

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

УТВЕРЖДАЮ
Зам. директора по УР
Н.А. Андреева
«22» 04 2024 г.

Б1.В.03 Структуры и алгоритмы обработки данных рабочая программа дисциплины

Кафедра **Информатики и вычислительной техники**
Направление подготовки **09.03.01. Информатика и вычислительная техника**
Профиль **Искусственный интеллект и машинное обучение**
Формы обучения **очная, заочная**

Распределение часов дисциплины по семестрам (для очной формы обучения (ОФО), курсам (для заочной формы обучения (ЗФО))

Вид учебной работы	ОФО		ЗФО	
	ЗЕ	часов/сем.	ЗЕ	часов/курс
Общая трудоемкость дисциплины, в том числе (по семестрам, курсам):	7	108/4 144/5	7	36/2 216/3
Контактная работа, в том числе (по семестрам, курсам):		64/4 72/5		10/2 28/3
Лекции		32/4 36/5		6/2 10/3
Лабораторные работы		16/4 36/5		14/3
Практические занятия		16/4		4/2 4/3
Семинарские занятия				
Самостоятельная работа студентов		44/4 36/5		26/2 179/3
Контроль		36/5		9/3
Число контрольных работ (по семестрам, курсам)				1/3
Число КР (по семестрам, курсам)				
Число КП (по семестрам, курсам)				
Число зачетов (с разбивкой по семестрам, курсам)		1/4		1/3
Число экзаменов (с разбивкой по семестрам, курсам)		1/5		1/3

Программу составил:

ст. преподаватель кафедры ИВТ Осмоловская Н.С.

Рабочая программа дисциплины

«Структуры и алгоритмы обработки данных»

Разработана в соответствии с ФГОС ВО

направления подготовки 09.03.01. Информатика и вычислительная техника,
утвержденным приказом Министерства образования и науки Российской Федерации
от 19 сентября 2017 г. № 929 .


Составлена на основании учебного плана

направления 09.03.01. Информатика и вычислительная техника профиля
«Искусственный интеллект и машинное обучение», одобренного Учёным советом СКФ
МТУСИ, протокол № 9 от 22.04.2024 г., и утвержденного директором СКФ МТУСИ
22.04.2024 г.

Рассмотрена и одобрена на заседании кафедры

«Информатика и вычислительная техника»

Протокол от «22» апреля 2024 г. № 9

Зав. кафедрой  Соколов С.В.

1. Цели изучения дисциплины

Целью освоения дисциплины «Структуры и алгоритмы обработки данных» является теоретическая и практическая подготовка, которая должна обеспечить получение у студентов знаний и навыков по организации структур данных и алгоритмах обработки данных.

2. Планируемые результаты обучения

Изучение дисциплины направлено на формирование у обучающегося способности решать: профессиональные задачи в соответствии с проектной деятельностью.

Результатом освоения дисциплины являются сформированные у обучающегося следующие компетенции:

Компетенции обучающегося, формируемые в результате освоения дисциплины (в части, обеспечиваемой дисциплиной)	
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения	
Знать:	
- основные языки программирования и языки работы с базами данных; - операционные системы и оболочки; - современные среды разработки программного обеспечения.	
Уметь:	
- составлять алгоритмы; - писать и отлаживать коды на языке программирования; - тестировать работоспособность программы; - интегрировать программные модули.	
Владеть:	
- языком программирования; - методами отладки и тестирования работоспособности программы.	

3. Место дисциплины в структуре образовательной программы

Требования к предварительной подготовке обучающегося предшествующие дисциплины, модули, темы):	
1	Б1.О.08. Информационные технологии и программирование
2	Б1.О.10. Математические основы баз данных
3	Б1.О.11. Дискретная математика
4	Б1.О.26. Введение в информационные технологии
Последующие дисциплины и практики, для которых освоение данной дисциплины необходимо:	
1	Б1.В.10. Функциональное программирование
2	Б1.О.14. Управление ИТ-проектами
3	Б1.В.ДВ.04.02. Математическое моделирование
4	Б2.В.02(П). Производственная практика (системы искусственного интеллекта)
5	Б3.01. Выполнение и защита выпускной квалификационной работы

4. Структура и содержание дисциплины

4.1 Очная форма обучения, 4 года (всего 252 часа, из них 136 часов аудиторных)

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компетенции	УМИО
1	2	3	4	5	6
Курс 2, Семестр 4					
Модуль 1 – Типовые алгоритмы обработки данных. Типы данных – 58 36 (16 Лек+12 ЛР+8 ПЗ) часов контактной работы – 22 часа СРС					
1.1	Лекция 1. Алгоритмы. Анализ сложности алгоритма. Алгоритмы сортировки. Рекурсивные алгоритмы. Реализация основных алгоритмов.	Лек.	4	ОПК-8	Л1.1 Л1.2 Л3.1
1.2	Практическое занятие 1. Анализ сложности и эффективности алгоритма. Составление эффективного программного кода.	ПЗ	2	ОПК-8	Л2.2 Л2.3 Л3.1
1.3	Массивы. Указатели. Связь указателей и массивов. Динамические массивы. Указатели и параметры функций. Работа с указателями и массивами в Python. Библиотека Stypes.	СРС	4	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.4	Массивы. Указатели. Связь указателей и массивов. Динамические массивы. Указатели и параметры функций. Язык C++.	СРС	4	ОПК-8	Л1.1 Л1.2 Л3.1
1.5	Лабораторная работа 1. Массивы. Указатели. Связь указателей и массивов. Динамические массивы. Указатели и параметры функций.	ЛР	4	ОПК-8	Л2.2 Л2.3 Л3.1
1.6	Лекция 2. Абстрактные типы данных (АТД). Примеры. Создание АТД. Списки. Основные операции. Односвязный список. Реализация односвязных списков на основе массивов.	Лек.	4	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.7	Практическое занятие 2. Создание и демонстрация работы абстрактных типов данных (АТД).	ПЗ	2	ОПК-8	Л2.2 Л2.3
1.8	Создание и демонстрация работы абстрактных типов данных	СРС	6	ОПК-8	Л1.1 Л1.2
1.9	Лекция 3. Списки. Односвязный список с динамическим распределением памяти. Двусвязный список. Задача "Word Count"(алфавитно-частотный словарь).	Лек.	4	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.10	Практическое занятие 3. Реализация задачи "Word Count"(алфавитно-частотный словарь). Подсчёт символов, слов и строк.	ПЗ	2	ОПК-8	Л2.2 Л2.3 Л3.1
1.11	Реализация задачи "Word Count"(алфавитно-частотный словарь). Работа с файлами.	СРС	4	ОПК-8	Л1.1 Л1.2
1.12	Лабораторная работа 2. Реализация задачи "Word Count"(алфавитно-частотный словарь). Работа с файлами. Подсчет символов в строке с учетом регистра. Подсчет подстрок в строке с учетом перекрытий	ЛР	4	ОПК-8	Л2.2 Л2.3 Л3.1
1.13	Лекция 4. Стек. Очередь. Понятие очереди с приоритетом. Реализация стека на статическом массиве. Реализация стека на динамическом	Лек.	4	ОПК-8	Л1.1 Л1.2 Л3.1

	массиве. Реализация стека на шаблоне класса. Детали реализации. Программная реализация очереди. Удаление элемента. Чтение элемента из очереди. Проверка на пустоту. Проверка на наполненность. Дек. Куча. Двоичная куча. Поддержание указателей на элементы кучи. Биномиальные (binomial), левацкие (leftlist) и косые (skew) кучи.				ЛЗ.2
1.14	Практическое занятие 4. Реализация стека. Реализация очереди.	ПЗ	2	ОПК-8	Л2.2 Л2.3
1.15	Реализация дека с помощью двусвязного списка. Примеры использования.	СРС	4	ОПК-8	Л1.1 Л1.2
1.16	Лабораторная работа 3. Реализация дека с помощью двусвязного списка. Реализация очереди.	ЛР	4	ОПК-8	Л2.2 Л2.3
Модуль 2 – Структура данных дерево и алгоритмы на деревьях – 50 28 (16 Лек+4 ЛР+8 ПЗ) часов контактной работы – 22 часа СРС					
2.1	Лекция 5. Хэширование. Прямая адресация. Хэш-таблицы. Хэш-функции. Контроль целостности данных.	Лек.	4	ОПК-8	Л1.1 Л1.2 ЛЗ.1
2.2	Лекция 6. Криптография. Поисковые системы. Управление кэшем. Виды хэширования.	Лек.	4	ОПК-8	Л1.1 Л1.2
2.3	Практическое занятие 5. Алгоритмы хеширования данных. Построение функций хеширования и алгоритмов хеширования данных, при решении задач на языке C++	ПЗ	2	ОПК-8	Л2.2 Л2.3 ЛЗ.1
2.4	Реализация алгоритмов хэширования. Хеширование в структуре данных. Двойное хеширование. Универсальное хэширование.	СРС	6	ОПК-8	Л1.1 Л1.2 ЛЗ.1
2.5	Лабораторная работа 4. Реализация алгоритмов хэширования. Составление хеш-таблиц. Построение хеш-таблиц из текстового файла. Построение хеш-таблиц для зарезервированных слов, используемого языка программирования.	ЛР	2	ОПК-8	Л2.2 Л2.3 ЛЗ.1 ЛЗ.2
2.6	Лекция 7. Деревья. Терминология. Виды деревьев. Рекурсивное определение дерева. Обходы дерева. Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Очереди с приоритетом.	Лек.	4	ОПК-8	Л1.1 Л1.2 ЛЗ.1 ЛЗ.2
2.7	Практическое занятие 6. Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Реализация алгоритма построения оптимального бинарного дерева поиска	ПЗ	2	ОПК-8	Л2.2 Л2.3 ЛЗ.1
2.8	Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Оптимальные бинарные деревья поиска. Алгоритм поиска оптимального дерева.	СРС	8	ОПК-8	Л1.1 Л1.2 ЛЗ.1 ЛЗ.2
2.9	Лабораторная работа 5. Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Реализация алгоритма построения оптимального бинарного дерева поиска с квадратичным временем работы.	ЛР	2	ОПК-8	Л2.2 Л2.3 ЛЗ.1 ЛЗ.2
2.10	Лекция 8. Сбалансированные деревья. AVL-деревья. Основные операции. Реализация.	Лек.	4	ОПК-8	Л1.1 Л1.2

