Лабораторная работа № 7

Хеширование

Цель работы: изучить принципов построения хеш-функций, обладающих равномерным распределением, исследовать статистические свойства хеш-функций, закрепить навыки структурного программирования.

Общие сведения

Известно, что использование ключей в качестве индекса в массиве обеспечивает высокую производительность операций поиска, добавления, удаления элементов, при этом предполагается, что ключи соответствуют диапазону индексов. Однако такими благоприятными свойствами обладают далеко не все ключи.

Хеширование – это способ хранения данных, при котором на основании значений ключей элементов вычисляются значения хеш-функции – индексы, которые используются для поиска, добавления, удаления элементов из хеш-таблиц (специализированных контейнеров, основанных на массивах или других аналогичных структурах).

Основной задачей хеш-функций (и соответственно алгоритмов, лежащих в их основе), является получение из входных значений ключей (в общем случае имеющих произвольное распределение и диапазон значений) «случайных» индексов, *равномерно* распределенных в заданном диапазоне значений.

Для оценки качества функций может использоваться статистическая оценка хи-квадрат:

$$\chi^2 = \frac{M}{N}\sum_{i=0}^{M-1}\left(F_i - \frac{N}{M}\right)^2,$$

где N – количество ключей, M – размер хеш-таблицы, F_i – количество ключей с хеш-значением i.

Если хеш-значения являются случайными, то значения этой функции статистистического распределения для $N > cM$ должно быть равно $M \pm \sqrt{M}$ с вероятностью $1 - 1/c$. Необходимо отметить, что требования оценки хи-квадрат являются достаточно жесткими, и не всегда удается добиться того, чтобы хеш-функция удовлетворяла этим требованиям.

Задание

1. Разработать и реализовать функцию, осуществляющую хеширование данных (тип данных определяется вариантом).
2. Разработать и реализовать функцию-генератор, осуществляющую формирование значений ключей в соответствии с заданным типом данных. Генерируемые ключи должны быть уникальны.
3. Исследовать статистические свойства разработанной хеш-функции при заданных размерах хеш-таблицы и количестве ключей.
4. Составить отчет, в котором привести листинг хеш-функции, гистограммы распределений индексов, формируемых хеш-функцией (для двух значений размера хеш-таблицы) и выводы по работе (дать оценку зависимости от размера таблицы и от природы исходных данных – если таковые имеются; оценить качество разработанной хеш-функции).

Таблица 1.

Варианты заданий

№	Тип данных	Размеры хеш-таблицы и количество ключей	Примечание

1	`struct Date {`
	`int Day;`
	`int Month;`
	`int Year;};`
		$M_1 = 256$
		$M_2 = 257$
		$K = 2000$
			Диапазон изменения **Year** равен [1920; 2008]. Величина распределена по нормальному закону с максимумом в 1970.

2	`struct AutoNumber {`
	`char Letter[3];`
	`int  Number; };`
		$M_1 = 512$
		$M_2 = 511$
		$K = 10000$
			В качестве значений массива используются буквы русского алфавита в нижнем регистре, кроме **й**, **ы**, **ъ**, **ь**.[1]

3	`struct Address {`
	`int Building;`
	`int Apartment; };`
		$M_1 = 64$
		$M_2 = 67$
		$K = 500$
			Диапазоны изменения **Building** и **Apartment** равны соответственно [1; 10] и [1; 80].

4	`struct Date {`
	`int Month;`
	`int Year; };`
		$M_1 = 64$
		$M_2 = 67$
		$K = 200$
			Диапазон изменения **Year** равен [1972; 1992].

5	`struct Address {`
	`char * Street;`
	`int Building; };`
		$M_1 = 512$
		$M_2 = 511$
		$K = 2000$
			В качестве значений поля **Street** использовать названия улиц города Сургута (см. приложение А).

6	`struct Card {`
	`int Suit;`
	`int Value; };`
		$M_1 = 16$
		$M_2 = 17$
		$K = 50$
			Диапазоны изменения **Suit** и **Value** равны соответственно [0; 3] и [0; 12].

7	`struct Person {` `char * Surname; };`	$M_1 = 512$ $M_2 = 511$ $K = 2000$	В качестве значений **Surname** использовать фамилии, приведенные в приложении Б.

8	`struct Book {`
	`int Year;`
	`int Pages; };`
		$M_1 = 1024$
		$M_2 = 1031$
		$K = 5000$
			Диапазоны изменения **Year** и **Pages** равны соответственно [2000; 2008] и [100; 600].

9	`struct Rect {`
	`double Width;`
	`double Height};`
		$M_1 = 128$
		$M_2 = 127$
		$K = 300$
			Поля **Width** и **Height** могут принимать значения из ряда 0,2; 0,4; … 4,0.

10	`struct Chess {`
	`char Letter;`
	`int Digit};`
		$M_1 = 16$
		$M_2 = 17$
		$K = 50$	

[1] В этом и других вариантах, в случае, если вероятность значений поля не оговаривается, считать, что значения из указанного (или предполагаемого) диапазона равновероятны.