

ДИСЦИПЛИНА	Структуры и алгоритмы обработки данных <small>(полное наименование дисциплины без сокращений)</small>
ИНСТИТУТ	Информационных технологий
КАФЕДРА	Математического обеспечения и стандартизации информационных технологий <small>полное наименование кафедры)</small>
ВИД УЧЕБНОГО МАТЕРИАЛА	Практические занятия <small>(в соответствии с пп.1-11)</small>
ПРЕПОДАВАТЕЛЬ	Бузыкова Юлия Сергеевна <small>(фамилия, имя, отчество)</small>
СЕМЕСТР	3 семестр, 2023-2024 уч. год <small>(указать семестр обучения, учебный год)</small>

Оглавление

1	Практика №1-2 Разработка и программная реализация задач, использующих внешние структуры данных - файлы (текстовые и двоичные).	3
2	Практика №3-4 . Разработка и программная реализация задач поиска данных в таблицах с применением механизма хеширования.	17
3	Практика № 5-6 Нелинейные структуры данных. Бинарное дерево.....	26
4	Практика №7-9 Разработка и программная реализация структуры данных – Бинарное дерево поиска. Разработка программ, использующих созданную структуру.	35
5	Практика 10-11 Разбор задач с применением: метода динамического программирования (задача о камнях, треугольник), метод решета	41
6	Практика №12-14 Разработка и программная реализация класса Граф: создание, алгоритмы обхода. Исследование алгоритмов Прима и Крускала построения минимального покрывающего дерева. Топологическая сортировка.	48
7	Практика № 15-16 Подробное рассмотрение реализации алгоритма Хаффмана. Упражнения по выполнению сжатия данных методом группового кодирования RLE, Шеннона-Фано, Лампеля-Зева	54

13 Практика №1-2 Разработка и программная реализация задач, использующих внешние структуры данных - файлы (текстовые и двоичные).

Тема. Внешние структуры данных: текстовый и двоичный файлы.

Цель. Получить навыки применения файловых потоков языка C++ (или файлов языка Си) по управлению текстовым и двоичным файлами.

Задание 1. Разработать программу, управления текстовым файлом **Требования**

1. Реализация ввода-вывода на основе файловых потоков C++: `ofstream`, `ifstream`.
2. Имя физического файла вводится пользователем и передается в функции обработки через параметр.
3. При открытии файла выполнять контроль его существования и открытия.

Примечание. При отладке программы можете имя физического файла определить через константу.

4. Разработать функции для выполнения операций над текстовым файлом.
 - 1) создание текстового файла средствами текстового редактора кодировки ASCII, содержащего десятичные числа по несколько чисел на строке;
 - 2) вывод содержимого текстового файла;
 - 3) добавление новой записи в конец файла;
 - 4) прочесть значение числа, указав его порядковый номер в файле, и вернуть его значение;
 - 5) определить количество чисел в файле.
5. Разработать приложение и выполнить тестирование всех функций. Приложение должно содержать диалоговый интерфейс на основе текстового меню.
6. Контроль открытия и существования файла выполнить в основной программе перед вызовом функции. Перед закрытием файла, проверить отсутствие ошибок ввода и вывода (метод `good`)/
7. Создать модуль и перенести в него все отлаженные функции. Исключить функции из приложения. Отладить приложение, подключив к нему модуль с функциями.
8. Разработать функции для реализации дополнительных операций,

4 определенных вариантом и сохранить их в модуле с остальными функциями.

9. Выполнить тестирование приложения в полном объеме.

Варианты к заданию 1

№	Дополнительные операции
1	Удаление значения строки с заданным номером, путем создания нового файла, удаление старого и переименование нового, указав ему имя удаленного файла.
2	Скопировать числа исходного файла, которые кратны 7, в новый файл.
3	Добавить в файл новую строку перед первой строкой. Путем создания нового файла.
4	Создать новый файл из значений исходного, записав в него числа, полученные путем деления каждого числа исходного файла на минимальное число, хранящееся в исходном файле.
5	Создать новый файл из значений исходного, умножив каждое число на значение последнего элемента исходного файла.
6	Создать новый файл из значений исходного, умножив каждое число на сумму первого и последнего чисел исходного файла.
7	Создать новый файл из значений исходного, поделив каждое число на число, равное количеству строк в файле.
8	Создать новый файл из значений исходного, поменяв местами первое число и последнее.
9	Создать новый файл из значений исходного, заменив все отрицательные числа файла, на квадрат минимального числа исходного файла.
10	Создать новый файл из значений исходного, умножив каждое четное число на максимальное число в файле.
11	Создать новый файл из значений исходного, разделив все нечетные по абсолютной величине числа на среднее арифметическое чисел в файле.
12	Создать новый файл из значений исходного, сформировав число по правилу: прибавить к каждому числу первое четное число исходного

5	файла.
13	Создать новый файл из значений исходного, поделив каждое число на наибольший общий делитель чисел файла.
14	Создать новый файл из значений исходного, добавив к каждому числу наибольшее и наименьшее из чисел файла исходного файла.
15	Создать новый файл из значений исходного, увеличив его на среднее арифметическое всех отрицательных чисел исходного файла.
16	Создать новый файл из значений исходного, размещая в одной строке исходное число и через пробел количество цифр в числе.
17	Создать новый файл из значений исходного, размещая в каждой строке три числа: исходное, количество цифр в числе, сумма цифр в числе.
18	Создать новый файл из значений исходного, размещая на строке по три числа из исходного файла.
19	Создать новый файл из значений исходного, размещая на строке значения: количество чисел до конца строки, сами числа через пробел.
20	Создать новый файл из значений исходного, размещая на строке значения: четное число – количество чисел до конца строки, сами числа через пробел.
21	Создать новый файл из значений исходного, разбив его данные на две примерно равные (возможно с разницей на одно значение) по длине части. Перед каждой частью указать число, равное количеству чисел в этой части. Каждая часть должна представлять одну строку текстового файла.
22	Создать новый файл из значений исходного, размещая на каждой строке заданное количество чисел, кроме, возможно, последней, где должны разместиться оставшиеся числа.
23	Создать новый файл из значений исходного, располагая на каждой строке количество чисел, который сформирует датчик случайных чисел при заданном диапазоне (от количества записей в исходном файле).
24	Создать новый файл из значений исходного. Значениями строк нового файла будут суммы каждого десятка чисел исходного, с указанием

6	перед суммой ее порядкового номера.
25	Создать новый файл из значений исходного, переписав в него только простые числа, располагая каждое на отдельной строке текстового файла.
26	Создать новый файл из значений исходного, по правилу: третья часть содержимого исходного файла должна стать первой частью нового файла, а две первых исходного должны расположиться после первой в новом файле.
27	Создать новый файл из значений исходного, представить в нем на двух отдельных строках значения для заполнения двух массивов: на первой строке нечетные числа, а на второй четные. В начале каждой строки указать количество чисел в строке.
28	Создать два новых файла из значений исходного, переписав в один из них первую половину чисел исходного, а в другой, оставшуюся часть. В исходный файл слить данные их двух новых файлов упорядоченными по возрастанию парами, т.е. прочитать первые числа двух файлов, сначала в исходный файл записать меньшее из них, а за ним большее.
29	Создать новый файл из значений исходного, размещая на каждой строке нового файла количество чисел в соответствии с числами последовательности Фибоначчи.
30	Создать новый файл из значений исходного, размещая в нем на строках количество чисел достаточное для заполнения квадратной матрицы заданного размера. В начале нового файла, на отдельной строке, указать размер матрицы одним числом. Протестировать заполнение матрицы данными нового файла.

Задание 2. Разработать программу управление двоичными файлами с записями фиксированной длины

Общие требования: файл состоит из записей определенной структуры, согласно варианту. Записи имеют ключ, уникальный в пределах файла.

Требования к подготовке и выполнению задания

1. Разработать структуру записи двоичного файла согласно варианту

7 задания.

2. Подготовить тестовые данные в текстовом файле с кодировкой ASCII, в соответствии со структурой записи варианта. При открытии файла выполнить контроль его существования и открытия.

Примечание. Реализация операций по чтению данных из файла будет проще, если значение для каждого поля записи размещать на отдельной строке текстового редактора.

3. Имя файла вводит пользователь.
4. При открытии файла обеспечить контроль существования и открытия файла.
5. При применении механизма прямого доступа к записи файла выполнить контроль присутствия записи с заданным номером в файле.
6. Разработать функции для выполнения операций:
 - преобразование тестовых данных из текстового файла в двоичный файл;
 - сохранение данных двоичного файла в текстовом, так, чтобы используя их можно было восстановить двоичный файл;
 - вывод всех записей двоичного файла;
 - доступ к записи по ее порядковому номеру в файле, используя механизм прямого доступа к записи в двоичном файле;
 - удаление записи с заданным значением ключа, выполнить путем замены на последнюю запись.
 - манипулирование записями в двоичном файле согласно дополнительным операциям, определенным в варианте;
7. Сохраните функции в новом модуле.
8. Разработать приложение, демонстрирующее выполнение всех операций, подключив к нему модуль с функциями.
9. Выполнить тестирование приложения, продемонстрировав выполнение всех операций.

Варианты к заданию 2

№	Задание	
1	Структура записи	Читательский абонемент: номер читательского билета – целое пятизначное число, ФИО, адрес.
	Доп. операция	1. Поиск записи с заданным значением ключа. 2. Удаление найденного значения.

