

Титульный лист материалов по дисциплине

ДИСЦИПЛИНА **Структуры и алгоритмы обработки данных (ч. 2)**

(полное наименование дисциплины без сокращений)

ИНСТИТУТ **ИТ**

КАФЕДРА **Математического обеспечения и стандартизации
информационных технологий**

(полное наименование кафедры)

ВИД УЧЕБНОГО **Практические работы**

МАТЕРИАЛА (в соответствии с пп.1-11)

ПРЕПОДАВАТЕЛЬ **Муравьёва Екатерина Андреевна**

(фамилия, имя, отчество)

СЕМЕСТР **3 семестр, 2023-2024 уч. год**

(указать семестр обучения, учебный год)

Практическая работа 3

Тема: применение хеш-таблицы для поиска данных в двоичном файле с записями фиксированной длины

Цель: получить навыки по разработке хеш-таблиц и их применении при поиске данных в других структурах данных (файлах).

Хеширование для достижения константного времени доступа к записи в таблице

Хеширование как преобразование исходных данных в выходную битовую строку находит применение в таких сферах, как контроль целостности при передаче данных (контрольные суммы), информационная безопасность (защита паролей, ЭЦП) и некоторые другие.

В том числе хеширование может быть использовано и для организации эффективного (с константным временем $O(1)$) поиска (также вставки и удаления) элементов данных в **динамическом множестве**.

Хеш-функция при этом создаёт отображение множества ключевых значений во множество индексов соответствующих записей данных в массиве в виде вспомогательной **хеш-таблицы** (рис. 1).

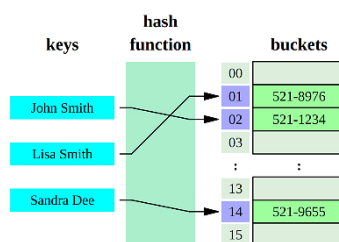


Рис. 1. Индексы элементов динамического множества данных как результат хеширования значения ключевых полей элементов полезных данных

В этом случае при вводе ключа поиска программа вычислит хеш и затем по хеш-таблице определит индекс искомой записи в массиве полезных данных, что открывает к ней прямой доступ.

Алгоритм хеш-функции может быть основан на делении (модальная арифметика, полиномиальный хеш), умножении (хеширование Фибоначчи), на подходе под названием «универсальное хеширование», а также некоторых других.

Например, для алгоритма, основанного на делении, хеш-функция может быть реализована на основе модальной арифметики:

$$h = K \bmod Q, \quad (1)$$

где K – ключевое значение, Q – наибольшее необходимое количество различных значений хеш-функции (и, как следствие, допустимое количество записей в динамическом множестве).

Если K – составное значение (например, строка символов), то его можно представить в виде полинома.

Примечание: в рамках данной практической работы целесообразно использовать алгоритмы, основанные на делении.

Одним из свойств хеш-функции является необязательность уникальности значений хеша для различных входных наборов данных. Это объясняет ненулевую вероятность возникновения **коллизии** – ситуации, когда по разным ключевым значениям может быть вычислено одинаковое хеш-значение. Таким образом, двум или более наборам данных может быть сопоставлен одинаковый индекс в массиве — а это недопустимо.

Для устранения (разрешения, преодоления) коллизии можно использовать методы **цепного хеширования** и **хеш с открытой адресацией**.

Цепным хешированием называется способ разрешения коллизий, когда динамическое множество полезных данных организуется в виде массива **линейных списков**, состоящих из элементов с одинаковыми хеш-значениями, т.е. индексами в массиве (рис. 2).

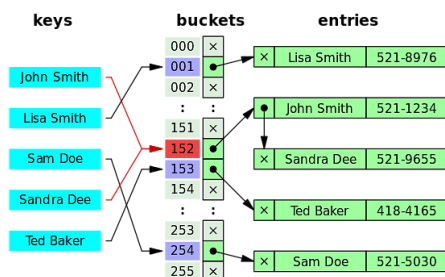


Рис. 2. Схема организации цепного хеширования

При этом в хеш-таблице ключам сопоставляются индексы головных элементов этих списков в массиве.