	Включение новой вершины в AVL-дерево. Удаление вершины.				ЛЗ.1 ЛЗ.2
2.11	Программная реализация AVL-дерева. Уменьшение количества дополнительной информации в AVL-дереве до двух бит на вершину. Выполнение на строго типизированном функциональном языке реализацию добавления в AVL дерево так, чтобы AVL-инвариант гарантировался типизацией.	СРС	8		
2.12	Практическое занятие 7. Программная реализация AVL-дерева. Построение AVL-дерева, которое будет, являться объединением деревьев T1 и T2.	ПЗ	4	ОПК-8	Л2.2 Л2.3 ЛЗ.1 ЛЗ.2
	Зачет			ОПК-8	Л1.1 Л1.2 ЛЗ.1 ЛЗ.2
Итого			108		
Курс 3, Семестр 5					
Модуль 1 – Динамические структуры данных – 48					
32 (16 Лек+16 ЛР) часов контактной работы – 16 часов СРС					
1.1	Лекция 1. Красно-черные деревья. Свойства красно-черных деревьев. Вращение. Добавление и удаление вершин. Дерево промежутков.	Лек.	4	ОПК-8	Л1.1 Л1.2 ЛЗ.1
1.2	Красно-черные деревья. Свойства красно-черных деревьев. Вращение. Добавление и удаление вершин. Дерево промежутков. Программная реализация красно-черных деревьев.	СРС	4	ОПК-8	Л1.1 Л1.2 ЛЗ.1 ЛЗ.2
1.3	Лабораторная работа 1. Программная реализация красно-черных деревьев. Добавление и удаление вершин. Дерево промежутков.	ЛР	4	ОПК-8	Л2.2 Л2.3 ЛЗ.1
1.4	Лекция 2. SPLAY-деревья – само перестраивающиеся деревья. Основные операции. Словарные операции. Алгоритм вставки узла в само перестраивающееся дерево. Вспомогательные процедуры для работы с указателями на родителей. Процедура splay. Процедура split. Процедура merge. Процедура find.	Лек.	4	ОПК-8	Л1.1 Л1.2 ЛЗ.1 ЛЗ.2
1.5	Программная реализация SPLAY-деревьев. Основные операции. Вспомогательные процедуры для работы с указателями на родителей. Процедуры: splay, split, merge, find.	СРС	4	ОПК-8	Л1.1 Л1.2 ЛЗ.1 ЛЗ.2
1.6	Лабораторная работа 2. Программная реализация SPLAY-деревьев. Построение двух корректных дерева поиска, построенные на одном и том же множестве ключей друг из друга последовательностью поворотов.	ЛР	4	ОПК-8	Л2.2 Л2.3 ЛЗ.1

1.7	Лекция 3. Б-деревья. Вращение. Добавление и удаление вершин. Применение. Исправление потенциальной разбалансировки на этапе поиска удаляемого элемента в дереве.	Лек.	4	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.8	Программная реализация Б-деревьев. Создание бинарного дерева поиска. Поиск образца в бинарном дереве. Подсчет количества узлов дерева. Подсчет числа листьев дерева. Удаление всех узлов из бинарного дерева	СРС	4	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.9	Лабораторная работа 3. Программная реализация Б-деревьев. Создание бинарного дерева поиска. Поиск образца в бинарном дереве. Подсчет количества узлов дерева. Подсчет числа листьев дерева. Удаление всех узлов из бинарного дерева	ЛР	4	ОПК-8	Л2.2 Л2.3 Л3.1
1.10	Лекция 4. Биномиальные деревья и кучи. Основные операции. Свойства биномиальных деревьев. Операции с биномиальной кучей. Куча Фибоначчи. Фибоначчиевы деревья.	Лек.	4	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.11	Программная реализация операций над биномиальными деревьями. Реализация операции формирования кучи из множества элементов. Визуализация кучи при выполнении операций.	СРС	4	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.12	Лабораторная работа 4. Разработка класс для структуры данных биномиальной куча. Реализация операции формирования кучи из множества элементов. Визуализация кучи.	ЛР	4	ОПК-8	Л2.2 Л2.3 Л3.1
Модуль 2 – Сложные структуры данных – 60 40 (20 Лек+20 ЛР) часов контактной работы – 20 часов СРС					
2.1	Лекция 5. Графы. Описание. Основные понятия и виды графов. Задачи на графах. Максимальное дерево и цикломатическое число.	Лек.	4	ОПК-8	Л1.1 Л1.2 Л3.1
2.2	Графы. Описание. Основные понятия и виды графов. Задачи на графах. Максимальное дерево и цикломатическое число. Теорема Рамсея. Изоморфизм графов.	СРС	4	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
2.3	Лабораторная работа 5. Задачи на графах. Построение матрицы смежности. Построение матрицы инцидентий.	ЛР	4	ОПК-8	Л2.2 Л2.3 Л3.1
2.4	Лекция 6. Системы непересекающихся множеств. Основные свойства и операции. Реализация на основе списков (или массива). Реализация на основе леса. Операции над частично упорядоченными множествами.	Лек.	6	ОПК-8	Л1.1 Л1.2 Л3.1
2.5	Системы непересекающихся множеств. Основные свойства и операции. Реализация на основе списков (или массива). Реализация на основе леса. Операции над частично упорядоченными множествами.	СРС	6	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
2.6	Лабораторная работа 6. Реализация с использованием списков. Построение списков связи. Построение перечней ребер.	ЛР	6	ОПК-8	Л2.2 Л2.3 Л3.1

2.7	Лекция 7. Алгоритмы на графах. Поиск в ширину. Поиск в глубину. Минимальные покрывающие деревья. Взвешенные графы. Алгоритмы поиска кратчайшего пути на взвешенном графе.	Лек.	6	ОПК-8	Л1.1 Л1.2 Л3.1
2.8	Взвешенные графы. Алгоритмы поиска кратчайшего пути на взвешенном графе: Дейкстры, Беллмана – Форда, Левита, Флойда – Уоршелла, Краскала.	СРС	6	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
2.9	Лабораторная работа 7. Программная реализация алгоритмов на графах. Построение остовного дерева путем обхода в глубину (в ширину). Построение матрицы кратчайших путей.	ЛР	6	ОПК-8	Л2.2 Л2.3 Л3.1
2.10	Лекция 8. Цифровой поиск. Хеширование с открытой адресацией. Хеш-функции для открытой адресации. Хеширование других данных. Обобщения цифрового поиска. Комбинации нескольких методов: цифровой поиск и поиск в последовательных таблицах.	Лек.	4	ОПК-8	Л1.1 Л1.2 Л3.1
2.11	Цифровой поиск. Хеширование с открытой адресацией. Хеш-функции для открытой адресации. Хеширование других данных. Обобщения цифрового поиска. Комбинации нескольких методов: цифровой поиск и поиск в последовательных таблицах.	СРС	4	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
2.12	Лабораторная работа 8. Обработка словаря с высечками. Использование многоуровневых высечек.	ЛР	4	ОПК-8	Л2.2 Л2.3 Л3.1
	Экзамен		36	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
Итого			144		

4.2. Заочная форма обучения, 5 лет (всего 252 часа, из них 38 часов аудиторных)

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компетенции	УМИО
1	2	3	4	5	6
Курс 2 сессия 3					
Модуль 1 – Типовые алгоритмы обработки данных. Типы данных – 36 10 (6 Лек + 4 ПЗ) часов контактной работы – 26 часов СРС					
1.1	Лекция 1. Алгоритмы. Анализ сложности алгоритма. Алгоритмы сортировки. Рекурсивные алгоритмы. Реализация основных алгоритмов.	Лек.	2	ОПК-8	Л1.1 Л1.2 Л3.1
1.2	Практическое занятие 1. Составление эффективного программного кода. Массивы. Указатели. Связь указателей и массивов. Динамические массивы. Указатели и параметры функций.	ПЗ	2	ОПК-8	Л1.1 Л1.2 Л2.2 Л2.3 Л3.1
1.3	Массивы. Указатели. Связь указателей и массивов. Динамические массивы. Указатели и параметры	СРС	6	ОПК-8	Л1.1 Л1.2

	функций. Работа с указателями и массивами в Python. Библиотека Ctypes.				Л3.1 Л3.2
1.4	Массивы. Указатели. Связь указателей и массивов. Динамические массивы. Указатели и параметры функций. Язык C++.	СРС	6	ОПК-8	Л1.1 Л1.2 Л3.1
1.5	Лекция 2. Абстрактные типы данных (АТД). Примеры. Создание АТД. Списки. Основные операции. Односвязный список. Реализация односвязных списков на основе массивов. Списки. Односвязный список с динамическим распределением памяти. Двусвязный список. Задача "Word Count"(алфавитно-частотный словарь).	Лек.	2	ОПК-8	Л1.1 Л1.2 Л2.2 Л2.3 Л2.4 Л3.1 Л3.2
1.6	Практическое занятие 2. Создание и демонстрация работы абстрактных типов данных (АТД). Реализация задачи "Word Count"(алфавитно-частотный словарь). Работа с файлами. Подсчет символов в строке с учетом регистра. Подсчет подстрок в строке с учетом перекрытий	ПЗ	2	ОПК-8	Л1.1 Л1.2 Л2.2 Л2.3 Л2.4 Л3.1
1.7	Создание и демонстрация работы абстрактных типов данных Реализация задачи "Word Count" (алфавитно-частотный словарь). Работа с файлами.	СРС	8	ОПК-8	Л1.1 Л1.2
1.8	Лекция 3. Стек. Очередь. Понятие очереди с приоритетом. Реализация стека на статическом массиве. Реализация стека на динамическом массиве. Реализация стека на шаблоне класса. Детали реализации. Программная реализация очереди. Удаление элемента. Чтение элемента из очереди. Проверка на пустоту. Проверка на наполненность. Дек. Куча. Двоичная куча. Поддержание указателей на элементы кучи. Биномиальные (binomial), левацкие (leftlist) и косые (skew) кучи.	Лек.	2	ОПК-8	Л1.1 Л1.2 Л2.2 Л2.3 Л3.1 Л3.2
1.9	Реализация дека с помощью двусвязного списка. Примеры использования. Программная реализация очереди.	СРС	6	ОПК-8	Л1.1 Л1.2
	Итого		36		
Курс 3 сессия 2					
Модуль 2 – Структура данных дерево и алгоритмы на деревьях – 108 14 (4 Лек+6 ЛР+4 ПЗ) часов контактной работы – 94 часа СРС					
2.1	Лекция 4. Хэширование. Прямая адресация. Хэш-таблицы. Хэш-функции. Контроль целостности данных. Криптография. Поисковые системы. Управление кэшем. Виды хэширования.	Лек.	2	ОПК-8	Л1.1 Л1.2 Л2.2 Л2.3
2.2	Практическое занятие 3. Алгоритмы хеширования данных. Построение функций хеширования и алгоритмов хеширования данных, при решении задач на языке C++	ПЗ	2	ОПК-8	Л2.2 Л2.3 Л3.1
2.3	Реализация алгоритмов хэширования. Хеширование в структуре данных. Двойное хеширование. Универсальное хэширование.	СРС	30	ОПК-8	Л1.1 Л1.2 Л3.1

2.4	Лабораторная работа 1. Реализация алгоритмов хэширования. Составление хеш-таблиц. Построение хеш-таблиц из текстового файла. Построение хеш-таблиц для зарезервированных слов, используемого языка программирования.	ЛР	2	ОПК-8	Л2.2 Л2.3 Л3.1 Л3.2
2.5	Лекция 5. Деревья. Терминология. Виды деревьев. Рекурсивное определение дерева. Обходы дерева. Бинарные деревья поиска. Сбалансированные деревья. AVL-деревья.	Лек.	2	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
2.6	Практическое занятие 4. Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Реализация алгоритма построения оптимального бинарного дерева поиска	ПЗ	2	ОПК-8	Л2.2 Л2.3 Л3.1
2.7	Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Оптимальные бинарные деревья поиска. Алгоритм поиска оптимального дерева.	СРС	32	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
2.8	Лабораторная работа 2. Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Реализация алгоритма построения оптимального бинарного дерева поиска с квадратичным временем работы.	ЛР	4	ОПК-8	Л2.2 Л2.3 Л3.1 Л3.2
2.9	Программная реализация AVL-дерева. Уменьшение количества дополнительной информации в AVL-дереве до двух бит на вершину. Выполнение на строго типизированном функциональном языке реализацию добавления в AVL дерево так, чтобы AVL-инвариант гарантировался типизацией.	СРС	32	ОПК-8	Л1.1 Л1.2 Л2.2 Л2.3 Л3.1 Л3.2
	Зачет			ОПК-8	Л1.1 Л1.2 Л2.2 Л2.3 Л3.1 Л3.2
Итого			108		