8	2	Структура записи	Счет в банке: номер счета – 7 разрядное число, ФИО, адрес.
		Доп. операция	1. Поиск записи с заданным значением ключа. 2. Обновить значение одного поля.
	3	Структура записи	Владелец телефона: номер телефона – последовательность символов, адрес, ФИО.
		Доп. операция	1. Сформировать текстовый файл из фамилий владельцев, чьи номера начинаются с введенных первых трех цифр (например, 434). 2. Удалить сведения о владельцах телефонов, которые начинаются с заданной цифры.
	4	Структура записи	Владелец автомобиля: номер машины, марка, сведения о владельце, сведения об угоне (логического типа).
		Доп. операция	1. Список автомобилей, числящихся в угоне. 2. Установить факт угона автомобиля с заданным номером.
	5	Структура записи	Пациент поликлиники: номер карточки, код хронического заболевания, фамилия лечащего врача.
		Доп. операция	1. Сформировать двоичный файл с записями о пациентах с заданным кодом заболевания. 2. Заменить фамилию, имя, отчество врача у указанных пациентов (список пациентов – это массив номеров карточек).
	6	Структура записи	Товар: название, код – шестизначное число, завод изготовитель, цена, страна (название).
		Доп. операция	1. Сформировать текстовый файл о поставках заданной страны, размещая на строке все сведения о товаре, кроме страны. Название страны записать в первой строке файла. 2. Добавить новую запись о товаре в конец файла.
	7	Структура записи	Специализация вуза: код специальности, название вуза, название специальности.
		Доп. операция	1. Сформировать список вузов, которые ведут

9		<p>подготовку специалистов по специальности с указанным номером. Список представить в текстовом файле.</p> <p>2. Изменить код специальности по названию специальности.</p>
8	Структура записи	Книга: ISBN – двенадцатизначное число, автор, название, год издания.
	Доп. операция	<p>1. Создать новый типизированный файл и перенести в него записи о книгах указанного автора за указанный год.</p> <p>2. Удалить книги, изданные в указанном году.</p>
9	Структура записи	Страховой полис: номер, компания, фамилия владельца.
	Доп. операция	<p>1. Сформировать список клиентов заданной страховой компании. Список сохранить в текстовом файле, где сведения об одном клиенте в одной строке.</p> <p>2. Изменить название компании у всех клиентов. Старое и новое названия известны.</p>
10	Структура записи	Англо-русский словарь: английское слово, русское слово.
	Доп. операция	<p>1. Сформировать список английских слов, начинающихся с указанной буквы и их русский перевод.</p> <p>2. Обновить запись, записав новый вариант русского перевода по заданному слову.</p>
11	Структура записи	Железнодорожная справка: номер поезда, пункт отправления, пункт назначения, время отправления.
	Доп. операция	<p>1. Сформировать справку по поездам, отправляющимся в указанный пункт назначения. Результаты записать в новый двоичный файл такой же структуры, как и исходный.</p> <p>2. Удалить сведения по указанному поезду.</p>
12	Структура записи	Регистрация малого предприятия: номер лицензии (текстовое значение), название, учредитель, признак

1		действия лицензии (0 действует, 1 отозвана).
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать список лицензий одного заданного учредителя. Записать полученные данные в текстовый файл, располагая записи построчно. 2. Отозвать указанные лицензии. Номера отзывааемых лицензий находятся в текстовом файле, каждый номер на отдельной строке.
13	Структура записи	Студент: номер зачетной книжки, номер группы, ФИО.
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать список студентов одной группы в двоичном файле той же структуры, что и исходный. 2. Перевести некоторых студентов в другую группу. Номера зачетных книжек переводимых студентов находятся в текстовом файле, каждый номер на отдельной строке. Номер новой группы известен и передается в операцию.
14	Структура записи	Справочная межгорода: код города, название города, страна.
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать в текстовом файле таблицу кодов указанной страны. В строке файла должны храниться: название города и код. 2. Обновить код города по названию города.
15	Структура записи	Найти и поздравить друга: <u>дата рождения</u> , имя
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать в текстовом файле список друзей, родившихся в указанном месяце. 2. Указать имя, родившегося в заданную дату.
16	Структура записи	Расписание занятий группы: номер группы, название дисциплины, номер пары, номер недели, номер дня недели, вид занятия, номер аудитории.
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать в двоичном файле расписание заданной группы на заданный день недели. 2. Обновить расписание, найти дисциплины, которые стоят на одинаковых парах в одной

1		аудитории, и определить для них новые аудитории.
17	Структура записи	Частотный словарь: слово, количество вхождений в текст.
	Доп. операция	<ol style="list-style-type: none"> 1. Определить, какое слово встречалось чаще всего в тексте. 2. Добавить в файл новую запись по слову. 3. Обновить количество вхождений некоторых слов, увеличив их количество на 1.
18	Структура записи	Читательский билет: номер, инвентарный номер книги, дата выдачи, дата возврата.
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать список читателей, которые не вернули книги в срок (дата возврата < текущей), полученные данные записать в двоичный файл. 2. Найти запись по заданным критериям: номер, инвентарный номер и вернуть указатель на запись. 3. Удалить запись о книге, которую читатель вернул в библиотеку.
19	Структура записи	Вызов такси: номер, фамилия водителя, время выезда, отметка о присутствии в гараже.
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать список машин, находящихся в гараже. Полученные данные записать в двоичный файл с той же структурой. 2. Обновить сведения о машине при выезде из гаража.
20	Структура записи	Продажи товаров: код товара, название, цена, дата продажи.
	Доп. операция	<ol style="list-style-type: none"> 1. Посчитать сумму, на которую были проданы товары за указанную дату. 2. Сформировать двоичный файл на основе записей исходного файла, содержащий сведения о товаре: код товара, количество продаж, цена товара. 3. Произвести расчет выручки от продаж на основе данных нового файла.

1	20	Структура записи	Сотрудник: табельный номер, должность, оклад, количество детей.
		Доп. операция	1. Увеличить оклад вдвое тех сотрудников, которые занимают указанные в текстовом файле должности. 2. Удалить сотрудников с табельными номерами, которые хранятся в двоичном файле.
	21	Структура записи	Расписание занятий группы: номер группы, название дисциплины, номер пары, номер недели, номер дня недели, вид занятия, номер аудитории.
		Доп. операция	1. Сформировать в двоичном файле расписание заданной группы на заданный день недели. 2. Обновить расписание, найти дисциплины, которые стоят на одинаковых парах в одной аудитории, и определить для них новые аудитории.
	22	Структура записи	Нагрузка по дисциплине: <u>код дисциплины, код направления подготовки</u> , название дисциплины, номер семестра проведения дисциплины.
		Доп. операция	1. Сформировать список дисциплин, читаемых заданному направлению в заданном семестре. 2. Заменить в записи по заданной дисциплине заданного семестра заданного направления название дисциплины.
	23	Структура записи	Нагрузка по дисциплине: <u>код дисциплины, 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.
		Доп. операция	1. Сформировать список кодов направлений, в учебном плане которых содержится заданная дисциплина и читается она два семестра. 2. Поменять местами значения двух заданных номерами записей в файле.
	24	Структура записи	Аэропорт (табло прибытия пассажирских авиарейсов сохраняет все данные в файле): пункт вылета, <u>номер</u>

1		рейса, дата прилета, время прилета, информация о задержке прилета в часах.
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать в новом двоичном файле информацию по рейсам, задержанным в указанную дату. 2. Изменить время прилета для заданного рейса на новое значение.
25	Структура записи	Учет заболеваний пациента. Структура записи о пациенте: <u>номер полиса</u> , фамилия, имя, отчество, код заболевания, дата установки диагноза, код врача.
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать список пациентов, которым поставлен диагноз с заданным кодом заболевания в новом двоичном файле. 2. Удалить сведения о пациенте с заданным ключом, сохраняя порядок следования остальных записей.
26	Структура записи	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.
	Доп. операция	<ol style="list-style-type: none"> 1. Заменить дату техосмотра у автомобиля с заданным номером на новую дату. 2. Сформировать список авто одной модели, прошедших техосмотр в заданную дату.
27	Структура записи	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).
	Доп. операция	<ol style="list-style-type: none"> 1. Сформировать список нарушений по автомобилю заданного номера. Результат сохранить в новом двоичном файле с той же структурой записи, что и исходный файл. 2. Увеличить сумму штрафа вдвое по всем авто за указанную дату и по заданной статье.

1	28	Структура записи	Справочник банков по городам страны. Об отдельном банке хранятся данные: наименование, <u>код банка</u> , адрес (город), форма собственности (коммерческий или государственный).
		Доп. операция	1. Сформировать сведения по банкам города, с коммерческой формой собственности. 2. Заменить у банка с заданным кодом форму собственности на противоположную.
	29	Структура записи	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.
		Доп. операция	1. Прочитать запись по заданному номеру в файле, определить номер кассы и сформировать список операций, выполненных этой кассой в новом двоичном файле. 2. Внести изменения в запись заданной операции на заданной кассе в указанную дату: увеличить скидку на товар на 2% от цены товара.
	30	Структура записи	Киноафиша города. Структура записи о сеансе: название кинотеатра, название фильма, дата, время начала, стоимость билета.
		Доп. операция	1. Увеличить стоимость билета на заданный сеанс по дате и времени. 2. Сформировать афишу кинотеатра на заданную дату и сохранить ее в новом текстовом файле.

Структура отчета

Титульный лист

Содержание отчета

Задание 1

1. Условие задания

1.1.Формулировка.

1.2.Требования

2. Тестовый пример

Копия содержания текстового файла на примере 20 записей

3. Реализация приложения

3.1. Функционал приложения

Прототипы функций, реализующих операции задания (Код модуля).

Для каждой функции указать:

1) Операцию, которую она реализует

2) Предусловие и постусловие

3.2. Код основной программы

4. Результаты тестирования

Скриншоты результатов тестирования

Задание 2

1. Условие задания

1.1. Формулировка условия

1.2. Требования задания

1.3. Формулировка задания варианта

2. Тестовый пример

Копия содержания текстового файла на примере 5 записей для преобразования в двоичный файл.

Содержание двоичного файла (результат операции вывода двоичного файла на экран), полученного из данных текстового файла.

3. Реализация приложения

3.1. Структура записи двоичного файла (из кода) и ее размер в байтах (ручной расчет и системный).

3.2. Изображение структуры двоичного файла с записями фиксированной длины.

3.3. Функционал приложения

Прототипы функций, реализующих операции задания (Код модуля).

Для каждой функции указать:

3) Операцию, которую она реализует

4) Предусловие и постусловие

3.4. Код основной программы

4. Результаты тестирования

Скриншоты результатов тестирования всех операций.

2₁ Практика №3-4 . Разработка и программная реализация задач поиска данных в таблицах с применением механизма хеширования.