Массив списков может стать на некотором этапе работы программы неоднородным – несколько длинных списков и множество пустых элементов массива. С одной стороны, массив, даже пустой, занимает память. С другой стороны, время доступа к данным в списке линейное, а не константное, т.е. налицо снижение эффективности поиска.

На практике создают сначала небольшой массив, а по мере заполнения элементами перестраивают его, т.е. увеличивают размер с **рехешированием** (пересчетом хешей с новым значением Q).

Критерием необходимости перестройки массива является соотношение n/m – **коэффициент нагрузки**, где n – это количество уже имеющихся записей, m – длина массива. При достижении значения этого коэффициента $0,75+$, следует увеличить длину массива вдвое. Это гарантирует, что длины списков будут относительно небольшими.

Другой способ преодоления коллизий – хеширование с открытой адресацией (рис. 3). Если в массиве в строке с определённым индексом записи нет, то адрес открыт и в соответствующую строку можно поместить новый элемент. Иначе – адрес закрыт (коллизия) и необходимо по некоему алгоритму осуществить **последовательность проб** – сместиться относительно закрытого адреса в поисках открытого. Все базовые операции (поиск, вставка, удаление элемента) так или иначе задействуют пробирование, но у каждой свои нюансы.

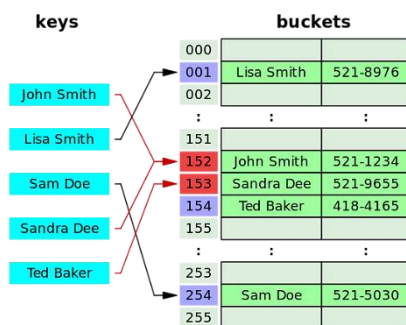


Рис. 3. Пример заполнения массива на основе открытой адресации

Распространённые схемы пробирования: линейное, квадратичное пробирование, двойное хеширование.

В наиболее простой схеме **линейного** пробирования смещение относительно адреса коллизии кратно целочисленной константе (эту константу следует задать так, чтобы они с длиной массива были взаимно просты):

$$\text{адрес} = h(x) + ci \quad (2)$$

где i – номер попытки разрешить коллизию; c – константа, определяющая шаг перебора.

В **квадратичной схеме** шаг перебора сегментов нелинейно зависит от номера попытки найти свободный сегмент:

$$\text{адрес} = h(x) + ci + di^2 \quad (3)$$

где i – номер попытки разрешить коллизию, c и d – константы.

В схеме **двойного хеширования** смещение относительно закрытого адреса кратно величине второй хеш-функции, схожей, но не эквивалентной основной:

$$\text{адрес} = h(x) + ih_2(x) \quad (4)$$

В случае открытой адресации имеет смысл создать массив сразу наибольшей длины. В противном случае при постепенном заполнении массива записями будет всё более длительной процедура поиска открытого адреса. Затраты времени на перестройку этого массива лишь снизят эффективность всей программы.

Задание 1

Ответьте на вопросы:

1. Расскажите о назначении хеш-функции.
2. Что такое коллизия?
3. Что такое «открытый адрес» по отношению к хеш-таблице?
4. Как в хеш-таблице с открытым адресом реализуется коллизия?
5. Какая проблема, может возникнуть после удаления элемента из хеш-таблицы с открытым адресом и как ее устранить?
6. Что определяет коэффициент нагрузки в хеш-таблице?
7. Что такое «первичный кластер» в таблице с открытым адресом?
8. Как реализуется двойное хеширование?

Задание 2

Разработать приложение, которое использует хеш-таблицу для организации прямого доступа к записям двоичного файла. Метод разрешения коллизии представлен в вашем варианте задания в таблице 1.

Для обеспечения прямого доступа к записи в файле элемент хеш-таблицы должен включать обязательные поля: ключ записи в файле, номер записи с этим ключом в файле. Элемент может содержать другие поля, требующиеся методу (указанному в вашем варианте), разрешающему коллизию.