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компетенции	УМИО
1	2	3	4	5	6
Курс 3 сессия 3					
Модуль 1 – Сложные структуры данных – 78 8 (4 Лек+4 ЛР) часов контактной работы – 70 часов СРС					
1.1	Лекция 1. Красно-черные деревья. Свойства красно-черных деревьев. Вращение. Добавление и удаление вершин. Дерево промежутков.	Лек.	2	ОПК-8	Л1.1 Л1.2 Л3.1
1.2	Красно-черные деревья. Свойства красно-черных деревьев. Вращение. Добавление и удаление вершин. Дерево промежутков. Программная реализация красно-черных деревьев.	СРС	20	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.3	Лабораторная работа 1. Программная реализация деревьев. Добавление и удаление вершин. Создание бинарного дерева поиска. Поиск образца в бинарном дереве.	ЛР	2	ОПК-8	Л2.2 Л2.3 Л3.1
1.4	Лекция 2. SPLAY-деревья. Процедуры: splay, split, merge, find. Б-деревья. Вращение. Добавление и удаление вершин. Применение. Биномиальные деревья и кучи. Куча Фибоначчи.	Лек.	2	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.5	Программная реализация SPLAY-деревьев. Основные операции. Вспомогательные процедуры для работы с указателями на родителей. Процедуры: splay, split, merge, find.	СРС	20	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.6	Программная реализация Б-деревьев. Создание бинарного дерева поиска. Поиск образца в бинарном дереве. Подсчет количества узлов дерева. Подсчет числа листьев дерева. Удаление всех узлов из бинарного дерева	СРС	15	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.7	Программная реализация операций над биномиальными деревьями. Реализация операции формирования кучи из множества элементов. Визуализация кучи при выполнении операций.	СРС	15	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
1.8	Лабораторная работа 2. Программная реализация SPLAY-деревьев. Построение двух корректных дерева поиска, построенные на одном и том же множестве ключей друг из друга последовательностью поворотов.	ЛР	2	ОПК-8	Л2.2 Л2.3 Л3.1
Модуль 2 – Структуры взаимосвязанных элементов – 21 6 (2 Лек+4 ЛР) часов контактной работы – 15 часов СРС					
2.1	Лекция 3. Графы. Описание. Основные понятия и виды графов. Задачи на графах. Максимальное дерево и цикломатическое число. Системы непересекающихся множеств. Основные свойства.	Лек.	2	ОПК-8	Л1.1 Л1.2 Л3.1
2.2	Графы. Описание. Основные понятия и виды графов. Задачи на графах. Максимальное дерево и цикломатическое число. Теорема Рамсея. Изоморфизм графов.	СРС	5	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
2.3	Лабораторная работа 3. Задачи на графах. Построение матрицы смежности. Построение матрицы инцидентий.	ЛР	2	ОПК-8	Л2.2 Л2.3 Л3.1

2.4	Системы непересекающихся множеств. Основные свойства и операции. Реализация на основе списков (или массива). Реализация на основе леса. Операции над частично упорядоченными множествами.	СРС	10	ОПК-8	Л1.1 Л1.2 Л3.1 Л3.2
2.5	Лабораторная работа 4. Реализация с использованием списков. Построение списков связи. Построение перечней ребер.	ЛР	2	ОПК-8	Л2.2 Л2.3 Л3.1
	Экзамен		9		
Итого			108		

5. Учебно-методическое и информационное обеспечение дисциплины

5.1 Рекомендуемая литература				
5.1.1. Основная литература				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л1.1	Т. О. Сундукова, Г. В. Ванькина.	Структуры и алгоритмы компьютерной обработки данных: учебное пособие	Москва, Саратов: ИНТУИТ, Ай Пи Ар Медиа, 2020	Э1
Л1.2	Г. С. Иванова, Т.Н. Ничушкина, Е. К. Пугачев.	Выбор алгоритмов обработки данных, тестирование и повышение качества программ: учебно-методическое пособие	Москва: МГТУ имени Н.Э. Баумана, 2020	Э2
5.1.2 Дополнительная литература				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л2.1	В. Т. Корниенко, С. Д. Ефимов	Реализация алгоритмов обработки данных в протоколах сетей связи: учебное пособие	Москва: Ай Пи Ар Медиа, 2025	Э3
Л2.2	Ю.М. Мартынюк	Структуры и алгоритмы компьютерной обработки данных: учебно-методическое пособие для проведения лабораторных работ	Тула: ТГПУ имени Л.Н. Толстого, 2021.	Э4
Л2.3	С.В. Засорин, О.А. Ломтева.	Структуры и алгоритмы компьютерной обработки данных: лабораторный практикум: учебное пособие	Москва: КУРС, 2024	Э5
Л2.4	М.М. Волков	Практикум по дисциплине: Структуры и алгоритмы обработки данных	Москва: МТУСИ, 2016	Э6
5.1.3 Учебно-методическое обеспечение				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л3.1	Лобзенко П.В., Щербань О.Г., Щербань И.В.	Учебное пособие для проведения лабораторных работ по дисциплине "Технологии языков программирования"	Ростов-на-Дону, СКФ МТУСИ, 2019 г.	Э7
Л3.2	В.В. Ландовский	Алгоритмы обработки данных: учебное пособие	Новосибирск: НГТУ, 2018	Э8
5.2 Электронные образовательные ресурсы				
Э1	https://www.iprbookshop.ru/89476.html			
Э2	https://www.iprbookshop.ru/115309.html			

Э3	https://www.iprbookshop.ru/149941.html
Э4	https://www.iprbookshop.ru/119695.html
Э5	https://www.iprbookshop.ru/144822.html
Э6	https://www.iprbookshop.ru/61551.html
Э7	https://elibr.mtuci.ru/view.php?book_id=170030
Э8	https://www.iprbookshop.ru/91316.html
5.3 Информационно-справочные системы и профессиональные базы данных	
5.3.1	Консультант Плюс – https://www.consultant.ru/
5.3.2	Гарант – https://www.garant.ru/
5.3.3	Консорциум Кодекс – https://docs.cntd.ru/
5.3.4	Научная электронная библиотека – https://www.elibrary.ru/
5.3.5	Профессиональная база данных: Федеральный портал «Российское образование»: [Электронный ресурс] – Режим доступа: http://www.edu.ru/ (открытый доступ)
5.3.6	Профессиональная база данных: Федеральный центр информационно-образовательных ресурсов: [Электронный ресурс] – Режим доступа: http://fcior.edu.ru/ (открытый доступ)
5.4 Программное обеспечение	
П.1	CodeBlocks Среда программирования на языке C/C++
П.2	IDLE Среда программирования на языке Python
П.3	LibreOffice Кроссплатформенный, свободно распространяемый офисный пакет
П.4	Яндекс браузер
П.6	Антивирус Касперского

6. Материально-техническое обеспечение дисциплины	
6.1 МТО лекционных занятий	
1	Лекционная аудитория, оснащенная проектором, ПК (ноутбуком), экраном
6.2 МТО лабораторных работ и практических занятий	
1	Учебные аудитории, оборудованные компьютерной техникой, с возможностью подключения к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ
6.3 МТО рубежных контролей, зачетов, экзаменов	
1	Учебные аудитории, оборудованные компьютерной техникой, с возможностью подключения к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ
6.4 МТО самостоятельной работы обучающихся	
1	Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ

7. Оценочные материалы	
Оценочные материалы и перечень видов оценочных средств текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.	
8. Методические рекомендации по освоению дисциплины	
Методические рекомендации по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.	

9. Особенности реализации дисциплины (модуля) при обучении инвалидов и лиц с ограниченными возможностями здоровья

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. При обучении лиц с ограниченными возможностями здоровья учитываются их индивидуальные психофизические особенности. Обучение инвалидов осуществляется также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).

Для лиц с нарушением слуха возможно предоставление учебной информации в визуальной форме (краткий конспект лекций; тексты заданий, напечатанные увеличенным шрифтом), на аудиторных занятиях допускается присутствие ассистента, а также сурдопереводчиков и тифлосурдопереводчиков.

Текущий контроль успеваемости осуществляется в письменной форме: обучающийся письменно отвечает на вопросы, письменно выполняет практические задания.

Доклад (реферат) также может быть представлен в письменной форме, при этом требования к содержанию остаются теми же, а требования к качеству изложения материала (понятность, качество речи, взаимодействие с аудиторией и т. д.) заменяются на соответствующие требования, предъявляемые к письменным работам (качество оформления текста и списка литературы, грамотность, наличие иллюстрационных материалов и т.д.).

Промежуточная аттестация для лиц с нарушениями слуха проводится в письменной форме, при этом используются общие критерии оценивания. При необходимости время подготовки к ответу может быть увеличено.

Для лиц с нарушением зрения допускается аудиальное предоставление информации, а также использование на аудиторных занятиях звукозаписывающих устройств (диктофонов и т.д.). Допускается присутствие на занятиях ассистента (помощника), оказывающего обучающимся необходимую техническую помощь.

Текущий контроль успеваемости осуществляется в устной форме.

При проведении промежуточной аттестации для лиц с нарушением зрения тестирование может быть заменено на устное собеседование по вопросам.

Для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, на аудиторных занятиях, а также при проведении процедур текущего контроля успеваемости и промежуточной аттестации могут быть предоставлены необходимые технические средства (персональный компьютер, ноутбук или другой гаджет); допускается присутствие ассистента (ассистентов), оказывающего обучающимся необходимую техническую помощь (занять рабочее место, передвигаться по аудитории, прочитать задание, оформить ответ, общаться с преподавателем).

10. Дополнения и изменения в Рабочей программе

Лист актуализации рабочей программы дисциплины «Б1.В.03 Структуры и алгоритмы обработки данных» для использования в 20___/20___ учебном году

Утверждаю

Зам. директора по УР _____
«__» _____ 20__ г.

Направление: 09.03.01. Информатика и вычислительная техника

Профиль: Искусственный интеллект и машинное обучение

Форма обучения: очная, заочная

(Возможны следующие варианты):

а) Рабочая программа действует без изменений.

б) В рабочую программу вносятся следующие изменения:

1)

2)

3)

Разработчик (и): _____
(ФИО, ученая степень, ученое звание)

«__» _____ 20__ г.

Рабочая программа пересмотрена и одобрена на заседании кафедры _____

Протокол № _____ от «__» _____ 20__ г.

Заведующий кафедрой _____

Оценочные материалы для проведения текущего контроля и промежуточной аттестации по дисциплине

1. Оценочные материалы для проведения промежуточной аттестации по дисциплине

1.1 Шкала оценивания компетенций

Шкала оценивания компетенций		
Оценка	Уровень освоения компетенции	Критерии оценивания
«Отлично»	Высокий уровень	Обучающийся показывает всестороннее, систематическое и глубокое знание основного и дополнительного учебного материала, дает полные, профессиональные и грамотные ответы на поставленные вопросы, что свидетельствует об освоении обучающимся взаимосвязи основных понятий дисциплины; усвоил основную и знаком с дополнительной рекомендованной литературой; проявляет творческие способности в понимании, изложении и использовании учебно-программного материала.
«Хорошо»	Повышенный уровень	Обучающийся показывает достаточный уровень знания основного и дополнительного учебного материала, дает в целом грамотные ответы на поставленные вопросы, что свидетельствует об освоении обучающимся взаимосвязи основных понятий дисциплины; усвоил основную и знаком с дополнительной рекомендованной литературой. Допускает не существенные погрешности в ответах, устраняет их без помощи преподавателя.
«Удовлетворительно»	Пороговый уровень	Обучающийся демонстрирует знание основного учебного материала в минимальном объеме, необходимом для дальнейшей учебы; в целом отвечает на поставленные вопросы, допуская не принципиальные ошибки; знаком с основной литературой. Допускает ошибки в ответах различного уровня, которые исправляет после наводящих вопросов и под руководством преподавателя.
«Неудовлетворительно»	Минимальный уровень не достигнут	Обучающийся обнаруживает существенные пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении, предусмотренных программой, заданий; не знаком с рекомендованной литературой, не может исправить допущенные ошибки. Как правило, оценка «неудовлетворительно» ставится обучающимся, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине.