Тема. Применение хеш-таблицы для поиска данных в двоичном файле с записями фиксированной длины

Цель. Получить навыки по разработке хеш-таблиц и их применению при поиске данных в других структурах данных (файлах).

Задание

Разработать приложение, которое использует хеш-таблицу для организации прямого доступа к записям двоичного файла, реализованного в практической работе 2.

Метод разрешения коллизии представлен в вашем варианте задания в таб.1.

Требования к выполнению

1. Создать приложение и включить в него три заголовочных файла: управление хеш-таблицей, управление двоичным файлом (практическая работа 2), управление двоичным файлом посредством хеш-таблицы. Имена заголовочным файлам определите сами. Подключите заголовочные файлы к приложению.
2. 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.
3. Управление хеш-таблицей.
 - 1) Определить структуру элемента хеш-таблицы и структуру хеш-таблицы в соответствии с методом разрешения коллизии, указанным в варианте. Определения разместить в соответствующем заголовочном файле. Все операции управления хеш-таблицей размещать в этом заголовочном файле.
 - 2) Тестирование операций выполнять в функции main приложения по мере их реализации.

- 1 После тестирования всех операций, создать в заголовочном файле функцию с именем testHeshT переместить в нее содержание функции main, проверить, что приложение выполняется.

Разработать операции по управлению хеш-таблицей.

- 3) Разработать хеш-функцию (метод определить самостоятельно), выполнить ее тестирование, убедиться, что хеш (индекс элемента таблицы) формируется верно.
- 4) Разработать операции: вставить ключ в таблицу, удалить ключ из таблицы, найти ключ в таблице, рехешировать таблицу. Каждую операцию тестируйте по мере ее реализации.
- 5) 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.

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

4. Управление двоичным файлом.

Операции управления двоичным файлом: создание двоичного файла из текстового, добавить запись в двоичный файл, удалить запись с заданным ключом из файла, прочитать запись файла по заданному номеру записи.

Примечание. Эти операции должны быть отлажены в практической работе 2, или уже в этой работе, если их пока нету.

Структура записи двоичного файла и все операции, по управлению файлом, должны быть размещены в соответствующем заголовочном файле.

Выполнить тестирование операций в main приложения, и содержание функции main переместить в соответствующую функцию заголовочного файла с именем testBinF.

5. Управление файлом посредством хеш-таблицы.

В заголовочный файл управления файлом посредством хеш-таблицы 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.

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.

6. Составить отчет

Структура отчета представлена в приложении 1.

7. Ответы 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?

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

2		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</u> <u>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

2		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 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.

2		
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.

Приложение 1 Образец оформления отчета по разработке приложения

Условие задачи

Разработать приложение, которое использует хеш-таблицу для организации прямого доступа к записям файла, структура записи которого приведена в варианте.

Задание варианта

№	Метод хеширования (способ реализации коллизий)
---	--

1. Постановка задачи

Дано.

Файл двоичный с записями фиксированной длины.

Структура записи файла согласно варианту (изобразить в форме таблицы, указав названия полей).

Результат.

Хеш-таблица.

2. Подход к решению

1) Файл двоичный из записей фиксированного размера.

Указать:

Структура записи файла из кода.

Размер записи файла в байтах.

Прототипы операций по управления двоичным файлом с указанием предусловия и постусловия.

2) Хеш-таблица

Структура элемента таблицы: изобразить в виде строки таблицы, указать названия полей, их назначение.

Привести код элемента таблицы, реализацию структуры таблицы.

Описание алгоритмов операций:

- 2
 - вставка ключа в таблицу
 - поиск записи по ключу в таблице и возвращение номера записи в файле
 - удаление элемента из хеш-таблицы
- 3) Алгоритм поиска записи с заданным ключом в файле посредством хеш-таблицы.
3. Код приложения
4. Тестирование операций управления файлом посредством хеш-таблицы
 - Скриншоты результатов выполнения операций с хеш-таблицей в соответствии с требованиями задания п.3.
 - Скриншоты содержания файла и хеш-таблицы тестирования требований задания п.5.
5. Результат тестирования.

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.

32 Практика № 5-6 Нелинейные структуры данных. Бинарное дерево

Тема. Нелинейные структуры данных. Бинарное дерево.

Цель. Получение умений и навыков разработки и реализаций операций над структурой данных бинарное дерево.

Оглавление

1. Задание.....	1
2. Варианты заданий.....	2
3. Требования к оформлению отчета	6
4. Контрольные вопросы.....	6

1. Задание

Разработать программу в соответствии с требованиями варианта. Варианты заданий представлены в таблице 1.

Выбор варианта по формуле: остаток от деления «Номер по списку в журнале учета посещений» на 20. Если остаток 0, то вариант 1.

Совет. Выполните реализацию средствами ООП, операции будут методами класса.

Для вариантов с 1 по 7

Вид дерева: идеально сбалансированное из n узлов (не AVL).

1. Реализовать операции общие для вариантов с 1 по 7
 - 1) Создать идеально сбалансированное бинарное дерево из n узлов. Структура узла дерева включает: информационная часть узла, указатель на левое и указатель на правое поддерево. Информационная часть узла определена вариантом.
 - 2) Отобразить дерево на экране, повернув его справа налево.
2. Реализовать операции варианта.
3. Разработать программу, демонстрирующую выполнение всех операций на ваших тестах и тестах преподавателя.
4. Оформить отчет.

Для вариантов с 8 по 15

Вид дерева: дерево выражения.

1. Реализовать операции общие для вариантов с 8 по 15
 - 1) Создать дерево выражений в соответствии с вводимым выражением. Структура узла

дерева включает: информационная часть узла - символического типа: либо знак операции +, -, * либо цифра, указатель на левое и указатель на правое поддерево. В дереве выражения операнды в листьях дерева.

Исходное выражение имеет формат:

<формула>: :=цифра|<формула><знак операции><формула>

- 2) Отобразить дерево на экране, используя алгоритм ввода дерева повернутым справа налево.
2. Реализовать операции варианта.
3. Разработать программу, демонстрирующую выполнение всех операций на ваших тестах и тестах преподавателя.
4. Оформить отчет (требования к оформлению отчета приведены после таблицы с вариантами).

Для вариантов с 16 по 20

Вид дерева: бинарное дерево поиска (БДП).

1. Реализовать операции общие для вариантов с 16 по 20
 - 1) Создать бинарное дерево поиска (информационная часть узла определена вариантом).
Для этого реализовать операцию вставки нового значения в БДП и использовать ее при создании дерева.
 - 2) Отобразить дерево на экране, повернув его справа налево.
2. Реализовать операции варианта.
3. Разработать программу демонстрирующую выполнение всех операций на ваших тестах и тестах преподавателя.
4. Оформить отчет.

2. Варианты заданий

Таблица 1. Варианты заданий

Вариант	Значение информационной части	Операции варианта

1	Целое число	<p>Определить высоту дерева</p> <p>Определить длину пути дерева (количество ребер), используя алгоритм прямого обхода</p> <p>Вычисляет среднее арифметическое всех чисел в дереве.</p>
2	Целое число	<p>Определить количество листьев с положительными значениями</p> <p>Определить, сколько узлов дерева содержат заданное число.</p> <p>Увеличить значения узлов вдвое, обходя дерево алгоритмом в ширину.</p>
3	Символьное значение	<p>Вернуть самый левый узел дерева</p> <p>Определить длину пути (количество ребер) от корня до ближайшего узла с заданным значением</p> <p>Найти максимальное значение среди значений листьев дерева</p>
4	Целое число	<p>Определить, в каком поддереве исходного дерева четных чисел.</p> <p>Создать копию исходного двоичного дерева.</p>
5	Вещественное число	<p>Вычислить среднее арифметическое чисел левого поддерева, а так же и правого, по отдельности.</p> <p>Удалить двоичное дерево</p>
6	Целое число	<p>Используя рекурсивный алгоритм определить количество уровней в дереве.</p> <p>Вернуть узел с максимальным значением, обходя дерево в ширину.</p> <p>Определить максимальное и минимальное</p>
7	Символьное значение	<p>Определить уровень, на котором находится заданное значение.</p> <p>Определить количество цифр в левом поддереве исходного дерева.</p>

8	Символьное значение	Вывод дерева выражений по ширине Вернуть самый левый лист дерева Вычислить значение выражения
9	Символьное значение	Проверить, является ли дерево деревом выражений. Вывести дерево, отобразить его формулу Определить, содержит ли дерево операцию *
10	Символьное значение	Определить функцию упрощения дерева - выражения, заменяя в нем все поддеревья, соответствующие формулам : $(f+0)$, $(0+f)$, $(f-0)$, $(f*1)$, $(1*f)$ на поддеревья, соответствующие формуле f . А поддеревья вида $(f*0)$ и $(0*f)$ заменить на вершину с 0. Отобразить дерево, используя алгоритм в обхода в ширину
11	Символьное значение	Построить дерево выражения по префиксной форме арифметического выражения, в котором операнды одно буквенные идентификаторы и знаки арифметических операций. Вывести построенное дерево прямым обходом в глубину.
12	Символьное значение	Построить дерево по постфиксной форме арифметического выражения (представлено в строковом формате), в котором операнды однозначные числа и операции - арифметические операции. Вывести построенное дерево, обходя его в обратном порядке
13	Символьное значение	Образовать префиксную форму выражения, содержащегося в дереве выражения, и записать ее в строку. Вычислить значение выражения по дереву. Вычислить значение выражения по префиксной форме.