1. Управление хеш-таблицей.
 - 1) Определить структуру элемента хеш-таблицы и структуру хеш-таблицы в соответствии с методом разрешения коллизии, указанным в варианте.
 - 2) Разработать хеш-функцию (метод определить самостоятельно), выполнить ее тестирование, убедиться, что хеш (индекс элемента таблицы) формируется верно.
 - 3) Разработать операции: вставить ключ в таблицу, удалить ключ из таблицы, найти ключ в таблице, рехешировать таблицу. Каждую операцию тестируйте по мере ее реализации.

- 4) Подготовить тесты (последовательность значений ключей), обеспечивающие:
 - вставку ключа без коллизии
 - вставку ключа и разрешение коллизии
 - вставку ключа с последующим рехешированием
 - удаление ключа из таблицы
 - поиск ключа в таблице

Примечание. Для метода с открытым адресом подготовить тест для поиска ключа, который размещен в таблице после удаленного ключа, с одним значением хеша для этих ключей.

- 5) Выполнить тестирование операций управления хеш-таблицей. При тестировании операции вставки ключа в таблицу предусмотрите вывод списка индексов, которые формируются при вставке элементов в таблицу.

Задание 3

Управление бинарным файлом посредством хеш-таблицы.

В заголовочный файл подключить заголовочные файлы: управления хеш-таблицей, управления двоичным файлом. Реализовать поочередно все перечисленные ниже операции в этом заголовочном файле, выполняя их тестирование из функции main приложения. После разработки всех операций выполнить их комплексное тестирование (программы (все базовые операции, изменение размера и рехеширование), тест-примеры определите самостоятельно. Результаты тестирования включите в отчет по выполненной работе).

Разработать и реализовать операции.

- 1) Прочитать запись из файла и вставить элемент в таблицу (элемент включает: ключ и номер записи с этим ключом в файле, и для метода с открытой адресацией возможны дополнительные поля).
- 2) Удалить запись из таблицы при заданном значении ключа и соответственно из файла.
- 3) Найти запись в файле по значению ключа (найти ключ в хеш-таблице, получить номер записи с этим ключом в файле, выполнить прямой доступ к записи по ее номеру).
- 4) Подготовить тесты для тестирования приложения:
Заполните файл небольшим количеством записей.
 - Включите в файл записи как не приводящие к коллизиям, так и приводящие.
 - Обеспечьте включение в файл такого количества записей, чтобы потребовалось рехеширование.

Заполните файл большим количеством записей (до 1 000 000).

Определите время чтения записи с заданным ключом: для первой записи файла, для последней и где-то в середине. Убедитесь (или нет), что время доступа для всех записей одинаково.

Составить отчет.

Таблица 1. Варианты заданий к практической работе

№	Тип хеш-таблицы (метод разрешения коллизии)	Структура записи двоичного файла
1	С открытой адресацией (смещение на 1)	Читательский абонемент: номер читательского билета – целое пятизначное число, ФИО, адрес.
2	С открытой адресацией (смещение на номер выполняемого подбора)	Счет в банке: номер счета – 7 разрядное число, ФИО, адрес.
3	С открытой адресацией (двойное хеширование)	Владелец телефона: номер телефона – последовательность символов, адрес, ФИО.
4	Цепное хеширование	Владелец автомобиля: номер машины, марка, сведения о владельце, сведения об угоне (логического типа).
5	Цепное хеширование	Пациент поликлиники: номер карточки, код хронического заболевания, фамилия лечащего врача.
6	Цепное хеширование	Товар: название, код – шестизначное число, завод изготовитель, цена, страна (название).
7	Цепное хеширование	Специализация вуза: код специальности, название вуза, название специальности.
8	Открытый адрес(двойное хеширование)	Книга: ISBN – двенадцатизначное число, автор, название, год издания.
9	Цепное хеширование	Страховой полис: номер, компания, фамилия владельца.
10	Открытый адрес(смещение на 1)	Англо-русский словарь: английское слово, русское слово.