1.2 Показатели, критерии и шкалы оценивания компетенций

1.2.1 Показатели, критерии и шкалы оценивания компетенций для очной формы обучения (ОФО) (Курс 2, Семестр 4)

Показатели компетенции	Критерии оценивания	Шкала оценивания
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения		
Знать:		
<ul style="list-style-type: none"> – основные языки программирования и языки работы с базами данных; – операционные системы и оболочки; 	Контрольная работа	Модуль 1 0÷15 «Неудовлетворительно» - 0-7 «Удовлетворительно» - 8-10 «Хорошо» - 11-12 «Отлично» - 13-15
<ul style="list-style-type: none"> – современные среды разработки программного обеспечения. 	Контрольная работа	Модуль 2 0÷15 «Неудовлетворительно» - 0-7 «Удовлетворительно» - 8-10 «Хорошо» - 11-12 «Отлично» - 13-15
Уметь:		
<ul style="list-style-type: none"> – составлять алгоритмы; – писать и отлаживать коды на языке программирования; 	Лабораторная работа 1 Практические занятия 1-2	Модуль 1 0÷5 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3 «Хорошо» - 4 «Отлично» - 5
<ul style="list-style-type: none"> – тестировать работоспособность программы; – интегрировать программные модули. 	Лабораторная работа 4 Практические занятия 5-6	Модуль 2 0÷7 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3 «Хорошо» - 4-5 «Отлично» - 6-7
Владеть:		
<ul style="list-style-type: none"> – языком программирования; 	Лабораторные работы 2-3 Практические занятия 3-4	Модуль 1 0÷5 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3 «Хорошо» - 4 «Отлично» - 5
<ul style="list-style-type: none"> – методами отладки и тестирования работоспособности программы. 	Лабораторная работа 5 Практическое занятие 7	Модуль 2 0÷7 «Неудовлетворительно» - 0-2 «Удовлетворительно» - 3 «Хорошо» - 4-5 «Отлично» - 6-7
	Зачет	0-100 «Не зачтено» - 0-40 «Зачтено» - 41-100

1.2.2 Показатели, критерии и шкалы оценивания компетенций для очной формы обучения (ОФО) (Курс 3, Семестр 5)

Показатели компетенции	Критерии оценивания	Шкала оценивания
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения		
Знать:		
<ul style="list-style-type: none"> – основные языки программирования и языки работы с базами данных; – операционные системы и оболочки; 	Контрольная работа	Модуль 1 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
<ul style="list-style-type: none"> – современные среды разработки программного обеспечения. 	Контрольная работа	Модуль 2 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
Уметь:		
<ul style="list-style-type: none"> – составлять алгоритмы; – писать и отлаживать коды на языке программирования; 	Лабораторные работы 1-2	Модуль 1 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
<ul style="list-style-type: none"> – тестировать работоспособность программы; – интегрировать программные модули. 	Лабораторные работы 5-6	Модуль 2 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
Владеть:		
<ul style="list-style-type: none"> – языком программирования; 	Лабораторные работы 3-4	Модуль 1 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
<ul style="list-style-type: none"> – методами отладки и тестирования работоспособности программы. 	Лабораторные работы 7-8	Модуль 2 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
	Экзамен	0-100 «Неудовлетворительно» - 0-40 «Удовлетворительно» - 41-60 «Хорошо» - 61-80 «Отлично» - 81-100

1.2.3 Показатели, критерии и шкалы оценивания компетенций для заочной формы обучения (ЗФО) (Курс 2 сессия 3 и Курс 3 сессия 2)

Показатели компетенции	Критерии оценивания	Шкала оценивания
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения		
Знать:		
<ul style="list-style-type: none"> – основные языки программирования и языки работы с базами данных; – операционные системы и оболочки; 	Контрольная работа	Модуль 1 0÷20 «Неудовлетворительно» - 0-13 «Удовлетворительно» - 14-15 «Хорошо» - 16-17 «Отлично» - 18-20
<ul style="list-style-type: none"> – современные среды разработки программного обеспечения. 	Контрольная работа	Модуль 2 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
Уметь:		
<ul style="list-style-type: none"> – составлять алгоритмы; – писать и отлаживать коды на языке программирования; 	Практическое занятие 1	Модуль 1 0÷15 «Неудовлетворительно» - 5-7 «Удовлетворительно» - 8-10 «Хорошо» - 11-12 «Отлично» - 13-15
<ul style="list-style-type: none"> – тестировать работоспособность программы; – интегрировать программные модули. 	Практические занятия 3-4	Модуль 2 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
Владеть:		
<ul style="list-style-type: none"> – языком программирования; 	Практическое занятие 2	Модуль 1 0÷15 «Неудовлетворительно» - 5-7 «Удовлетворительно» - 8-10 «Хорошо» - 11-12 «Отлично» - 13-15
<ul style="list-style-type: none"> – методами отладки и тестирования работоспособности программы. 	Лабораторные работы 1-2	Модуль 2 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
	Зачет	0-100 «Не зачтено» - 0-40 «Зачтено» - 41-100

1.2.3 Показатели, критерии и шкалы оценивания компетенций для заочной формы обучения (ЗФО) (Курс 3 сессия 3)

Показатели компетенции	Критерии оценивания	Шкала оценивания
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения		
Знать:		
<ul style="list-style-type: none"> – основные языки программирования и языки работы с базами данных; – операционные системы и оболочки; 	Контрольная работа	Модуль 1 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
<ul style="list-style-type: none"> – современные среды разработки программного обеспечения. 	Контрольная работа	Модуль 2 0÷10 «Неудовлетворительно» - 0-3 «Удовлетворительно» - 4-5 «Хорошо» - 6-7 «Отлично» - 8-10
Уметь:		
<ul style="list-style-type: none"> – составлять алгоритмы; – писать и отлаживать коды на языке программирования; 	Лабораторная работа 1	Модуль 1 0÷20 «Неудовлетворительно» - 11-13 «Удовлетворительно» - 14-15 «Хорошо» - 16-17 «Отлично» - 18-20
<ul style="list-style-type: none"> – тестировать работоспособность программы; – интегрировать программные модули. 	Лабораторная работа 3	Модуль 2 0÷20 «Неудовлетворительно» - 11-13 «Удовлетворительно» - 14-15 «Хорошо» - 16-17 «Отлично» - 18-20
Владеть:		
<ul style="list-style-type: none"> – языком программирования; 	Лабораторная работа 2	Модуль 1 0÷20 «Неудовлетворительно» - 11-13 «Удовлетворительно» - 14-15 «Хорошо» - 16-17 «Отлично» - 18-20
<ul style="list-style-type: none"> – методами отладки и тестирования работоспособности программы. 	Лабораторная работа 4	Модуль 2 0÷20 «Неудовлетворительно» - 11-13 «Удовлетворительно» - 14-15 «Хорошо» - 16-17 «Отлично» - 18-20
	Экзамен	0-100 «Неудовлетворительно» - 0-40 «Удовлетворительно» - 41-60 «Хорошо» - 61-80 «Отлично» - 81-100

1.3 Оценочные материалы: типовые контрольные задания, иные материалы

1.3.1.1 Оценочные материалы для очной формы обучения (ОФО) (Курс 2, Семестр 4)

Модуль 1 (50 баллов):

Модуль содержит 4 лекционных занятия, 4 практических и 3 лабораторных работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 15. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 35 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы (ОПК-8):

1. Что такое сложность алгоритма?
2. Как оценивается временная сложность алгоритма?
3. Какие методы используются для оценки временной сложности алгоритма?
4. Назовите наиболее известные алгоритмы сортировки.
5. Какой алгоритм сортировки считается оптимальным по среднему случаю?
6. Объясните принцип пузырьковой сортировки.
7. Опишите алгоритм быстрой сортировки (QuickSort).
8. Как работает метод вставок (Insertion Sort)?
9. Когда целесообразно применять рекурсию?
10. Почему рекурсивные решения иногда приводят к проблемам производительности?
11. Приведите пример классической рекурсивной задачи.
12. Чем отличаются прямые и косвенные рекурсии?
13. Что такое абстрактный тип данных (АТД)? Зачем нужны АТД?
14. Перечислите распространенные примеры АТД.
15. Как реализуются списки в рамках концепции АТД?
16. Приведите реализацию списка с использованием массива.
17. Чем отличается односвязный список от двусвязного?
18. Что понимается под списком с динамическим выделением памяти?
19. Объясните особенности односвязного списка.
20. Что представляет собой очередь с приоритетом?
21. В чём заключается отличие реализаций очередей на массивах и списках?
22. Каковы базовые операции над стеком?
23. Как реализуется стек на статическом массиве?
24. В чём особенность стека на динамической структуре данных?
25. Расскажите о реализации очереди с приоритетом.
26. Что такое дек (deque)? Приведите его применение.
27. Что такое бинарная куча (binary heap)?
28. В чём различие между min-кучей и max-кучей?
29. Что значит поддерживать свойство кучи (heap property)?
30. Опишите шаги процедуры увеличения ключа (increase-key) в бинарной куче.
31. Каково назначение биномиальной кучи (binomial heap)?
32. Чем различаются биномиальная куча и левосторонняя куча (leftist heap)?
33. Что представляет собой косая куча (skew heap)?
34. Объясните различия между операциями слияния (merge) в разных видах куч.
35. Можно ли эффективно реализовать очередь с приоритетом на основе двоичной кучи?

Практическое занятие №1. Анализ сложности и эффективности алгоритма.

Контрольные вопросы (ОПК-8):

1. Что такое временная сложность алгоритма?
2. Какие основные классы временной сложности существуют?
3. Почему важно учитывать обе характеристики: временную и пространственную сложность?
4. Зачем применяется оптимизация алгоритмов, и какие подходы используются чаще всего?
5. Какие типы сортировок бывают и как классифицируются методы сортировки?
6. Что такое сортировка выбором, какова её временная сложность и область применения?
7. Объясните суть быстрой сортировки (QuickSort) и назовите особенности её реализации. Почему быстрая сортировка считается эффективной?
8. Опишите процесс сортировки вставками. Когда этот алгоритм предпочтителен?
9. Для чего применяется сортировка подсчетом (Counting Sort)? В чём её преимущество перед обычными методами?
10. Какой способ сортировки называют сортировкой кучей (HeapSort)? Обоснуйте его использование.
11. Назовите различия между сортировкой Шелла и обычной сортировкой вставками. Укажите преимущества и ограничения ShellSort.
12. Дайте характеристику пузырьковой сортировке (Bubble Sort). Есть ли ситуации, когда её применение оправдано?

Практическое занятие №2. Создание и демонстрация работы абстрактных типов данных (АТД).