14	Символьное значение	Подсчитать количество узлов на заданном уровне. Вычислить значение выражения в левом поддереве. Вывести префиксную форму левого поддерева.
15	Символьное значение	Вычислить значение выражения в левом поддереве. Вычислить значение выражения в правом поддереве. Вернуть корень дерева и вычислить значение выражения, используя значения левого и правого подвыражений.
16	Целое число	1) Определить сумму значений, находящихся в листьях дерева, используя алгоритмом обратного обхода 2) Удалить максимальный элемент дерева. Считать, что такой элемент один. 3) Вставить новый элемент в дерево 4) Определить количество узлов в каждом поддереве.
17	Целое число	1) Определить среднее арифметическое всех узлов дерева, используя алгоритм обхода в «ширину». 2) Определить количество узлов в дереве. 3) Удалить самый левый лист дерева 4) Определить уровень, на котором находится заданное значение.
18	Символьное значение	1) Определить уровень, на котором находится узел с заданным значением. 2) Вставить новый узел в дерево 3) Определить, какое из поддеревьев выше. 4) Удалить дерево.
19	Содержит символьное значение и счетчик (подсчитывает сколько раз в дерево вставлялся символ)	1) Вставить новое значение в дерево с учетом того, что он может вставляться несколько раз (частота появления в тексте). 2) Определить, количество цифр в дереве, используя алгоритм обратного обхода. 3) Удалить узлы, содержащие не цифры. 4) Вывести значения узлов в порядке убывания.

20	Содержит текст и количество в нем цифр.	<p><u>Условие.</u> Узлы включаются в дерево в соответствии со значением количества цифр в тексте.</p> <p>1) Определите количество узлов, текст которых содержит более трех цифр.</p> <p>2) Удалить узел, не содержащий в тексте цифр.</p> <p>3) Вывести текст, в котором максимальное количество цифр.</p> <p>4) Создание копии дерева.</p>
----	---	---

3. Требования к оформлению отчета

- 1) Для каждой представленной в программе функции предоставить отчет по ее разработке в соответствии с требованиями разработки программы (подпрограммы).
- 2) Представить алгоритм основной программы и таблицу имен, используемых в алгоритме.
- 3) Ответить на контрольные вопросы.

4. Контрольные вопросы

1. Что определяет степень дерева?
2. Какова степень сильноветвящегося дерева?
3. Что определяет путь в дереве?
4. Как рассчитать длину пути в дереве?
5. Какова степень бинарного дерева?
6. Может ли дерево быть пустым?
7. Дайте определение бинарного дерева?
8. Дайте определение алгоритму обхода.
9. Приведите рекуррентную зависимость для вычисления высоты дерева.
10. Изобразите бинарное дерево, корень которого имеет индекс 6, и которое представлено в памяти таблицей вида

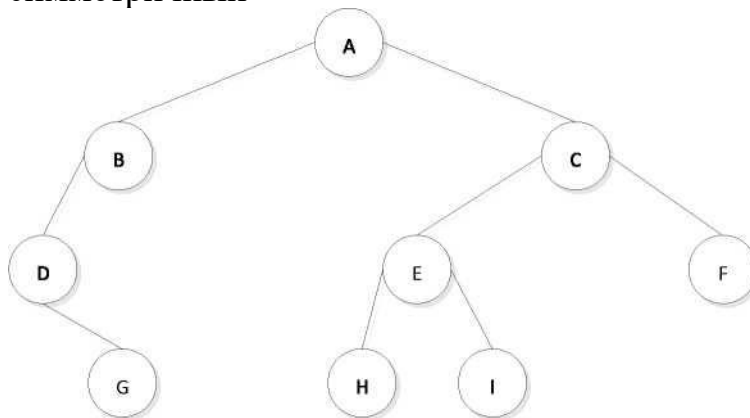
<u>Индекс</u>	<u>key</u>	<u>left</u>	<u>right</u>
1	12	7	3
2	15	8	NULL
3	4	10	NULL
4	10	5	9
5	2	NULL	NULL

3

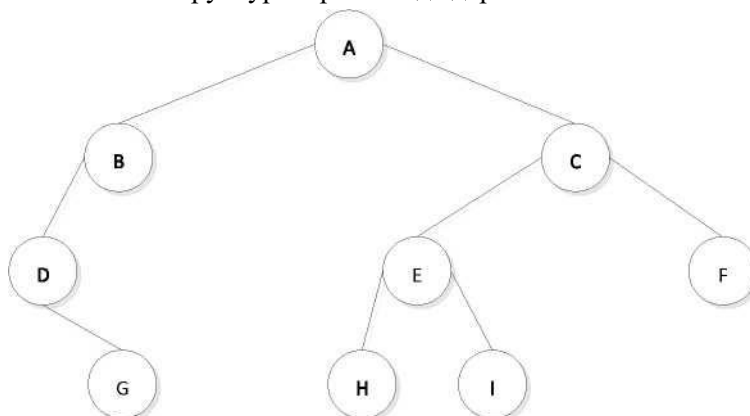
6	18	1	4
7	7	NULL	NULL
8	14	6	2
9	21	NULL	NULL
10	5	NULL	NULL

11. Укажите путь обхода дерева по алгоритму:
прямой; обратный;

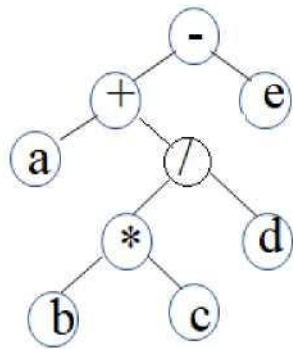
симметричный



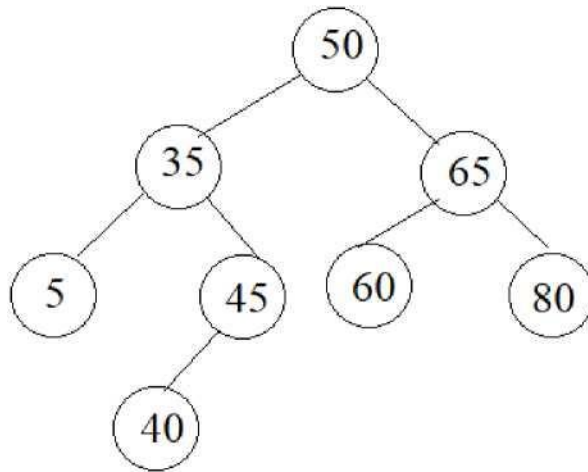
12. Какая структура используется в алгоритме обхода дерева методом в «ширину»?
13. Выведите путь при обходе дерева в «ширину». Продемонстрируйте использование структуры при обходе дерева.



14. Какая структура используется в не рекурсивном обходе дерева методом в «глубину»?
15. Выполните прямой, симметричный, обратный методы обхода дерева выражений.



16. Для каждого заданного арифметического выражения постройте бинарное дерево выражений:
 - 1) $a + b - c * d + e$
 - 2) $/ a - b * c d$
 - 3) $a b c d / - *$
 - 4) $* - / + a b c d e$
17. В каком порядке будет проходиться бинарное дерево, если алгоритм обхода в ширину будет запоминать узлы не в очереди, а в стеке?
18. Постройте бинарное дерево поиска, которое в результате симметричного обхода дало бы следующую последовательность узлов?
40 45 46 50 65 70 75
19. Приведенная ниже последовательность получена путем прямого обхода бинарного дерева поиска. Постройте это дерево.
50 45 35 15 40 46 65 75 70
20. Дано следующее бинарное дерево поиска



Покажите дерево

- 1) после включения узлов 1 48 75 100
- 2) после удаления узлов 5, 35
- 3) после удаления узла 45
- 4) после удаления узла 50
- 5) после удаления узлов 5, 35
- 6) после удаления узла 65 и вставки его снова
- 7) после удаления узлов 5, 35

43 Практика №7-9 Разработка и программная реализация структуры данных – Бинарное дерево поиска. Разработка программ, использующих созданную структуру.

Тема. Сбалансированные деревья поиска (СДП) и их применение для поиска

данных в файле

Цель

- получить навыки в разработки и реализации алгоритмов управления бинарным деревом поиска и сбалансированными бинарными деревьями поиска (АВЛ - деревьями);
- получить навыки в применении файловых потоков прямого доступа к данным файла;
- получить навыки в применении сбалансированного дерева поиска для прямого доступа к записям файла.

Задания

Задание 1. Для всех вариантов. Разработать приложение, которое использует бинарное дерево поиска (БДП) для поиска записи с ключом в файле, структура которого представлена в задании 2 вашего варианта.

1. Разработать класс (или библиотеку функций) «Бинарное дерево поиска». Тип информационной части узла дерева: ключ и ссылка на запись в файле (как в практическом задании 2). Методы: включение элемента в дерево, поиск ключа в дереве, удаление ключа из дерева, отображение дерева.
2. Разработать класс (библиотеку функций) управления файлом (если не создали в практическом задании 2). Включить методы: создание двоичного файла записей фиксированной длины из заранее подготовленных данных в текстовом файле; поиск записи в файле с использованием БДП; остальные методы по вашему усмотрению.
3. Разработать и протестировать приложение.
4. Подготовить отчет

Задание 2

Разработать приложение, которое использует сбалансированное дерево поиска, предложенное в варианте, для доступа к записям файла.

1. Разработать класс СДП с учетом дерева варианта. Структура информационной части узла дерева включает ключ и ссылку на запись в файле (адрес места размещения). Основные методы: включение элемента в дерево; поиск ключа в дереве с возвратом ссылки; удаление ключа из дерева; вывод дерева в форме дерева (с отображением структуры дерева).
2. Разработать приложение, которое создает и управляет СДП в соответствии с заданием.
3. Выполнить тестирование.
4. Определить среднее число выполненных поворотов (число поворотов на общее число

- 3 вставленных ключей) при включении ключей в дерево при формировании дерева из двоичного файла.

