Лабораторная работа №15
Тема: «Хеширование»

Цель работы: сформировать знания и умения по работе с подпрограммами, приобрести навыки написания программ с использованием хеш-функций.

Время выполнения: 4 часа.

Теоретические сведения

Хеш-функция – функция, преобразовывающая входную последовательность данных произвольного размера в выходную последовательность фиксированного размера. Процесс преобразования данных называется ***хешированием***. Результат хеширования – хеш-код (хеш-сумма, хеш).

Использовать хеш-функцию или нет, зависит от того насколько целесообразно применение ее свойств, а также свойств алгоритма, по которому она может быть реализована на ЯП. В одних ситуациях наиболее важна высокая скорость работы, в других равномерное распределение хеш-кодов и т. п. Далее будут рассмотрены два наиболее известных метода хеширования.

Метод деления

Пусть k – ключ (тот, что необходимо хешировать), а N – максимально возможное число хеш-кодов. Тогда метод хеширования посредством деления будет заключаться во взятии остатка от деления k на N:

h(k)=k mod N, где mod – операция взятия остатка от деления.

Например, на вход подаются следующие ключи:

3, 6, 7, 15, 32, 43, 99, 100, 133, 158.

Определим N равным 10, из чего последует, что возможные значения хешей лежат в диапазоне 0…9. Используя данную функцию, получим следующие значения хеш-кодов:

h(3)=3, h(6)=6, h(7)=7, h(15)= 5, h(32)=2, h(42)=2, h(99)=9, h(100)=0, h(133)=3, h(158)=8.

На C++ программу, выполняющую хеширование методом деления можно записать так:

```
#include "stdafx.h"
#include <iostream>
using namespace std;
int HashFunction(int k)
{
return (k%10);
}
void main()
{
setlocale(LC_ALL, "Rus");
int key;
cout<<"Ключ > "; cin>>key;
cout<<"HashFunction("<<key<<")="<<HashFunction(key)<<endl;
system("pause>>void");
}
```

Во избежание большого числа коллизий рекомендуется выбирать N простым числом, и не рекомендуется степенью с основанием 2 и показателем m (2m). Вообще, по возможности, следует выбирать N, опираясь на значения входящих ключей. Так, например если все или большинство k=10m (m – натуральное число), то неудачным выбором будет N=10*m и N=10m.

Метод умножения

Получить из исходной последовательности ключей последовательность хеш-кодов, используя метод умножения (мультипликативный метод), значит воспользоваться хеш-функцией:

$$h(k)=\lfloor N*(\{k*A\})\rfloor$$

Здесь A – рациональное число, по модулю меньшее единицы ($0<A<1$), а k и N обозначают то же, что и в предыдущем методе: ключ и размер хеш-таблицы. Также правая часть функции содержит три пары скобок:

( ) – скобки приоритета;
⌊ ⌋ – скобки взятия целой части;
{ } – скобки взятия дробной части.

Аргумент хеш-функции k ($k\geq 0$) в результате даст значение хеш-кода h(k)=x, лежащие в диапазоне 0…N-1. Для работы с отрицательными числами можно число x взять по модулю.

От выбора A и N зависит то, насколько оптимальным окажется хеширование умножением на определенной последовательности. Не имея сведений о входящих ключах, в качестве N следует выбрать одну из степеней двойки, т. к. умножение на 2m равносильно сдвигу на m разрядов, что компьютером производиться быстрее. Неплохим значением для A (в общем случае) будет (√5-1)/2≈0,6180339887. Оно основано на свойствах золотого сечения:

Золотое сечение – такое деление величины на две части, при котором отношение большей части к меньшей равно отношению всей величины к ее большей части.

Отношение большей части к меньшей, выраженное квадратичной иррациональностью:

$$\varphi=(\sqrt{5}+1)/2\approx1,6180339887$$

Для мультипликативной хеш-функции было приведено обратное отношение:

$$1/\varphi=(\sqrt{5}-1)/2\approx0,6180339887$$

При таком A, хеш-коды распределяться достаточно равномерно, но многое зависит от начальных значений ключей.

Для демонстрации работы мультипликативного метода, положим N=13, A=0,618033. В качестве ключей возьмем числа: 25, 44 и 97. Подставим их в функцию:

h(k)=⌊13*({25*0,618033})⌋=⌊13*{15,450825}⌋=⌊13*0,450825⌋=⌊5,860725⌋=5

h(k)=⌊13*({44*0,618033})⌋=⌊13*{27,193452}⌋=⌊13*0,193452⌋=⌊2,514876⌋=2

h(k)=⌊13*({97*0,618033})⌋=⌊13*{59,949201}⌋=⌊13*0,949201⌋=⌊12,339613⌋=12

Реализация метода на C++ с использованием оговоренных N и A:

```
#include "stdafx.h"
#include <iostream>
using namespace std;
int HashFunction(int k)
{
```

```
int N=13; double A=0.618033;
int h=N*fmod(k*A, 1);
return h;
}
void main()
{
setlocale(LC_ALL, "Rus");
int key;
cout<<"Ключ > "; cin>>key;
cout<<"HashFunction("<<key<<")="<<HashFunction(key)<<endl;
system("pause>>void");
}
```

Общее задание к лабораторной работе №15

Составить хеш-функцию в соответствии с заданным вариантом и проанализировать ее. При необходимости доработать хеш-функцию. Используя полученную хеш-функцию разработать на языке программирования C++ программу, которая должна выполнять следующие функции:

- создавать хеш-таблицу;
- добавлять элементы в хеш-таблицу;
- просматривать хеш-таблицу;
- искать элементы в хеш-таблице;
- удалять элементы из хеш-таблицы.

Описание хеш-функции

Хеш-функция основана на возведении суммы кодов символов ключа в квадрат и извлечение из полученного квадрата нескольких средних цифр. При этом коды символов умножаем на частное кода и произведения тройки на порядковый номер символа в ключе (1ч6). Звучит убого, вот так выглядит формула суммы:

где - код символа с индексом "i";

Возведенная в квадрат сумма колеблется от 7997584 до 22781529, а это семизначное или восьмизначное число. Для адресации сегментов хеш-таблицы необходимо четырехзначное число, не превышающее 2000. Откинем у квадрата суммы 2 первых и два последних разряда, так у нас получится трехзначное или четырехзначное число. Для того, чтобы адрес не превысил максимально допустимый адрес 1999, будем брать остаток от деления на 2000 до тех пор, пока он не попадет в нужный диапазон.

Экспериментальный анализ хеш-функции

Экспериментальное исследование проводится следующим образом:

формируются случайным образом ключи заданного формата в количестве, превышающем количество сегментов хеш-таблицы в 2…3 раза;

для каждого сформированного ключа вычисляется хеш-функция, и подсчитывается, сколько раз вычислялся адрес того или иного сегмента хеш-таблицы.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретическую часть лабораторной работы.

2. Реализовать индивидуальное задание по вариантам, представленные в теоретических сведениях, сделать скриншоты работающих программ. Написать комментарии.

3. Написать отчет, содержащий:

1. Титульный лист, на котором указывается:

а) полное наименование министерства образование и название учебного заведения;

б) название дисциплины;

в) номер практического занятия;

г) фамилия преподавателя, ведущего занятие;

д) фамилия, имя и номер группы студента;

е) год выполнения лабораторной работы.

2. Индивидуальное задание из раздела «Теоретические сведения» с кодом, комментариями и скриншотами работающих программ.

3. Построение блок-схем.