Контрольные вопросы (ОПК-8):

1. Что такое абстрактный тип данных (АТД)?
2. В чем отличие структуры данных от абстрактного типа данных?
3. Приведи три примера классических АТД.
4. Каковы основные элементы описания любого АТД?
5. Какие операции характерны для очереди?
6. Чем отличается реализация списка от множества?
7. В каком случае удобнее использовать массив, а в каком – связный список?
8. Какие ограничения накладывает концепция АТД на доступ к данным внутри объекта?
9. Можно ли изменять внутреннюю структуру данных при сохранении внешнего интерфейса АТД?
10. В чем состоит важность принципа сокрытия деталей реализации в концепции АТД?

Практическое занятие №3. Реализация задачи "Word Count"(алфавитно-частотный словарь). Подсчёт символов, слов и строк.

Контрольные вопросы (ОПК-8):

1. Что значит термин «Word Count» применительно к обработке текста?
2. Какие этапы включает в себя решение задачи «Word Count»?
3. Как определяется слово в рамках задания «Word Count»?
4. Какая структура данных подходит для хранения алфавитно-частотного словаря?
5. В чём разница между количеством символов и количеством слов в тексте?
6. Нужно ли учитывать регистр букв при построении частотного словаря?
7. Следует ли включать знаки препинания в счётчик слов?

8. Если файл большой, какое оптимальное решение позволяет минимизировать потребление оперативной памяти?
9. Что произойдёт, если входной файл пуст? Как должен вести себя алгоритм обработки?
10. Сколько строк выводится в итоговом отчёте «Word Count» по умолчанию?

Практическое занятие №4. Реализация стека. Реализация очереди.

Контрольные вопросы (ОПК-8):

1. Что такое стек (Stack) и какие основные операции с ним выполняются?
2. Чем характеризуется порядок извлечения элементов из стека?
3. Приведи пример ситуации, когда удобно использовать стек.
4. Что такое очередь (Queue) и какие ключевые операции она поддерживает?
5. Каково основное свойство порядка добавления и удаления элементов в очереди?
6. Изобрази графически операцию push() и pop() для стека.
7. Изобрази графически операции enqueue() и dequeue() для очереди.
8. В чём принципиальное различие между реализацией стека и очередью?
9. Почему важна проверка границ массива при реализации стека и очереди?
10. Чем полезна конструкция LIFO ("Last In First Out") и FIFO ("First In First Out") в реальных приложениях?

Лабораторная работа №1. Массивы. Указатели. Связь указателей и массивов. Динамические массивы. Указатели и параметры функций.

Контрольные вопросы (ОПК-8):

1. Что такое массив в C/C++ и как он объявляется?
2. Как связаны массивы и указатели в языках C/C++?
3. Что такое индекс массива и каковы его границы?
4. Объясните, что значит выражение &arr[i] (arr – массив)?
5. Как объявляются и инициализируются динамические массивы?
6. Что означает передача массива в качестве параметра функции?
7. Как осуществляется доступ к элементам массива через указатель?
8. Чем отличается статический массив от динамически выделенного?
9. Что случится, если обратиться к несуществующему элементу массива?
10. Как реализуется передача двумерного массива в функцию через указатели?

Лабораторная работа №2. Реализация задачи "Word Count"(алфавитно-частотный словарь). Работа с файлами. Подсчет символов в строке с учетом регистра. Подсчет подстрок в строке с учетом перекрытий.

Контрольные вопросы (ОПК-8):

1. Какие стандартные библиотеки C++ используются для работы с файлами?
2. Как открыть файл для чтения и записи в программе на C++?
3. Как считать строку из файла с помощью стандартной библиотеки iostream?
4. Что такое частота появления символа в строке и как её измеряют?
5. Как учитывается регистр символов при подсчёте частоты символов?
6. Напиши псевдокод функции для нахождения всех вхождений подстроки в строку с учётом перекрытия.
7. Что произойдет, если попытаться прочитать файл, которого не существует?
8. Чем различаются глобальные переменные и локальные переменные в контексте программы на C++?

9. Какие функции или конструкции языка C++ подходят для отображения результата в виде алфавитно-частотного словаря?
10. Что необходимо учесть при передаче большого объёма данных между функциями в контексте обработки текста?

***Лабораторная работа №3. Реализация дека с помощью двусвязного списка.
Реализация очереди.***

Контрольные вопросы (ОПК-8):

1. Что такое дек (deque) и какими основными операциями он обладает?
2. Чем отличается реализация очереди (FIFO) от реализации дека?
3. Как реализуются операции вставки и удаления элементов в двусвязном списке?
4. В чём главное преимущество двусвязного списка перед односвязным списком?
5. Нарисуйте схему двусвязного списка с двумя элементами.
6. Почему важно проверять границы контейнера при удалении элемента из начала или конца?
7. Как добавить новый узел в конец двусвязного списка?
8. Что произойдёт, если удалить последний элемент из пустой очереди?
9. Какие стандартные контейнеры STL можно использовать для реализации дека и очереди?
10. Как передавать ссылку на объект двусвязного списка в функцию на языке C++?

Модуль 2 (50 баллов):

Модуль содержит 4 лекционных занятий, 3 практических и 2 лабораторных работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 15. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 35 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы (ОПК-8):

1. Что такое хэширование и как оно связано с хэш-таблицей?
2. Что такое коллизия и какие методы её разрешения существуют?
3. Как организовать прямую адресацию в хэш-таблице?
4. Какие критерии хорошей хэш-функции?
5. Какие проблемы возникают при плохих хэш-функциях?
6. Что такое равномерное распределение значений хэш-функции?
7. В чём отличие открытых адресов от закрытых адресов в разрешении коллизий?
8. Чем полезны хэш-коды и как они влияют на безопасность данных?
9. Какие функции контроля целостности данных обеспечиваются средствами хэширования?
10. Чем отличается симметричное шифрование от асимметричного?
11. Что такое цифровая подпись и как она защищает данные?
12. Что такое система открытого ключа и как она организована?
13. Как организованы поисковые индексы и что такое обратные индексы?
14. Какие технологии используют поисковые системы для повышения скорости выдачи результатов?
15. Что такое последовательное хэширование и где оно применяется?

16. В чём особенность двойного хэширования?
17. Какие виды хэширования известны и где они используются?
18. Что такое универсальное хэширование и почему оно популярно?
19. Что такое согласованное хэширование и когда оно полезно?
20. В чём состоят потенциальные угрозы безопасности при неправильном выборе хэш-функции?
21. Что такое корень дерева и лист узла?
22. Как определить высоту дерева и диаметр дерева?
23. В чём отличие полных деревьев от почти-полных деревьев?
24. Какая операция необходима для поддержания баланса в дереве?
25. Какими свойствами обладают бинарные деревья поиска (BST)?
26. Что такое сбалансированное дерево и зачем оно нужно?
27. Какие варианты обхода дерева существуют (in-order, pre-order, post-order)?
28. Как реализован класс бинарного дерева поиска на языке C++?
29. В чём отличия бинарного дерева поиска от AVL-дерева?
30. Как обрабатываются дубликаты в бинарном дереве поиска?
31. Что такое очередь с приоритетом и как она реализована?
32. Какие методы применяются для поддержания упорядоченности в структуре данных с приоритетами?
33. Что такое внутреннее представление AVL-дерева?
34. Какие правила сохранения баланса соблюдаются в AVL-дереве?
35. Как осуществляется включение новой вершины в AVL-дерево?
36. Как изменяется высота дерева после включения нового узла?
37. Какие вращения необходимы для восстановления баланса в AVL-дереве?
38. В чём разница между левым и правым вращениями?
39. Как происходит удаление вершины из AVL-дерева?
40. Какие изменения происходят в дереве после удаления вершины?

Практическое занятие №5. Алгоритмы хеширования данных. Построение функций хеширования и алгоритмов хеширования данных, при решении задач на языке C++

Контрольные вопросы (ОПК-8):

1. Что такое хеширование и для чего оно применяется?
2. Какие свойства должны обладать хорошие хеш-функции?
3. В чём опасность возникновения коллизий и как они решаются?
4. Приведите пример простой хеш-функции на языке C++.
5. Чем отличаются методы открытой адресации и цепочек для разрешения коллизий?
6. Как можно повысить качество хеш-функции?
7. Как построить хеш-таблицу с открытым размещением на языке C++?
8. В чём особенность идеальной хеш-функции?
9. Чем отличается хеширование целых чисел от хеширования строковых данных?
10. Какова цель универсального хеширования и как оно реализовано?

Практическое занятие №6. Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Реализация алгоритма построения оптимального бинарного дерева поиска

Контрольные вопросы (ОПК-8):

1. Что такое бинарное дерево поиска?
2. Какие основные операции можно выполнять с бинарным деревом поиска?

3. Какая последовательность действий выполняется при вставке нового элемента в бинарное дерево поиска?
4. Какую структуру данных целесообразно использовать для представления узлов бинарного дерева поиска в языке C++?
5. Какие случаи возможны при удалении узла из бинарного дерева поиска?
6. В чём отличие поиска в бинарном дереве поиска от линейного поиска?
7. Как построить оптимальный бинарный поисковый алгоритм и зачем он нужен?
8. Как посчитать высоту бинарного дерева поиска?
9. Как выполнить поиск минимального значения в бинарном дереве поиска?
10. В чём заключается идея сбалансированности бинарного дерева поиска и зачем она важна?

Практическое занятие №7. Программная реализация AVL-дерева. Построение AVL-дерева, которое будет, являться объединением деревьев T1 и T2.

Контрольные вопросы (ОПК-8):

1. Что такое AVL-дерево и в чём его отличие от обычного бинарного дерева поиска?
2. Какими характеристиками обладает идеально сбалансированное AVL-дерево?
3. Какая максимальная разница высот левого и правого поддеревьев допустима в узле AVL-дерева?
4. Какие операции необходимо выполнить при включении новой вершины в AVL-дерево?
5. Какие виды вращений применяются для восстановления баланса в AVL-дереве?
6. В чём отличие правого поворота от левого поворота в AVL-дереве?
7. Какие шаги требуются для правильного объединения двух AVL-деревьев T1 и T2, если известно, что ключи в T1 меньше ключей в T2?
8. Как реализовать объединение двух AVL-деревьев в одном классе на языке C++?
9. В каких случаях возникает необходимость переупорядочить дерево после добавления новой вершины?
10. Что произойдёт, если попробовать объединить два AVL-дерева, одно из которых содержит ключи больше другого, без соблюдения условий правильной последовательности ключей?

Лабораторная работа №4. Реализация алгоритмов хэширования. Составление хеш-таблиц. Построение хеш-таблиц из текстового файла. Построение хеш-таблиц для зарезервированных слов, используемого языка программирования.

Контрольные вопросы (ОПК-8):

1. Что такое хэширование и как оно используется в хеш-таблицах?
2. Какие основные компоненты входят в структуру хеш-таблицы?
3. Что такое коллизия, и какие методы её разрешения существуют?
4. Приведите пример простейшей хеш-функции для строк.
5. Как создать простую хеш-таблицу на языке C++?
6. Какая стратегия хранения используется в случае коллизий методом цепочек?
7. Что такое открытое размещение (открытые адреса) и как оно применяется в хеш-таблицах?
8. Как обеспечить эффективную загрузку хеш-таблицы (коэффициент заполнения)?
9. Какие меры предосторожности надо принять при создании хеш-таблицы из текстового файла?
10. Как автоматизировать создание хеш-таблицы для набора зарезервированных слов конкретного языка программирования (например, C++)?

Лабораторная работа №5. Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Реализация алгоритма построения оптимального бинарного дерева поиска с квадратичным временем работы.