5. Оформить отчет

6. Варианты индивидуальных заданий задания 2

№	Сбалансированное дерево поиска (СДП)	Структура элемента множества (ключ - подчеркнутое поле) остальные поля представляют данные элемента
1	Красно - черное	Читательский абонемент: номер читательского - целое пятизначное число, ФИО, Адрес
2	АВЛ	Счет в банке: номер счета 7 разрядное число, ФИО, Адрес
3	Косое	Владелец телефона: номер телефона
4	Рандомизированное	Владельцев автомобилей. номер машины, марка, сведения о владельце.
5	В-дерево	Пациент поликлиники: номер карточки, код хронического заболевания, Фамилия лечащего врача
6	Красно - черное	Товар: название, код - шестизначное число
7	АВЛ	Специализация вуза: код специальности, название вуза
8	Косое	Книга: ISBN - двенадцатизначное число, Автор, Название
9	Рандомизированное	Страховой полис: номер, компания, фамилия владельца
10	В-дерево	Англо - русский словарь: английское слово, русское слово
11	Красно - черное	Железнодорожная справка: номер поезда, пункт отправления, пункт назначения, время отправления
12	АВЛ	Регистрация малого предприятия: номер лицензии, название, учредитель

13	Косое	Студент: номер зачетной книжки, номер группы ФИО
14	Рандомизированное	Справочная межгорода: код города, название города
15	В-дерево	Тест. Файл хранит сведения по тестам. Тесты пронумерованы случайными 4-хзначными целыми числами. Для каждого теста по 4 вопроса (текст с номером вопроса), номер верного ответа..
16	Красно - черное	Частотный словарь: слово, количество вхождений в текст
17	АВЛ	Частотный словарь: слово, количество вхождений в текст.
18	Косое	Читательский билет: номер читательского билета, ФИО читателя, номер телефона.
19	Рандомизированное	Вызов такси: номер машины, фамилия водителя, время выезда, отметка о присутствии в гараже.
20	В-дерево	Продажи товаров: код товара (артикул), название, цена, дата изготовления
21	Красно - черное	Сотрудник организации: табельный номер, название должности, оклад, дата приема на работу.
22	АВЛ	Расписание занятий группы: номер группы, количество студентов, шифр направления, шифр профиля.
23	Косое	Нагрузка по дисциплине: код дисциплины, код направления подготовки, название дисциплины, номер семестра проведения дисциплины.
24	Рандомизированное	Нагрузка по дисциплине: код дисциплины, код направления подготовки, название дисциплины, номера семестров проведения дисциплины, количество часов в семестре.
25	В-дерево	Аэропорт (табло прибытия пассажирских авиарейсов сохраняет все данные в файле): пункт вылета, номер рейса, дата прилета,

		время прилета, информация о задержке прилета в часах.
26	Цепное хеширование	Учет заболеваний пациента. Структура записи о пациенте: номер полиса, фамилия, имя, отчество, код заболевания, дата установки диагноза, код врача.
27	Открытый адрес (смещение на 1)	Учет техосмотра автомобилей. Структура записи об автомобиле: Номер автомобиля(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: номер автомобиля, 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, код банка, адрес (город), форма собственности (коммерческий или государственный).
30	Цепное хеширование	Касса магазина. Структура записи операции по кассе: номер кассы, 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, количество свободных мест

Задание 3

Выполнить анализ алгоритма поиска записи с заданным ключом при применении структур данных:

- 3 - хеш - таблица;
- бинарное дерево поиска;
- СДП

Требования по выполнению задания

1. Протестировать на данных: а) небольшого объема; б) большого объема.
2. Построить хеш-таблицу из чисел файла.
3. Осуществить поиск введенного целого числа в двоичном дереве поиска, в сбалансированном дереве и в хеш-таблице. Оформить таблицу результатов по форме:

Вид поисковой структуры	Количество элементов, загруженных в структуру в момент выполнения поиска	Емкостная сложность: объем памяти для структуры	Количество выполненных сравнений, время на поиск ключа в структуре
-------------------------	--	---	--

4. Провести анализ алгоритма поиска ключа на исследованных поисковых структурах на основе данных, представленных в таблице.
5. Оформить отчет

Приложение 1 Структура оформления отчета Отчет по каждому заданию включить в один отчет.

1. Титульный лист
2. Отчет по заданию 1

Условие задания

1. Постановка задачи

Разработайте приложение, которое использует БДП для организации прямого доступа к записям файла, структура записи которого приведена в варианте (отобразить задачу варианта)

Дано.

Файл двоичный с записями фиксированной длины.

Структура записи файла:

Результат.

Приложение, выполняющее операции (перечислить для файла и поисковой структурой)

4 2. Подход к решению

Описать ваш выбор решения. Например, класс или просто операции оформляются функциями. Описать методы (операции)

Например:

1) БДП — класс

2) *Структура элемента БДП: ключ, ссылка на запись в файле.*

3) Методы класса БДП: вставить ключ в БДП, удалить ключ из БДП, найти ключ в БДП, отобразить БДП в консоль.

3. Файл двоичный из записей фиксированного размера.

Структура записи файла: (привести)

Операции по управлению файлом(перечислить)

4. Операции по управлению БДП Перечислить и привести их прототипы

5. Алгоритмы операций на псевдокоде:

- вставка в БДП элемента;
- поиск записи по ключу в БДП и возвращение ссылки на запись в файле;

6. - удаление элемента БДП;

7. Код приложения

8. Содержание файла (лучше взять из текстового).

9. Скриншот результатов тестирования операций с БДП на файле.

3. Отчет по заданию 2

По форме отчета задания 1.

Привести ответ на задание: определить среднее число выполненных поворотов (число поворотов на общее число вставленных ключей) при включении ключей в дерево при формировании дерева из двоичного файла.

4. Отчет по заданию 3

Таблица результатов и
выводы.

54 Практика 10-11 Разбор задач с применением: метода динамического программирования (задача о камнях, треугольник), метод решета

Тема. Алгоритмические стратегии или методы разработки алгоритмов. Перебор и методы его сокращения.

Цель. Получить навыки применения методов, позволяющих сократить число переборов в задачах, которые могут быть решены только методом перебора всех возможных вариантов решения.

Разработка и программная реализация задач с применением метода сокращения числа переборов

Задание

Требования

1. Разработать алгоритм решения задачи с применением метода, указанного в варианте и реализовать программу.

- 1) Оценить количество переборов при решении задачи стратегией «в лоб» - грубой силы
- 2) Привести анализ снижения числа переборов при применении метода.

3) Оформить отчет в соответствии с требованиями документирования разработки ПО: Постановка задачи, Описание алгоритмов и подхода к решению, Код, результаты тестирования.

№ Варианта	Задача	Метод
1	Посчитать число последовательностей нулей и единиц длины n , в которых не встречаются две идущие подряд единицы.	Динамическое программирование
2	Дана последовательность целых чисел. Необходимо найти ее самую длинную строго возрастающую подпоследовательность.	Динамическое программирование
3	Дана строка из заглавных букв латинского алфавита. Необходимо найти длину наибольшего палиндрома, который можно получить вычеркиванием некоторых букв из данной строки.	Динамическое программирование

4	Имеется рюкзак с ограниченной вместимостью по массе; также имеется набор вещей с определенным весом и ценностью. Необходимо подобрать такой набор вещей, чтобы он помещался в рюкзаке и имел максимальную ценность (стоимость).	Динамическое программирование
5	Дано прямоугольное поле размером $n*m$ клеток. Можно совершать шаги длиной в одну клетку вправо или вниз. Посчитать, сколькими способами можно попасть из левой верхней клетки в правую нижнюю.	Динамическое программирование
6	Дано прямоугольное поле размером $n*m$ клеток. Можно совершать шаги длиной в одну клетку вправо, вниз или по диагонали вправо вниз. В каждой клетке записано некоторое натуральное число. Необходимо попасть из верхней левой клетки в правую нижнюю. Вес маршрута вычисляется как сумма чисел со всех посещенных клеток. Необходимо найти маршрут с минимальным весом.	Динамическое программирование

7

Нам нужно соорудить путь, соединяющий

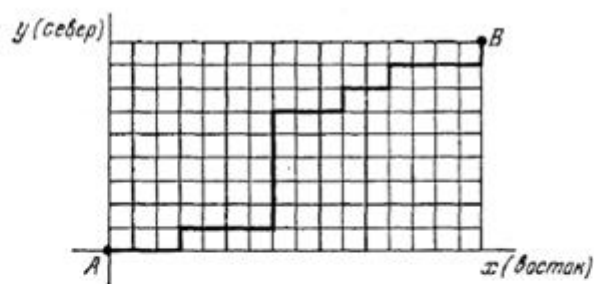
Динамическое
программирование

Рис. 13.1.

два пункта A и B , из которых второй лежит к северо-востоку от первого. Для простоты допустим, что прокладка пути состоит из ряда шагов, и на каждом шаге мы можем двигаться либо строго на восток, либо строго на север; любой путь из A в B представляет собой ступенчатую ломаную линию, отрезки которой параллельны одной из координатных осей (рис. 13.1). Затраты на сооружение каждого из таких отрезков известны. Требуется проложить такой

путь из A в B , при котором суммарные затраты минимальны.

8	<p>Черепашке нужно попасть из пункта А в пункт В. Поле движения разбито на квадраты. Известно время движения вверх и вправо в каждой клетке (улицы). На каждом углу она может поворачивать только на север или только на восток. Найти минимальное время, за которое черепашка может попасть из А в В.</p>	Динамическое программирование
9	<p>Треугольник. Треугольник имеет вид представленный на рисунке. Напишите программу, которая вычисляет наибольшую сумму чисел, расположенных на пути начинающемся в верхней точке треугольника и заканчивающегося на основании треугольника.</p> <pre> 7 3 8 8 1 0 2 7 4 4 4 5 2 6 5 </pre>	Динамическое программирование
10	Из листа клетчатой бумаги вырезали фигуру ровно по границам клеточек. Разработать программу определения площади вырезанной фигуры.	метода ветвей и границ
11	Разработать программу расстановки на стандартной 64-клеточной шахматной доске 8 ферзей так, чтобы ни один из них не находился под боем другого.	метода ветвей и границ
12	Разработать программу поиска и вывода всех гамильтоновых циклов в произвольном графе.	метода ветвей и границ

13	<p>Пронумеровать позиции в матрице размером 5×5 следующим образом: если номер i ($1 \leq i \leq 25$) соответствует позиции (x, y) то номер $i+1$ может соответствовать позиции с координатами (z, w), вычисляемыми по одному из следующих правил:</p> <p>$(z, w) = (x \pm 3, y)$ $(z, w) = (x, y \pm 3)$ $(z, w) = (x \pm 2, y \pm 2)$</p> <p>Написать программу, которая последовательно нумерует позиции матрицы при заданных координатах позиции, в которой поставлен номер 1 (т.е. в некоторой позиции матрицы содержится номер 1).</p>	метода ветвей и границ
14	<p>Замок состоит из комнат. Сам замок имеет прямоугольную форму и разделен на $M \times N$ клеток ($M \leq 50$; $M \geq 50$). каждая клетка может иметь от 0 до 4 стен. Написать программу, которая определяет</p> <ul style="list-style-type: none"> - количество комнат в замке. - площадь наибольшей комнаты - какую стену в замке следует удалить, чтобы получить комнату наибольшей площади. <p>Пример плана замка</p>	метода ветвей и границ