11	Открытый адрес(двойное хеширование)	Железнодорожная справка: номер поезда, пункт отправления, пункт назначения, время отправления.
12	Цепное хеширование	Регистрация малого предприятия: номер лицензии (текстовое значение), название, учредитель, признак действия лицензии (0 действует, 1 отозвана).
13	Открытый адрес(двойное хеширование)	Студент: номер зачетной книжки, номер группы, ФИО.
14	Цепное хеширование	Справочная межгорода: код города, название города, страна.
15	Открытый адрес (смещение на 1)	Найти и поздравить друга: <u>дата рождения</u> , имя
16	Цепное хеширование	Расписание занятий группы: номер группы, название дисциплины, номер пары, номер недели, номер дня недели, вид занятия, номер аудитории.
17	Открытый адрес (смещение на 1)	Частотный словарь: слово, количество вхождений в текст.
18	Открытый адрес (двойное хеширование)	Читательский билет: номер, инвентарный номер книги, дата выдачи, дата возврата.
19	Цепное хеширование	Вызов такси: номер, фамилия водителя, время выезда, отметка о присутствии в гараже.
20	Открытый адрес(смещение на 1)	Продажи товаров: код товара, название, цена, дата продажи.
21	Открытый адрес(двойное хеширование)	Сотрудник: табельный номер, должность, оклад, количество детей.
22	Цепное хеширование	Расписание занятий группы: номер группы, название дисциплины, номер пары, номер недели, номер дня недели, вид занятия, номер аудитории.
23	Открытый адрес(двойное хеширование)	Нагрузка по дисциплине: <u>код</u>

		<u>дисциплины</u> , <u>код направления подготовки</u> , название дисциплины, номер семестра проведения дисциплины.
24	Цепное хеширование	Нагрузка по дисциплине: <u>код дисциплины</u> , <u>код направления подготовки</u> , название дисциплины, номера семестров проведения дисциплины (не более двух). <u>Подсказка</u> . Если только один семестр, то второе поле должно содержать 0.
25	Открытый адрес (смещение на 1)	Аэропорт (табло прибытия пассажирских авиарейсов сохраняет все данные в файле): пункт вылета, <u>номер рейса</u> , дата прилета, время прилета, информация о задержке прилета в часах.
26	Цепное хеширование	Учет заболеваний пациента. Структура записи о пациенте: <u>номер полиса</u> , фамилия, имя, отчество, код заболевания, дата установки диагноза, код врача.
27	Открытый адрес (смещение на 1)	Учет техосмотра автомобилей. Структура записи об автомобиле: <u>Номер</u> (код региона, цифровой код, буквенный код), Модель, Цвет, Сведения о владельце (Фамилия, Имя, Адрес), дата последнего техосмотра.
28	Открытый адрес (смещение на 1)	Учет нарушений ПДД. Структура записи о нарушении ПДД: <u>номер автомобиля</u> , фамилия и инициалы владельца, модель, дата нарушения, место нарушения (текстом), статья (КоАП), наказание (сумма штрафа).
29	Открытый адрес (двойное хеширование)	Справочник банков по городам страны. Об отдельном банке хранятся данные: наименование, <u>код банка</u> , адрес (город), форма

		собственности (коммерческий или государственный).
30	Цепное хеширование	Касса магазина. Структура записи операции по кассе: <u>номер кассы</u> , код товара, количество товара, цена товара, процентная скидка на товар, сумма за товар с учетом скидки.
31	Открытый адрес (смещение на 1)	Киноафиша города. Структура записи о сеансе: название кинотеатра, название фильма, дата, время начала, стоимость билета.

Содержание отчёта:

1. Титульный лист.
2. Цель работы.
3. Ход работы (по каждому заданию):
 - a. Формулировка задачи.
 - b. Описание алгоритмов операций (вставка ключа в таблицу, поиск записи по ключу в таблице и возвращение номера записи в файле, удаление элементы из хеш-таблицы)
 - c. Код программы с комментариями.
 - d. Результаты тестирования (до 1 000 000). Обязательно скриншоты результатов выполнения операций с хеш-таблицей.
4. Вывод (решены ли задачи, достигнута ли цель, выводы о полученных результатах).

Для сдачи практической работы потребуется:

- отчёт – оформляется в виде электронного документа в форматах Word или PDF, прикрепляется к соответствующему заданию в СДО;
- программные проекты, реализованные по заданиям;
- доклад по результатам выполнения практической работы (по отчёту).