Контрольные вопросы (ОПК-8):

1. Что такое бинарное дерево поиска (BST)?
2. Какие операции поддерживает бинарное дерево поиска?
3. Какая гарантия правильности сохраняется при вставке новых элементов в бинарное дерево поиска?
4. Какое условие определяет оптимальное бинарное дерево поиска?
5. Почему оптимальная структура дерева влияет на производительность поиска?
6. Какие действия выполняются при вставке элемента в бинарное дерево поиска?
7. Какова вычислительная сложность алгоритма поиска в плохо сбалансированном бинарном дереве?
8. Почему построение оптимального бинарного дерева требует квадратичного времени работы?
9. В чём особенность оптимальной структуры бинарного дерева относительно равномерного распределения элементов?
10. Как изменится алгоритм поиска в бинарном дереве поиска, если оно оптимально построено?

1.3.2.1 Вопросы, выносимые на зачет по дисциплине «Структуры и алгоритмы обработки данных» (ОПК-8)

1. Что такое сложность алгоритма?
2. Как оценивается временная сложность алгоритма?
3. Какие методы используются для оценки временной сложности алгоритма?
4. Назовите наиболее известные алгоритмы сортировки.
5. Какой алгоритм сортировки считается оптимальным по среднему случаю?
6. Объясните принцип пузырьковой сортировки.
7. Опишите алгоритм быстрой сортировки (QuickSort).
8. Как работает метод вставок (Insertion Sort)?
9. Когда целесообразно применять рекурсию?
10. Почему рекурсивные решения иногда приводят к проблемам производительности?
11. Приведите пример классической рекурсивной задачи.
12. Чем отличаются прямые и косвенные рекурсии?
13. Что такое абстрактный тип данных (АТД)? Зачем нужны АТД?
14. Перечислите распространенные примеры АТД.
15. Как реализуются списки в рамках концепции АТД?
16. Приведите реализацию списка с использованием массива.
17. Чем отличается односвязный список от двусвязного?
18. Что понимается под списком с динамическим выделением памяти?
19. Объясните особенности односвязного списка.
20. Что представляет собой очередь с приоритетом?
21. В чём заключается отличие реализаций очередей на массивах и списках?
22. Каковы базовые операции над стеком?
23. Как реализуется стек на статическом массиве?
24. В чём особенность стека на динамической структуре данных?
25. Расскажите о реализации очереди с приоритетом.
26. Что такое дек (deque)? Приведите его применение.
27. Что такое бинарная куча (binary heap)?
28. В чём различие между min-кучей и max-кучей?
29. Что значит поддерживать свойство кучи (heap property)?
30. Опишите шаги процедуры увеличения ключа (increase-key) в бинарной куче.

31. Каково назначение биномиальной кучи (binomial heap)?
32. Чем различаются биномиальная куча и левосторонняя куча (leftist heap)?
33. Что представляет собой косая куча (skew heap)?
34. Объясните различия между операциями слияния (merge) в разных видах куч.
35. Можно ли эффективно реализовать очередь с приоритетом на основе двоичной кучи?
36. Что такое хэширование и как оно связано с хэш-таблицей?
37. Что такое коллизия и какие методы её разрешения существуют?
38. Как организовать прямую адресацию в хэш-таблице?
39. Какие критерии хорошей хэш-функции?
40. Какие проблемы возникают при плохих хэш-функциях?
41. Что такое равномерное распределение значений хэш-функции?
42. В чём отличие открытых адресов от закрытых адресов в разрешении коллизий?
43. Чем полезны хэш-коды и как они влияют на безопасность данных?
44. Какие функции контроля целостности данных обеспечиваются средствами хэширования?
45. Чем отличается симметричное шифрование от асимметричного?
46. Что такое цифровая подпись и как она защищает данные?
47. Что такое система открытого ключа и как она организована?
48. Как организованы поисковые индексы и что такое обратные индексы?
49. Какие технологии используют поисковые системы для повышения скорости выдачи результатов?
50. Что такое последовательное хэширование и где оно применяется?
51. В чём особенность двойного хэширования?
52. Какие виды хэширования известны и где они используются?
53. Что такое универсальное хэширование и почему оно популярно?
54. Что такое согласованное хэширование и когда оно полезно?
55. В чём состоят потенциальные угрозы безопасности при неправильном выборе хэш-функции?
56. Что такое корень дерева и лист узла?
57. Как определить высоту дерева и диаметр дерева?
58. В чём отличие полных деревьев от почти-полных деревьев?
59. Какая операция необходима для поддержания баланса в дереве?
60. Какими свойствами обладают бинарные деревья поиска (BST)?
61. Что такое сбалансированное дерево и зачем оно нужно?
62. Какие варианты обхода дерева существуют (in-order, pre-order, post-order)?
63. Как реализован класс бинарного дерева поиска на языке C++?
64. В чём отличия бинарного дерева поиска от AVL-дерева?
65. Как обрабатываются дубликаты в бинарном дереве поиска?
66. Что такое очередь с приоритетом и как она реализована?
67. Какие методы применяются для поддержания упорядоченности в структуре данных с приоритетами?
68. Что такое внутреннее представление AVL-дерева?
69. Какие правила сохранения баланса соблюдаются в AVL-дереве?
70. Как осуществляется включение новой вершины в AVL-дерево?
71. Как изменяется высота дерева после включения нового узла?
72. Какие вращения необходимы для восстановления баланса в AVL-дереве?
73. В чём разница между левым и правым вращениями?
74. Как происходит удаление вершины из AVL-дерева?
75. Какие изменения происходят в дереве после удаления вершины?

1.3.1.2 Оценочные материалы для очной формы обучения (ОФО) (Курс 3, Семестр 5)

Модуль 1 (50 баллов):

Модуль содержит 4 лекционных занятия и 4 лабораторных работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 10. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 40 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы (ОПК-8):

1. Перечислите пять свойств красно чёрного дерева. Какое из них гарантирует сбалансированность?
2. При вставке узла в красно чёрное дерево могут возникнуть три случая нарушения свойств. Опишите каждый случай и действия по их исправлению.
3. Какие виды вращений используются в красно чёрных деревьях?
4. Приведите пример левого поворота вокруг узла.
5. Опишите процесс удаления узла из красно чёрного дерева.
6. Как обрабатываются случаи, когда удаляемый узел чёрный?
7. Что такое дерево промежутков?
8. Как оно организовано и для каких задач применяется?
9. Опишите алгоритм поиска всех интервалов, пересекающих произвольно заданный интервал, в дереве промежутков.
10. В чём ключевое отличие SPLAY дерева от других сбалансированных деревьев (AVL, красно чёрных)?
11. Какие операции в SPLAY дереве называют словарными?
12. Как реализуются операции в SPLAY-дереве?
13. Опишите процедуру splay (протаскивание) узла к корню.
14. Перечислите возможные случаи вращений (zig, zig zig, zig zag).
15. Как работают операции split и merge в SPLAY дереве? Для чего они нужны?
16. Опишите алгоритм вставки нового узла в SPLAY дерево.
17. Зачем в SPLAY деревьях нужны вспомогательные процедуры для работы с указателями на родителей? Как они поддерживаются?
18. Как выполняется операция find в SPLAY дереве? Что происходит с найденным узлом?
19. Сравните амортизированную сложность основных операций SPLAY деревьев с красно чёрными деревьями.
20. Что такое B дерево? Назовите его основные параметры (минимальную степень t , максимальное количество ключей в узле).
21. Опишите процесс вставки ключа в B дерево.
22. Когда и как происходит расщепление узлов?
23. Опишите процесс удаления ключа из B дерева. Какие случаи требуют слияния или заимствования ключей?
24. Почему B деревья широко применяются в базах данных и файловых системах?
25. Как в B дереве выполняется исправление потенциальной разбалансировки на этапе поиска удаляемого элемента?
26. Что такое биномиальное дерево B_k ? Перечислите его свойства (количество узлов, высота, степень).
27. Какие основные операции поддерживает биномиальная куча?
28. Оцените их временную сложность.
29. Опишите алгоритм слияния (merge) двух биномиальных куч.
30. В чём преимущества биномиальной кучи перед обычной двоичной? В чём недостатки?

31. Что такое куча Фибоначчи? Какие операции в ней имеют лучшие амортизированные оценки по сравнению с биномиальной кучей?
32. Опишите структуру кучи Фибоначчи: корневой список, каскадная обрезка, пометки узлов.
33. Как выполняется операция `extract_min` в куче Фибоначчи? Какие шаги необходимы и как достигается амортизированная сложность $O(\log n)$?
34. Что такое фибоначчьево дерево? Как оно связано с кучей Фибоначчи?
35. Предложите сценарии, в которых использование `SPLAY` дерева предпочтительнее красно-чёрного, и наоборот. Аргументируйте ответ.

Лабораторная работа №1. Программная реализация красно-черных деревьев. Добавление и удаление вершин. Дерево промежутков.

Контрольные вопросы (ОПК-8):

1. Какие свойства красно-черного дерева нарушаются при вставке нового узла?
2. Опишите алгоритм удаления узла из красно-черного дерева.
3. Какие дополнительные действия требуются, если удаляемый узел черный?
4. Как выполняется балансировка после удаления?
5. Как проверить корректность красно-черного дерева после вставки и удаления?
6. Что такое дерево промежутков (`interval tree`)?
7. Какая дополнительная информация хранится в каждом узле и как она используется при поиске пересекающихся интервалов?
8. Опишите процесс вставки интервала в дерево промежутков. Как поддерживается поле максимального правого конца в узлах при вставке и удалении?
9. Использовались рекурсивные или итеративные алгоритмы для операций балансировки?
10. Какие преимущества и недостатки каждого подхода вы можете выделить?

Лабораторная работа №2. Программная реализация `SPLAY`-деревьев. Построение двух корректных деревьев поиска, построенные на одном и том же множестве ключей друг от друга последовательностью поворотов.

Контрольные вопросы (ОПК-8):

1. Какие основные операции поддерживает `SPLAY`-дерево? Опишите алгоритм операции `splay` и её роль в работе дерева.
2. В чем отличие `SPLAY`-дерева от других сбалансированных деревьев (например, AVL или красно-черных) с точки зрения гарантий времени работы?
3. Как выполняется вставка нового ключа в `SPLAY`-дерево? Приведите последовательность шагов.
4. Какие возможны случаи вращений (`zig`, `zig-zig`, `zig-zag`) при выполнении операции `splay`? Приведите примеры для каждого случая.
5. Объясните, как реализуются операции `split` и `merge` в `SPLAY`-дереве. Для каких целей они используются?
6. Как в вашей реализации организовано хранение указателей на родительские узлы? Почему это важно для операции `splay`?
7. В задании требуется построить два корректных дерева поиска на одном и том же множестве ключей, которые можно преобразовать друг в друга последовательностью поворотов. Докажите, что для любых двух бинарных деревьев поиска с одинаковым набором ключей существует такая последовательность.
8. Опишите алгоритм преобразования одного дерева поиска в другое с помощью поворотов (вращений). Какие типы поворотов (левый, правый) используются?
9. Приведите пример двух деревьев поиска, построенных на множестве ключей $\{1,2,3,4,5\}$, и покажите последовательность поворотов, переводящую одно дерево в другое.
10. Какие сложности могут возникнуть при реализации преобразования деревьев с помощью поворотов? Как вы проверяли корректность вашей реализации?