15	Автозаправка. Вдоль кольцевой дороги расположено M городов. В каждом городе есть автозаправка. Известна стоимость $Z[i]$ заправки горючим в городе с номером i и стоимость $C[i]$ проезда по дороге, соединяющей i -ый и $(i+1)$ -й города и стоимость проезда между первым и M -ым городами. Города пронумерованы по часовой стрелке. Определить для жителей каждого города тот город, в котором им выгодно заправляться, и направление «по часовой стрелке» или «против часовой стрелки»	метода ветвей и границ
16	В массиве размером $M \times N$, заполненном нулями и единицами найти квадратный блок, состоящий из одних нулей.	метода ветвей и границ
17	Задача о коммивояжере	метода ветвей и границ

18	Монетная система некоторого государства состоит из монет достоинством $a_1 = 1 < a_2 < \dots < a_n$. Требуется выдать сумму наименьшим возможным количеством монет.	Жадный алгоритм
19	Разработать процедуру оптимального способа расстановки скобок в произведении последовательности матриц, размеры которых равны $(5, 10, 3, 12, 5, 50, 6)$, чтобы количество скалярных умножений стало минимальным (максимальным).	Жадный алгоритм
20	Решить задачу о раскраске вершин графа. Применить к задаче управлению светофорами на сложном перекрестке. (Источник постановки задачи - книга: А. Ахо, Д. Хопкрофт, Дж. Ульман Структуры данных и алгоритмы).	Жадный алгоритм

21	<p>Задача о загрузке машины.</p> <p>Имеется определенный набор предметов P_1, P_2, \dots, P_n (каждый в единственном экземпляре); известны их веса q_1, q_2, \dots, q_n и стоимости c_1, c_2, \dots, c_n. Грузоподъемность машины равна Q. Спрашивается, какие из предметов нужно взять в машину, чтобы их суммарная стоимость (при суммарном весе $\leq Q$) была максимальна?</p>	Динамическое программирование
22	<p>Задача о загрузке машины.</p> <p>Имеется определенный набор предметов P_1, P_2, \dots, P_n (каждый в единственном экземпляре); известны их веса q_1, q_2, \dots, q_n и стоимости c_1, c_2, \dots, c_n. Грузоподъемность машины равна Q. Спрашивается, какие из предметов нужно взять в машину, чтобы их суммарная стоимость (при суммарном весе $\leq Q$) была максимальна?</p>	Жадный алгоритм

64 Практика №12-14 Разработка и программная реализация класса Граф: создание, алгоритмы обхода. Исследование алгоритмов Прима и Крускала построения минимального покрывающего дерева. Топологическая сортировка.

Тема. Основные алгоритмы работы с графами.

Цель. Получение практических навыков по выполнению операций над структурой данных граф.

Задание.

1. Разработать класс «Граф», обеспечивающий хранение и работу со структурой данных «граф», в соответствии с вариантом индивидуального задания. Реализовать метод ввода графа с клавиатуры, наполнение графа осуществлять с помощью метода добавления одного ребра. Реализовать метод вывода графа и методы, выполняющие задачи, определенные вариантом индивидуального задания.
2. Разработать программу, демонстрирующую работу всех методов класса.
3. Произвести тестирование программы на графе, предложенном в таблице 2.
4. Составить отчет, отобразив в нем описание выполнения всех этапов разработки, тестирования и код всей программы со скриншотами результатов тестирования.

Таблица 1. Варианты заданий

Номер варианта	Представление графа в памяти	Задачи
1	Матрица смежности	Определить центр графа. Составить программу реализации алгоритма Крускала построения остовного дерева минимального веса. Разработать доступный способ (форму) вывода результирующего дерева на экран монитора.
2	Список смежных вершин	Определить мосты графа. Составить программу реализации алгоритма Крускала построения остовного дерева минимального веса. Разработать доступный способ (форму) вывода результирующего дерева на экран монитора.

3	Матрица смежности	Найти и Вывести Эйлеров цикл в графе. Составить программу реализации алгоритма Прима построения остовного дерева минимального веса. Разработать доступный способ (форму) вывода результирующего дерева на экран монитора.
4	Список смежных вершин	Определить диаметр графа. Составить программу реализации алгоритма Прима построения остовного дерева минимального веса. Разработать доступный способ (форму) вывода результирующего дерева на экран монитора.
5	Матрица смежности	Вывести все цепочки в графе, используя метод поиска в глубину. Составить программу нахождения кратчайших путей в графе заданным методом «Построения дерева решений». Вывести пути, полученные методом
6	Список смежных вершин	Вывести все цепочки в графе, используя метод поиска в ширину. Составить программу нахождения кратчайших путей в графе заданным методом «Естественное слияние». Вывести пути, полученные методом
7	Матрица смежности	Определить, является ли граф связным. Составить программу нахождения кратчайшего пути в графе от заданной вершины к другой заданной вершине методом «Дейкстры» и вывести этот путь.
8	Список смежных вершин	Составить программу нахождения кратчайших путей методом «Флойда». Используя результат алгоритма вывести путь между вводимыми парами вершин.
9	Матрица смежности	Определить медиану неориентированного графа. Составить программу нахождения кратчайших путей методом «Йена». Используя результат алгоритма вывести путь между вводимыми парами вершин.
10	Список смежных вершин	Определить <i>Эксцентриситет</i> заданного узла. Составить программу нахождения кратчайших путей методом «Беллмана-Форда». Используя результат алгоритма вывести путь между вводимыми парами вершин.

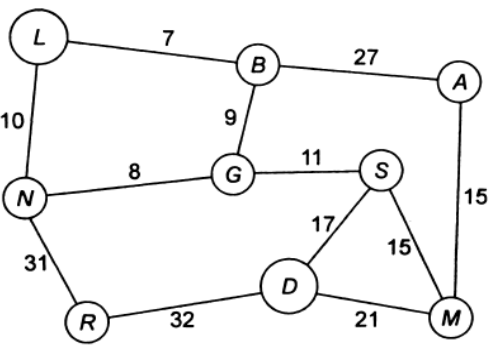
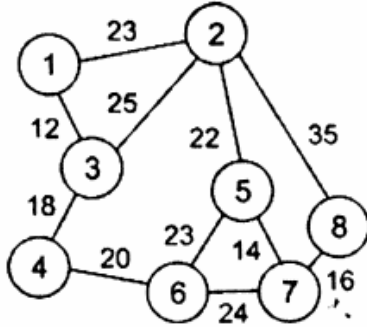
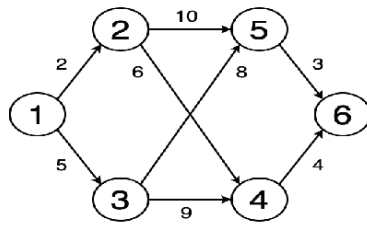
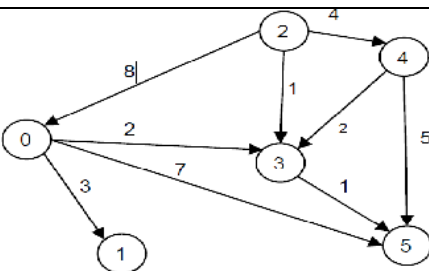
11	Матрица смежности	Определить, является ли граф связанным. Составить программу нахождения кратчайшего пути в графе от заданной вершины к другой заданной вершине методом «Дейкстры». Вывести этот путь.
12	Список смежных вершин	Определить, есть ли в графе Эйлера цикл. Составить программу реализации алгоритма построения остова дерева обходом дерева в глубину для неориентированного графа. Разработать доступный способ (форму) вывода результирующего дерева на экран монитора.
13	Матрица смежности	Определить глубину графа. Составить программу реализации алгоритма построения остова дерева методом поиска в ширину в неориентированном графе. Разработать доступный способ (форму) вывода результирующего дерева на экран монитора.
14	Список смежных вершин	Определить глубину графа. Составить программу, которая определяет является ли граф ациклическим.
15	Список смежных вершин	Определить степень вершины графа. Составить программу нахождения кратчайшего пути в графе от заданной вершины к другой заданной вершине методом «Дейкстры». Вывести этот путь.
16	Матрица смежности	Найти корень ациклического графа. Составить программу нахождения самого длинного пути в ациклическом графе. Вывести кратчайший путь между двумя заданными вершинами.
17	Список смежных вершин	Составить «рейтинг» вершин с наибольшим количеством соседей. Составить программу нахождения кратчайшего пути в графе от заданной вершины к другой заданной вершине методом «Дейкстры». Вес каждого ребра в пути не должны превышать К. Вывести этот путь.
18	Матрица смежности	Определить цикл с наименьшим весом, состоящий из К вершин. Наличие ребра между вершинами означает знакомство двух людей. Определить, можно ли разбить людей на 2 группы, чтобы в каждой группе были только незнакомые люди.

9	Список смежных вершин	Найти и вывести цикл Гамильтона. Составить программу нахождения кратчайшего пути в графе от заданной вершины к другой заданной вершине методом «Дейкстры». Вес каждого ребра в пути не должны быть меньше К. Вывести этот путь.
20	Матрица смежности	Определить цикл с наибольшим весом, состоящий из К вершин. Наличие ребра между вершинами означает знакомство двух людей. Определить, можно ли опосредованно познакомиться их всех между собой.

Таблица 2. Примеры графов для тестирования алгоритмов

№	Граф
1	
2	
3	

5 4	
5	
5	
7	
8	

5 9	
10	
11	
12	

75 Практика № 15-16 Подробное рассмотрение реализации алгоритма Хаффмана. Упражнения по выполнению сжатия данных методом группового кодирования RLE, Шеннона-Фано, Лемпеля-Зева

Тема. Кодирование и сжатие данных методами без потерь.

