Отчет о проделанной работе

Дисциплина: Алгоритмы и структуры данных

Студент: Хабутдинов Р.А.

Группа: ПРО-201в

# Задание на лабораторную работу, содержащее используемые хеш-функцию, метод устранения коллизий

Построить хеш-таблицу, содержащую последовательность из m=45 элементов размерности n=2. Элементы генерируются с помощью датчика случайных чисел.

Хеш-функция - первая цифра квадрата ключа.

Метод разрешения конфликта - квадратичные пробы.

# Входные и выходные данные

## Входные данные

array – сгенерированная последовательность:

89, 22, 25, 63, 20, 61, 42, 14, 81, 53, 18, 26, 55, 12, 79, 62, 87, 69, 60, 46, 84, 88, 19, 33, 29, 10, 66, 28, 54, 73, 43, 91, 31, 97, 59, 50, 52, 92, 76, 47, 90, 65, 39, 44, 77

## Выходные данные

hashMap – хеш-таблица:

0-> -, 1->42, 2->14, 3->63, 4->22,

5->20, 6->25, 7->89, 8->87, 9->29,

10->81, 11->53, 12->61, 13->69, 14->73,

15->26, 16->84, 17->12, 18->46, 19->18,

20->66, 21->76, 22->79, 23->88, 24->91,

25->31, 26->33, 27->54, 28->55, 29->65,

30->77, 31-> -, 32->28, 33->92, 34->97,

35->59, 36-> -, 37->10, 38->50, 39->62,

40-> -, 41-> -, 42-> -, 43-> -, 44->90,

45-> -, 46-> -, 47-> -, 48-> -, 49-> -,

50->43, 51->52, 52->60, 53-> -, 54->44,

55-> -, 56-> -, 57-> -, 58-> -, 59-> -,

60-> -, 61-> -, 62-> -, 63-> -, 64-> -,

65->39, 66->47, 67->19

k zap: 0.6619718

sr prob: 5.787234

# Порядок построения хеш-таблицы

Создаем массив размером 45\*1.5 для хранения хеш-таблицы. В цикле находим для каждого элемента его хеш.

**Формула нахождения хеш-кода элемента:** Integer.parseInt(Integer.toString(el \* el).substring(0, 1))

Если по адресу равному вычисленному остатку от деления хеш-кода на размер хеш-таблицы, в массиве значения нет, то присваиваем туда значения текущего элемента.

Если по адресу уже есть значение, значит, прибавляем к хеш-коду элемента i\*i и находим остаток от деления на размер хеш-таблицы, где i – порядковый номер разрешения коллизии.

Если по вычисленному новому значению нет элемента, то записываем туда элемент, если есть повторяем предыдущий шаг до тех пор, пока не найдем пустой элемент в массиве хеш-таблицы.

# Формулы расчета требуемых параметров хеш-таблицы

## Коэффициент заполнения хеш-таблицы

**(**float**)**hashMap**.**size**()** **/** hashMapSize**;**

hashMap**.**size**()** // Кол-во заполненных элементов Хеш-таблицы

hashMapSize // Размер Хеш-таблицы

## Среднее число проб для размещения некоторого ключа

int sum **=** 0**;** // Сумма всех проб

probArray // Массив, который содержит кол-во проб при добавлении элемента в Хеш-таблицу

//Считаем сумму всех проб

**for** **(**Integer el **:** probArray**)** **{**

sum **+=** el**;**

**}**

//Сумму всех проб / кол-во элементов в Рандомном массиве

**(**float**)**sum **/** array**.**size**();** // Среднее число проб