Лабораторная работа №3. Программная реализация В-деревьев. Создание бинарного дерева поиска. Поиск образца в бинарном дереве. Подсчет количества узлов дерева. Подсчет числа листьев дерева. Удаление всех узлов из бинарного дерева

Контрольные вопросы (ОПК-8):

1. Какие параметры определяют порядок В-дерева и как они задаются?
2. Опишите алгоритм вставки ключа в В-дерево. При каких условиях происходит расщепление узла?
3. Как реализован поиск ключа в В-дереве? В чём отличие от поиска в бинарном дереве?
4. Приведите условие, при котором при удалении ключа из В-дерева требуется слияние узлов.
5. Как организовано хранение узлов В-дерева: массив ключей, ссылки на потомков, количество ключей?
6. Какие функции реализованы для бинарного дерева поиска (создание, добавление, обход)?
7. Как выполняется подсчёт количества узлов в бинарном дереве? Укажите рекурсивный способ.
8. Как определяется, является ли узел листом? Приведите условие и способ подсчёта листьев.
9. Как удалить все узлы бинарного дерева? Как избежать утечек памяти?
10. В чём заключаются особенности поиска образца (поддерева) в бинарном дереве?

Лабораторная работа №4. Разработка класса для структуры данных биномиальной кучи. Реализация операции формирования кучи из множества элементов. Визуализация кучи.

Контрольные вопросы (ОПК-8):

1. Какие поля содержит класс узла биномиальной кучи?
2. Как организовано хранение списка корневых деревьев?
3. Опишите алгоритм слияния двух биномиальных куч.
4. Как реализована операция вставки нового элемента?
5. Как выполняется извлечение минимального элемента?
6. Как формируется куча из массива элементов (через вставки или построение)?
7. Какие методы предусмотрены для визуализации кучи?
8. Как обеспечивается уникальность деревьев каждого порядка?
9. Как реализована операция уменьшения ключа и для чего она нужна?
10. Какие сложности возникают при отладке слияния и извлечения минимума?

Модуль 2 (50 баллов):

Модуль содержит 4 лекционных занятия и 4 лабораторных работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 10. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 40 баллов. Общее максимальное количество баллов за модуль 2 составляет 50 баллов.

Вопросы для контрольной работы (ОПК-8):

1. Дайте определение графа. Чем ориентированный граф отличается от неориентированного?
2. Что такое степень вершины? Приведите формулу суммы степеней.
3. Какие графы называют связными? Приведите пример несвязного графа.
4. Что такое цикломатическое число графа? Как оно вычисляется?
5. Как найти максимальное остовное дерево взвешенного графа?
6. Опишите структуру данных «система непересекающихся множеств» (DSU).

7. Какие операции она поддерживает?
8. В чём разница между реализацией DSU на массиве и на лесе (с эвристиками)?
9. Что такое частично упорядоченное множество?
10. Приведите пример отношения порядка.
11. Какие операции можно выполнять над частично упорядоченным множеством?
12. Опишите алгоритм поиска в ширину (BFS). Для каких задач он применяется?
13. Чем поиск в глубину (DFS) отличается от BFS? Приведите пример использования DFS.
14. Что такое минимальное покрывающее дерево (MST)? Назовите два классических алгоритма его построения.
15. Как работает алгоритм Краскала для MST?
16. Как работает алгоритм Прима для MST?
17. Для решения каких задач применяют алгоритмы поиска кратчайшего пути на взвешенном графе?
18. Опишите алгоритм Дейкстры. Какое ограничение на веса рёбер у него есть?
19. Для каких графов используют алгоритм Беллмана-Форда?
20. Что такое цифровой поиск? В чём его отличие от обычного бинарного поиска?
21. Опишите структуру цифрового дерева (trie) для набора строк.
22. Что такое хеширование с открытой адресацией?
23. Перечислите основные методы разрешения коллизий.
24. Какие требования предъявляются к хеш-функции для открытой адресации?
25. Как выполняется поиск элемента в хеш-таблице с открытой адресацией?
26. Какие типы данных можно хешировать?
27. Приведите примеры хеш-функций для чисел и строк.
28. Что такое обобщение цифрового поиска?
29. В каких структурах оно используется?
30. Как можно комбинировать цифровой поиск с поиском в последовательных таблицах?
31. Приведите пример задачи, где эффективно использовать сочетание цифрового поиска и хеширования.
32. Что такое «лес» в контексте DSU? Как он ускоряет операции?
33. Как вычислить цикломатическое число для связного графа?
34. В чём заключается эвристика «сжатие пути» в DSU?
35. Как найти компоненты связности графа с помощью BFS или DSU?

Лабораторная работа №5. Задачи на графах. Построение матрицы смежности. Построение матрицы инцидентности.

Контрольные вопросы (ОПК-8):

1. Как в вашей программе задаётся граф: список рёбер, матрица смежности или иной формат?
2. Опишите алгоритм построения матрицы смежности по списку рёбер.
3. Какие размерности имеет матрица инцидентности для неориентированного графа?
4. Как отличаются матрицы инцидентности для ориентированного и неориентированного графов?
5. Как реализовано заполнение матрицы инцидентности: что записывается в ячейку при инцидентности вершины и ребра?
6. Какие структуры данных выбраны для хранения матриц (список списков, ptrу, массив)?
7. Как проверяется корректность ввода графа перед построением матриц?
8. Какие операции над графом вы реализовали с использованием матрицы смежности (поиск соседей, проверка связности)?
9. Как выводится матрица смежности и матрица инцидентности на экран (форматирование)?
10. Какие ограничения по размеру графа накладывает выбранное представление?

Лабораторная работа №6. Реализация с использованием списков. Построение списков связи. Построение перечней рёбер.

Контрольные вопросы (ОПК-8):

1. Как организовать список смежности на основе массива списков?
2. Как добавить ребро в список смежности для ориентированного графа?
3. Как удалить ребро из списка смежности?
4. Как проверить наличие ребра между двумя вершинами в списке смежности?
5. Как построить перечень рёбер по готовому списку смежности?
6. Как по перечню рёбер восстановить список смежности?
7. Как оценить память, занимаемую списком смежности и перечнем рёбер?
8. Как реализовать обход графа в глубину, используя только список смежности?
9. Как учесть кратные рёбра при построении списков связи?
10. Какие операции с графом эффективнее выполнять на списках смежности, а какие – на перечне рёбер?

Лабораторная работа №7 Программная реализация алгоритмов на графах. Построение остовного дерева путем обхода в глубину (в ширину). Построение матрицы кратчайших путей.

Контрольные вопросы (ОПК-8):

1. Как реализовать построение остовного дерева обходом в глубину?
2. Как организовать обход в ширину для получения остова?
3. Чем отличается структура остовного дерева при BFS и DFS?
4. Как хранить полученное остовное дерево (список рёбер, матрица)?
5. Как реализовать алгоритм Флойда-Уоршелла для матрицы кратчайших путей?
6. Как инициализировать матрицу весов перед поиском кратчайших путей?
7. Как восстановить пути по готовой матрице кратчайших расстояний?
8. Как учесть отсутствие ребра (бесконечность) при реализации?
9. Как проверить корректность построенной матрицы кратчайших путей?
10. Какие структуры данных выбрать для эффективной реализации алгоритмов?

Лабораторная работа №8. Обработка словаря с высечками. Использование многоуровневых высечек.

Контрольные вопросы (ОПК-8):

1. Как хранить словарь для эффективной обработки высечек?
2. Как реализовать одноуровневую высечку по ключу?
3. Как организовать многоуровневую высечку (вложенные высечки)?
4. Как обрабатывать конфликты при наложении высечек?
5. Какой алгоритм используется для применения высечки ко всем элементам словаря?
6. Как восстановить исходный словарь после высечек?
7. Какие структуры данных подходят для хранения многоуровневых высечек?
8. Как обеспечить независимость высечек разных уровней?
9. Как оценить временную сложность применения высечек?
10. Как выполнить откат последней высечки?

1.3.2.2 Вопросы, выносимые на экзамен по дисциплине «Структуры и алгоритмы обработки данных» (ОПК-8)

1. Что такое сложность алгоритма?
2. Как оценивается временная сложность алгоритма?
3. Назовите наиболее известные алгоритмы сортировки.
4. Опишите алгоритм быстрой сортировки (QuickSort).
5. Когда целесообразно применять рекурсию?
6. Что такое абстрактный тип данных (АТД)? Зачем нужны АТД?

7. Чем отличается односвязный список от двусвязного?
8. Что представляет собой список с динамическим выделением памяти?
9. Каковы базовые операции над стеком?
10. Как реализуется очередь с приоритетом?
11. Что такое бинарная куча (binary heap)?
12. В чём различие между min-кучей и max-кучей?
13. Что такое хэширование и как оно связано с хэш-таблицей?
14. Что такое коллизия и какие методы её разрешения существуют?
15. Какие критерии хорошей хэш-функции?
16. Что такое корень дерева и лист узла?
17. Какими свойствами обладают бинарные деревья поиска (BST)?
18. Какие варианты обхода дерева существуют (in-order, pre-order, post-order)?
19. Что такое сбалансированное дерево и зачем оно нужно?
20. В чём отличия бинарного дерева поиска от AVL-дерева?
21. Какие вращения необходимы для восстановления баланса в AVL-дереве?
22. Перечислите пять свойств красно-чёрного дерева. Какое из них гарантирует сбалансированность?
23. При вставке узла в красно-чёрное дерево могут возникнуть три случая нарушения свойств. Опишите каждый случай и действия по их исправлению.
24. Какие виды вращений используются в красно-чёрных деревьях? Приведите пример левого поворота.
25. Опишите процесс удаления узла из красно-чёрного дерева. Как обрабатываются случаи, когда удаляемый узел чёрный?
26. Почему красно-чёрные деревья гарантируют высоту $O(\log n)$, в отличие от обычных BST?
27. В каких сценариях красно-чёрное дерево предпочтительнее AVL-дерева, и наоборот?
28. Опишите процедуру splay (протаскивание) узла к корню.
29. Что такое B-дерево? Назовите его основные параметры (минимальную степень t , максимальное количество ключей в узле).
30. Почему B-деревья широко применяются в базах данных и файловых системах?
31. Что такое биномиальное дерево B_k ? Перечислите его свойства.
32. Опишите алгоритм слияния (merge) двух биномиальных куч.
33. Что такое куча Фибоначчи? Какие операции в ней имеют лучшие амортизированные оценки по сравнению с биномиальной кучей?
34. Дайте определение графа. Чем ориентированный граф отличается от неориентированного?
35. Что такое степень вершины? Приведите формулу суммы степеней.
36. Какие графы называют связными? Приведите пример несвязного графа.
37. Опишите структуру данных «система непересекающихся множеств» (DSU). Какие операции она поддерживает?
38. В чём разница между реализацией DSU на массиве и на лесе (с эвристиками)?
39. Опишите алгоритм поиска в ширину (BFS). Для каких задач он применяется?
40. Чем поиск в глубину (DFS) отличается от BFS? Приведите пример использования DFS.
41. Что такое минимальное покрывающее дерево (MST)? Назовите два классических алгоритма его построения.
42. Как работает алгоритм Краскала для MST?
43. Как работает алгоритм Прима для MST?
44. Опишите алгоритм Дейкстры. Какое ограничение на веса рёбер у него есть?
45. Для каких графов используют алгоритм Беллмана-Форда?
46. Что такое цифровой поиск? В чём его отличие от обычного бинарного поиска?
47. Опишите структуру цифрового дерева (trie) для набора строк.

48. Что такое хеширование с открытой адресацией? Перечислите основные методы разрешения коллизий.
49. Как организовать список смежности для графа?
50. Как по списку смежности построить перечень рёбер?
51. Как реализовать построение остовного дерева обходом в глубину (или ширину)?
52. Как реализовать алгоритм Флойда–Уоршелла для матрицы кратчайших путей?
53. Что такое многоуровневые высечки при обработке словаря?
54. Как обеспечить откат (undo) последней высечки в словаре?
55. Сравните амортизированную сложность основных операций SPLAY-деревьев с красно-чёрными деревьями.