Цель. Получение практических навыков и знаний по выполнению сжатия данных рассматриваемыми методами

Требуется выполнить три задания. Программировать алгоритмы не обязательно.

Оглавление

Задание 1. Применение алгоритма группового сжатия текста RLE	1
Задание 2. Исследование алгоритмов группового сжатия (методы Лемпеля –Зива: LZ77, LZ78) на примерах.....	2
Задание 3. Разработать программы (или только алгоритмы на псевдокоде или описать словесно) сжатия и восстановления текста методами Шеннона-Фано и Хаффмана. Представить процесс выполнения алгоритма для задачи варианта. Все результаты представить в отчете.....	2
Таблица 1. Варианты к заданиям 2 и 3	3
Приложение 1. Примеры оформления выполнения заданий 2 и 3	9
Приложение 2. Оформление отчета по заданию 3	10

Задание 1. Применение алгоритма группового сжатия текста RLE

Сжать текст, используя метод RLE (run length encoding/кодирование длин серий/групповое кодирование).

Требования к выполнению задания

- 1) Описать процесс сжатия алгоритмом RLE.
- 2) **Придумать текст**, в котором есть длинные (в разумных пределах) серии из повторяющихся символов. Выполнить сжатие текста. Рассчитать коэффициент сжатия.
- 3) **Придумать текст**, в котором много неповторяющихся символов и между ними могут быть серии. Выполнить групповое сжатие, показать коэффициент сжатия. Применить алгоритм разделения текста при групповом кодировании, позволяющий повысить эффективность сжатия этого текста. Рассчитать коэффициент сжатия после применения алгоритма.
- 4) Оформить отчет. В отчете представьте ответы на вопросы пунктов этого задания с 1 по 3.

Задание 2. Исследование алгоритмов группового сжатия (методы

Шемпеля -Зива: LZ77, LZ78) на примерах.

Тексты для сжатия по вариантам в табл. 1 в столбце 1 (для LZ77) и в столбце 2 (для LZ78).

Требования к выполнению заданию

- 1) Выполнить каждую задачу варианта задания, представив алгоритм решения в виде таблицы и указав результат сжатия.

Примеры оформления решения в отчете представлены в Приложении I этого документа.

- 2) Описать процесс восстановления сжатого текста.
- 3) Сформировать отчет, включив задание, вариант задания, результаты выполнения задания варианта (таблица) .

Задание 3. Разработать программы (или только алгоритмы на псевдокоде или описать словесно) сжатия и восстановления текста методами Шеннона-Фано и Хаффмана. Представить процесс выполнения алгоритма для задачи варианта. Все результаты представить в отчете.

Требования к выполнению заданию

1. Сформировать отчет по разработке каждого алгоритма в соответствии с требованиями.

1.1. По методу Шеннона-Фано.

- 1) Данными для выполнения задания является текст, представленный в табл. 1 в столбце 3.
- 2) Привести постановку задачи, описать алгоритм формирования префиксного дерева и алгоритм кодирования, декодирования.
- 3) Представить таблицу формирования кода для текста варианта задания.
- 4) Изобразить префиксное дерево формирования кода.
- 5) Рассчитать коэффициент сжатия.

1.2. По методу Хаффмана.

- 1) Данные для выполнения задания: ваша фамилия имя отчество.
- 2) Привести постановку задачи, описать алгоритм формирования префиксного дерева и алгоритм кодирования, декодирования.
- 3) Построить таблицу частот встречаемости символов в исходной строке для чего сформировать алфавит исходной строки и посчитать количество вхождений (частот) символов и их вероятности появления.
- 4) Изобразить префиксное дерево Хаффмана.
- 5) Упорядочить построенное дерево слева-направо (при необходимости) и изобразить его.

- 5 б) Провести кодирование исходной строки по аналогии с примером:

п 1010 у 11000 п 1010 к 1011 и 00 н 11001 «» 010 в 011 а 11010 с 11011 и 00 л 100 и 11100
 «» к и р и л л о в и ч
 010 1011 00 11101 00 100 100 11110 011 00 11111

- 7) Рассчитать коэффициенты сжатия относительно кодировки ASCII и относительно равномерного кода.
 - 8) Рассчитать среднюю длину полученного кода и его дисперсию.
 - 9) По результатам выполненной работы сделать выводы и сформировать отчет. Отобразить результаты выполнения всех требований (с 1 по 8), предъявленных в задании и оформить разработку программы: постановка, подход к решению, код, результаты тестирования.
- 1.3. По желанию!!! Реализовать и отладить программу. Применить алгоритм Хаффмана для архивации данных текстового файла. Выполнить практическую оценку сложности алгоритма Хаффмана. Провести архивацию этого же файла любым архиватором. Сравнить коэффициенты сжатия разработанного алгоритма и архиватора.

Таблица 1. Варианты к заданиям 2 и 3

Вариант	Сжатие данных по методу Лемпеля-Зива LZ77 Используя двухсимвольный алфавит (0, 1) закодировать следующую фразу:	Закодировать следующую фразу, используя код LZ78	Закодировать фразу методами Шеннона-Фано
	1	2	3
1	0001010010101001101	кукурукурекурекун	Ана, дэус, рики, паки, Дормы кормы констунтаки, Дэус дэус канадэус - бац!

2	010010001001000010 1	упупапекапекаууп	One, two, Freddy's coning for you Three, four, aetter lock your door Five, six, graa a crucifix Seven, eight, gonna stay up late.
3	010010101001000010 1	лорлоралоранранлоран	Эне-бене, рики-таки, Буль-буль-буль, Караки- шмаки Эус- деус- краснодеус бац
4	010000100000010000 1	пропронепронепрнепрона с	Кони-кони, коникони, Мы сидели на балконе, Чай пили, воду пили, По-турецки говорили.
5	101000100101010001 01 1	какатанекатанекатата	Прибавь к ослиной голове Еще одну, получишь две. Но сколько б ни было ослов, Они и двух не свяжут слов.
6	000101110110100111	менменаменаменатеп	По-турецки говорили. Чяби, чяряби Чяряби, чяби-чяби. Мы набрали в рот воды.
7	110101011001100001 00 1	долделдолдилделдил	Тише, мыши, кот на крыше, А котята ещё выше. Кот пошёл за молоком, А котята кувырком.

8	010110110110100010 00 1	sarsalsarsanlasanl 33	Мой котёнок очень странный, Он не хочет есть сметану, К молоку не прикасался И от рыбки отказался.
9	000100101100100010 00 1	klonklonklonklonkl	Эни-бени рити-Фати. Дорба, дорба сентибрати. Дэл. Дэл. Кошка. Дэл. Фати!
10	111010011011000110 1	tertrektekertektrek	Самолёт-вертолёт! Посади меня в полёт! А в полёте пусто - Выросла капуста.
11	101010011011001110 10	aigaoneaigaoreaigao	Кот пошёл за молоком, А котята кувырком. Кот пришёл без молока, А котята ха-ха-ха.
12	000100101010100110 1	connerconneconnerce	Цветом мой зайчишка - белый, А ещё, он очень смелый! Не боится он лисицы, Льва он тоже не боится.
13	010110110010101010 11	weaweraweawerwea	Эне, бене, лики, паки, Цуль, буль-буль, Калики-цваки, Эус- беус, кликмадеус, бокс.
14	001010011001000000 1	porpoterpoterporter	Ана-дэус-рики-паки, Дормы-кормыконсту- таки, Энус-дэус-кана- дэусБАЦ!

15	10110111100110001101	nantopnento ⁵ ponantonen	Раз, два - упала гора; три, четыре - прицепило; пять, шесть - бьют шерсть; семь, восемь - сено косим.
16	0100101010010000101	roporopoterropoterter	Зуба зуба, зуба зуба, Зуба дони дони мэ, А шарли буба раз два три, А ми раз два три замри.
17	0001000010101001101	weawerawe ¹ aerweaerwea	Плыл по морю чемодан, В чемодане был диван, На диване ехал слон. Кто не верит - выйди вон!
18	1110100110111001101	sionsinossionsinos ¹	Дрынцы- брынцыбубен-цы, Раз- звонилисьудальцы, Диги-диги-диги-дон, Выхо-ди-скорее-вон!
19	0001000010101001101	conconconconacon ¹	Перводан, друго ¹ дан, На колоде барабан; Свистель, коростель, Пятерка, шестерка, утюг.
20	0100101010010000101	nantopnento ¹ ponantonen	Эни бэни рики паки Турбаурбасентибряки. Может - выйдет, может - нет, В общем - полный Интернет

21	1010011011000110111	terhrekhekerhekhrek	Матросская шапка, Верёвка в руке, Тяну я кораблик По быстрой реке. И скачут лягушки За мной по пятам, И просят меня: — Прокати, капитан!
22	10100110110011101001	aigaoneaigaoreaigao	Спать пора! Уснул бычок, Лёг в коробку на бочок. Сонный мишка лёг в кровать, Только слон не хочет спать. Головой кивает слон, Он слонихе шлёт поклон.
23	1001010101001101000	connerconneconnerse	Я свою сестренку Лиду Никому не дам в обиду! Я живу с ней очень дружно, Очень я ее люблю. А когда мне будет нужно, Я и сам ее побью.
24	11011001010101011010	werwerrwererwererwer	Как большой, сидит Андрюшка На ковре перед крыльцом. У него в руках игрушка — Погремушка с бубенцом. Мальчик смотрит — что за чудо? Мальчик очень удивлён, Не поймёт он: ну откуда Раздаётся этот звон?
25	0100110010000001001	porpoherpoherporher	Нет, напрасно мы решили Прокатить кота в машине: Кот кататься не привык — Опрокинул грузовик.

