешировании вторичной хеш-функцией 1 раз | 2 E737 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1  4 - ---- ------ ---- -1 0  5 - ---- ------ ---- -1 0  6 - ---- ------ ---- -1 0  7 - ---- ------ ---- -1 0  8 - ---- ------ ---- -1 0  9 - ---- ------ ---- -1 0 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1  4 - ---- ------ ---- -1 0  5 - ---- ------ ---- -1 0  6 - ---- ------ ---- -1 0  7 - ---- ------ ---- -1 0  8 - ---- ------ ---- -1 0  9 2 E737 bbbbb aaa 9 1 | 0 |
| Запись **hash** попадает в таблицу при хешировании вторичной хеш-функцией несколько раз | 2 G737 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1  4 - ---- ------ ---- -1 0  5 - ---- ------ ---- -1 0  6 - ---- ------ ---- -1 0  7 - ---- ------ ---- -1 0  8 - ---- ------ ---- -1 0  9 - ---- ------ ---- -1 0 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1  4 - ---- ------ ---- -1 0  5 - ---- ------ ---- -1 0  6 - ---- ------ ---- -1 0  7 - ---- ------ ---- -1 0  8 - ---- ------ ---- -1 0  9 2 G737 bbbbb aaa 9 1 | 0 |
| Запись **hash** уже содержится в таблице | 2 D738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | -1 |
| Совпадение ключей | 2 D738 aaaaa bbb | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | -2 |

**searchRecord** (Hash **hash**)**:**

|  |  |  |  |
| --- | --- | --- | --- |
| Ситуация | Входные данные | Таблица | Значение функции |
| Запись **hash** есть в таблице | 2 G738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 3 |
| Записи **hash** нет в таблице | 2 F738 asdfgh ccc | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | -1 |
| Записи **hash** удалена | 2 G738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 2 | -1 |

**deleteRecord** (Hash **hash):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ситуация | Входные данные | Таблица до | Таблица после | Значение функции |
| Запись **hash** есть в таблице | 2 F738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 2  3 2 G738 bbbbb aaa 3 1 | 0 |
| Записи **hash** нет в таблице | 2 F738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 1 |
| Запись **hash** уже удалена | 2 F738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 2  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 2  3 2 G738 bbbbb aaa 3 1 | 1 |