26	10111100110001101101	nanhornenhorokanhoken	Наша Таня громко плачет: Уронила в речку мячик. — Тише, Танечка, не плачь: Не утонет в речке мяч.
27	0101010010000101010	rorororoherroroherher	У меня живёт козлёнок, Я сама его пасу. Я козлёнка в сад зелёный Рано утром отнесу. Он заблудится в саду — Я в траве его найду.
28	1000010101001101000	ceaceraceaerceaercea	Плыл по морю чемодан, В чемодане был диван, На диване ехал слон. Кто не верит - выйди вон!
29	1010011011100110111	suonsunossuonsunos	Воробей по лужице Прыгает и кружится. Перышки взъерошил он, Хвостик распушил. Погода хорошая! Чил-чив-чил!
30	0100001010100110100	aokaonaokaonaok	Пять зелёных лягушат В воду броситься спешат — Испугались цапли! А меня они смешат: Я же этой цапли Не боюсь ни капли!

Приложение 1. Примеры оформления выполнения заданий 2 и 3

1. Оформление задания 2

1. Для метода Лемпеля –Зива LZ77 для сжатия двоичного кода

Исходный текст	000000001111111111110 00000000011011110
LZ-код	0.00.100.001.011.1011.1101. 1010.0110.10010.10001.10110.
R	2 3 4
Вводимые коды	– 10 11 100 101 110 111 1000 1001 1010 1011 1100

Где LZ – сжатый текст (в данном примере в связи с небольшим размером исходного текста размер текста не уменьшился)

R отмечает шаги кодирования, после которых происходит переход на представление кодов A увеличенным числом разрядов R. Так, на первом шаге вводится код 10 для комбинации 00, и поэтому на следующих двух шагах $R = 2$, после третьего шага $R = 3$, после седьмого шага $R = 4$, т.е. в общем случае $R = K$ после шага $2^{K-1} - 1$.

2. Для метода Лемпеля –Зива LZ78 для сжатия текста

В отличие от LZ77, работающего с уже полученными данными, LZ78 ориентируется на данные, которые только будут получены (LZ78 не использует скользящее окно), он хранит словарь из уже просмотренных фраз.

Пример 1. Дан текст аааааааа. Сжать текст, используя метод LZ78

Словарь

Ссылка на символ	Символы словаря	код
1	а	<0,а>
2	а	<0,а>
3	аа	<1,а>
4	аа	<1,а>
5	ааа	<3,а>

Таблица 1

Символ	Кол-во	1-я цифра	2-я цифра	3-я цифра	4-я цифра	5-я цифра	Код	Кол-во бит
пробел	5	0	0	0			000	15
ш	4	0	0	1			001	12
е	3	0	1	0			010	9
,	3	0	1	1			011	9
и	3	1	0	0			100	9
т	3	1	0	1	0		1010	12
ы	2	1	0	1	1		1011	8
к	2	1	1	1	0		1110	8
н	1	1	1	1	1		1111	4
о	1	1	1	0	0	0	11000	5
а	1	1	1	0	0	1	11001	5
м	1	1	1	0	1	0	11010	5
р	1	1	1	0	1	1	11011	5
								106

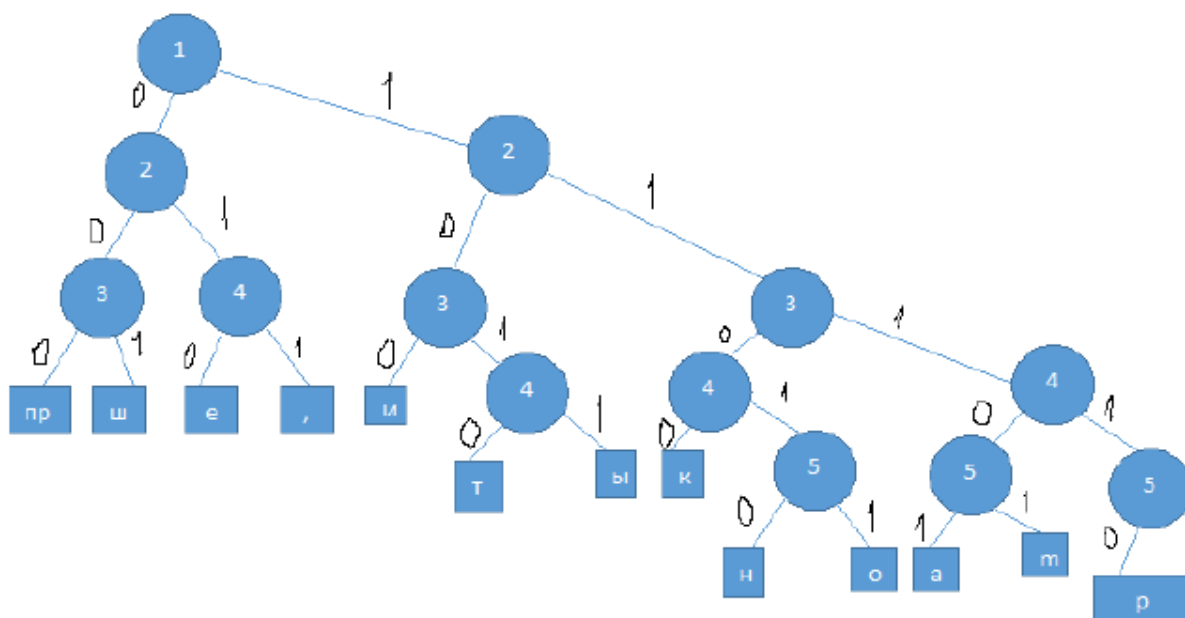
2) Оценка объема и коэффициента сжатия

Объем незакодированной фразы – $30 \cdot 8 \text{ бит} = 240 \text{ бит}$.

Объем закодированной фразы – 106 бит.

Коэффициент сжатия: $106/240=0,44$

3) Дерево префиксного кода Фано



2. По методу Хаффмана

Провести кодирование(сжатие) исходной строки символов «Фамилия Имя Отчество» с использованием алгоритма Хаффмана. Исходная строка символов, таким образом, определяет индивидуальный вариант задания для каждого студента.

Для выполнения задания 3 необходимо выполнить следующие действия:

Исходная строка примера **пупкин василий кириллович**

1. Отобразить таблицу частот текста

Пример таблицы, построенной по тексту **пупкин василий кириллович**

Алфавит	п	у	к	и	н	« »	в
Кол. вх.	2	1	2	6	1	2	2
Вероятн.	0.08	0.04	0.08	0.24	0.04	0.08	0.08
Алфавит	а	с	л	й	р	о	ч
Кол. вх.	1	1	3	1	1	1	1
Вероятн.	0.04	0.04	0.12	0.04	0.04	0.04	0.04

(« » обозначают пробел в исходной строке)

Вероятность появления символа: количество вхождений символа в текст/количество символов в тексте. Например, символ п входит 2 раза, всего символов 25, $2/25=0,08$.

2. Отобразить таблицу данных, отсортированная по убыванию частот

Алфавит	и	л	п	к	« »	в	у
Кол. вх.	6	3	2	2	2	2	1
Вероятн.	0.24	0.12	0.08	0.08	0.08	0.08	0.04

Алфавит	н	а	с	й	р	о	ч
Кол. вх.	1	1	1	1	1	1	1
Вероятн.	0.04	0.04	0.04	0.04	0.04	0.04	0.04

3. Построить и отобразить дерево кодирования Хаффмана, в данном примере оно имеет вид:

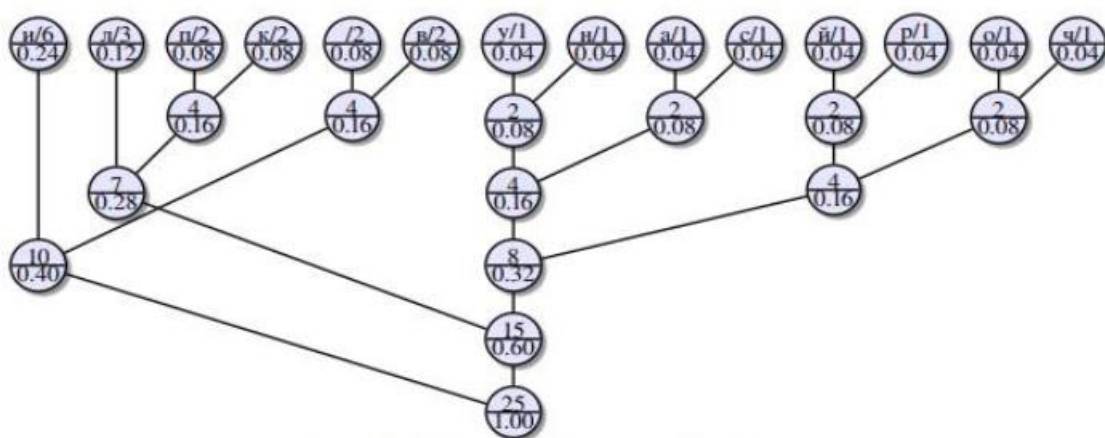


Рисунок. Дерево кодирования Хаффмана (указаны частоты)

4. Упорядочить построенное дерево слева-направо (при необходимости).

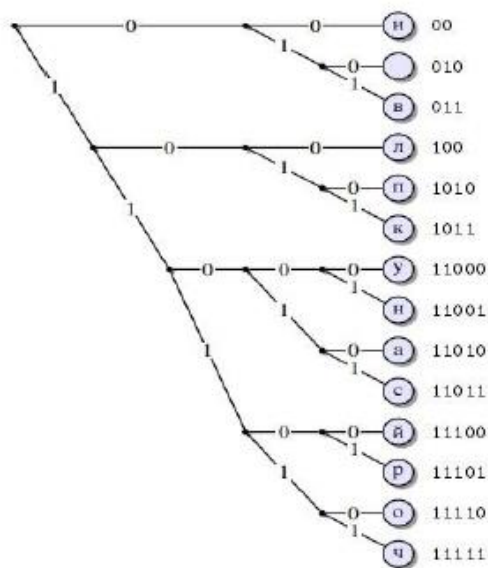


Рисунок. Упорядоченное дерево кодирования Хаффмана

5. Провести кодирование исходной строки по аналогии с примером:

п у п к и н « » в а с и л и й
 1010 11000 1010 1011 00 11001 010 011 11010 11011 00 100 00 11100
 « » к и р н л л о в и ч
 010 1011 00 11101 00 100 100 11110 011 00 11111

6. Рассчитать коэффициенты сжатия относительно кодировки ASCII и относительно равномерного кода.