1.3.3.1 Оценочные материалы для заочной формы обучения (ЗФО) (Курс 2, Сессия 3)

Модуль 1 (50 баллов):

Модуль содержит 3 лекционных и 2 практических занятия. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 20. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 15 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы (ОПК-8):

1. Что такое сложность алгоритма?
2. Как оценивается временная сложность алгоритма?
3. Какие методы используются для оценки временной сложности алгоритма?
4. Назовите наиболее известные алгоритмы сортировки.
5. Какой алгоритм сортировки считается оптимальным по среднему случаю?
6. Объясните принцип пузырьковой сортировки.
7. Опишите алгоритм быстрой сортировки (QuickSort).
8. Как работает метод вставок (Insertion Sort)?
9. Когда целесообразно применять рекурсию?
10. Почему рекурсивные решения иногда приводят к проблемам производительности?
11. Приведите пример классической рекурсивной задачи.
12. Чем отличаются прямые и косвенные рекурсии?
13. Что такое абстрактный тип данных (АТД)? Зачем нужны АТД?
14. Перечислите распространенные примеры АТД.
15. Как реализуются списки в рамках концепции АТД?
16. Приведите реализацию списка с использованием массива.
17. Чем отличается односвязный список от двусвязного?
18. Что понимается под списком с динамическим выделением памяти?
19. Объясните особенности односвязного списка.
20. Что представляет собой очередь с приоритетом?
21. В чём заключается отличие реализаций очередей на массивах и списках?
22. Каковы базовые операции над стеком?
23. Как реализуется стек на статическом массиве?
24. В чём особенность стека на динамической структуре данных?
25. Расскажите о реализации очереди с приоритетом.
26. Что такое дек (deque)? Приведите его применение.
27. Что такое бинарная куча (binary heap)?
28. В чём различие между min-кучей и max-кучей?

29. Что значит поддерживать свойство кучи (heap property)?
30. Опишите шаги процедуры увеличения ключа (increase-key) в бинарной куче.
31. Каково назначение биномиальной кучи (binomial heap)?
32. Чем различаются биномиальная куча и левосторонняя куча (leftist heap)?
33. Что представляет собой косая куча (skew heap)?
34. Объясните различия между операциями слияния (merge) в разных видах куч.
35. Можно ли эффективно реализовать очередь с приоритетом на основе двоичной кучи?

Практическое занятие №1. Составление эффективного программного кода. Массивы. Указатели. Связь указателей и массивов. Динамические массивы. Указатели и параметры функций.

Контрольные вопросы (ОПК-8):

1. Зачем применяется оптимизация алгоритмов, и какие подходы используются чаще всего?
2. Какие типы сортировок бывают и как классифицируются методы сортировки?
3. Что такое массив в C/C++ и как он объявляется?
4. Как связаны массивы и указатели в языках C/C++?
5. Как объявляются и инициализируются динамические массивы?
6. Что означает передача массива в качестве параметра функции?
7. Как осуществляется доступ к элементам массива через указатель?
8. Чем отличается статический массив от динамически выделенного?
9. Что случится, если обратиться к несуществующему элементу массива?
10. Как реализуется передача двумерного массива в функцию через указатели?

Практическое занятие №2. Создание и демонстрация работы абстрактных типов данных (АТД). Реализация задачи "Word Count"(алфавитно-частотный словарь). Работа с файлами. Подсчет символов в строке с учетом регистра. Подсчет подстрок в строке с учетом перекрытий.

Контрольные вопросы (ОПК-8):

1. Что такое абстрактный тип данных (АТД)?
2. В чем отличие структуры данных от абстрактного типа данных?
3. В чем состоит важность принципа сокрытия деталей реализации в концепции АТД?
4. Что значит термин «Word Count» применительно к обработке текста?
5. Как определяется слово в рамках задания «Word Count»?
6. Какая структура данных подходит для хранения алфавитно-частотного словаря?
7. Если файл большой, какое оптимальное решение позволяет минимизировать потребление оперативной памяти?
8. Что произойдет, если входной файл пуст? Как должен вести себя алгоритм обработки?
9. Какие стандартные библиотеки C++ используются для работы с файлами?
10. Как открыть файл для чтения и записи в программе на C++?

1.3.3.1.1 Оценочные материалы для заочной формы обучения (ЗФО) (Курс 3, Сессия 2)

Модуль 2 (50 баллов):

Модуль содержит 2 лекционных занятия, 2 практические и 2 лабораторные работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 10. За выполненные и защищенные практические и лабораторные занятия студент получает максимум 10 баллов. Общее максимальное количество баллов за модуль 2 составляет 50 баллов.

Вопросы для контрольной работы (ОПК-8):

1. Что такое хэширование и как оно связано с хэш-таблицей?
2. Что такое коллизия и какие методы её разрешения существуют?
3. Как организовать прямую адресацию в хэш-таблице?
4. Какие критерии хорошей хэш-функции?
5. Какие проблемы возникают при плохих хэш-функциях?
6. Что такое равномерное распределение значений хэш-функции?
7. В чём отличие открытых адресов от закрытых адресов в разрешении коллизий?
8. Чем полезны хэш-коды и как они влияют на безопасность данных?
9. Какие функции контроля целостности данных обеспечиваются средствами хэширования?
10. Чем отличается симметричное шифрование от асимметричного?
11. Что такое цифровая подпись и как она защищает данные?
12. Что такое система открытого ключа и как она организована?
13. Как организованы поисковые индексы и что такое обратные индексы?
14. Какие технологии используют поисковые системы для повышения скорости выдачи результатов?
15. Что такое последовательное хэширование и где оно применяется?
16. В чём особенность двойного хэширования?
17. Какие виды хэширования известны и где они используются?
18. Что такое универсальное хэширование и почему оно популярно?
19. Что такое согласованное хэширование и когда оно полезно?
20. В чём состоят потенциальные угрозы безопасности при неправильном выборе хэш-функции?
21. Что такое корень дерева и лист узла?
22. Как определить высоту дерева и диаметр дерева?
23. В чём отличие полных деревьев от почти-полных деревьев?
24. Какая операция необходима для поддержания баланса в дереве?
25. Какими свойствами обладают бинарные деревья поиска (BST)?
26. Что такое сбалансированное дерево и зачем оно нужно?
27. Какие варианты обхода дерева существуют (in-order, pre-order, post-order)?
28. Как реализован класс бинарного дерева поиска на языке C++?
29. В чём отличия бинарного дерева поиска от AVL-дерева?
30. Как обрабатываются дубликаты в бинарном дереве поиска?
31. Что такое очередь с приоритетом и как она реализована?
32. Какие методы применяются для поддержания упорядоченности в структуре данных с приоритетами?
33. Что такое внутреннее представление AVL-дерева?
34. Какие правила сохранения баланса соблюдаются в AVL-дереве?
35. Как осуществляется включение новой вершины в AVL-дерево?
36. Как изменяется высота дерева после включения нового узла?
37. Какие вращения необходимы для восстановления баланса в AVL-дереве?
38. В чём разница между левым и правым вращениями?
39. Как происходит удаление вершины из AVL-дерева?
40. Какие изменения происходят в дереве после удаления вершины?

Практическое занятие №3. Алгоритмы хеширования данных. Построение функций хеширования и алгоритмов хеширования данных, при решении задач на языке C++

Контрольные вопросы (ОПК-8):

1. Что такое хеширование и для чего оно применяется?
2. Какие свойства должны обладать хорошие хеш-функции?
3. В чём опасность возникновения коллизий и как они решаются?
4. Приведите пример простой хеш-функции на языке C++.
5. Чем отличаются методы открытой адресации и цепочек для разрешения коллизий?
6. Как можно повысить качество хеш-функции?
7. Как построить хеш-таблицу с открытым размещением на языке C++?
8. В чём особенность идеальной хеш-функции?
9. Чем отличается хеширование целых чисел от хеширования строковых данных?
10. Какова цель универсального хеширования и как оно реализовано?

Практическое занятие №4. Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Реализация алгоритма построения оптимального бинарного дерева поиска.

Контрольные вопросы (ОПК-8):

1. Что такое бинарное дерево поиска?
2. Какие основные операции можно выполнять с бинарным деревом поиска?
3. Какая последовательность действий выполняется при вставке нового элемента в бинарное дерево поиска?
4. Какую структуру данных целесообразно использовать для представления узлов бинарного дерева поиска в языке C++?
5. Какие случаи возможны при удалении узла из бинарного дерева поиска?
6. В чём отличие поиска в бинарном дереве поиска от линейного поиска?
7. Как построить оптимальный бинарный поисковый алгоритм и зачем он нужен?
8. Как посчитать высоту бинарного дерева поиска?
9. Как выполнить поиск минимального значения в бинарном дереве поиска?
10. В чём заключается идея сбалансированности бинарного дерева поиска и зачем она важна?

Лабораторная работа №1. Реализация алгоритмов хеширования. Составление хеш-таблиц. Построение хеш-таблиц из текстового файла. Построение хеш-таблиц для зарезервированных слов, используемого языка программирования.

Контрольные вопросы (ОПК-8):

1. Что такое хеширование и как оно используется в хеш-таблицах?
2. Какие основные компоненты входят в структуру хеш-таблицы?
3. Что такое коллизия, и какие методы её разрешения существуют?
4. Приведите пример простейшей хеш-функции для строк.
5. Как создать простую хеш-таблицу на языке C++?
6. Какая стратегия хранения используется в случае коллизий методом цепочек?
7. Что такое открытое размещение (открытые адреса) и как оно применяется в хеш-таблицах?
8. Как обеспечить эффективную загрузку хеш-таблицы (коэффициент заполнения)?
9. Какие меры предосторожности надо принять при создании хеш-таблицы из текстового файла?
10. Как автоматизировать создание хеш-таблицы для набора зарезервированных слов конкретного языка программирования (например, C++)?

Лабораторная работа №2. Реализации класса «бинарное дерево поиска». Бинарные деревья поиска. Реализация алгоритма построения оптимального бинарного дерева поиска с квадратичным временем работы.

Контрольные вопросы (ОПК-8):

1. Что такое бинарное дерево поиска (BST)?
2. Какие операции поддерживает бинарное дерево поиска?
3. Какая гарантия правильности сохраняется при вставке новых элементов в бинарное дерево поиска?
4. Какое условие определяет оптимальное бинарное дерево поиска?
5. Почему оптимальная структура дерева влияет на производительность поиска?
6. Какие действия выполняются при вставке элемента в бинарное дерево поиска?
7. Какова вычислительная сложность алгоритма поиска в плохо сбалансированном бинарном дереве?
8. Почему построение оптимального бинарного дерева требует квадратичного времени работы?
9. В чём особенность оптимальной структуры бинарного дерева относительно равномерного распределения элементов?
10. Как изменится алгоритм поиска в бинарном дереве поиска, если оно оптимально построено?

1.3.3.2 Вопросы, выносимые на зачет по дисциплине «Структуры и алгоритмы обработки данных» (ОПК-8)

1. Что такое сложность алгоритма?
2. Как оценивается временная сложность алгоритма?
3. Какие методы используются для оценки временной сложности алгоритма?
4. Назовите наиболее известные алгоритмы сортировки.
5. Какой алгоритм сортировки считается оптимальным по среднему случаю?
6. Объясните принцип пузырьковой сортировки.
7. Опишите алгоритм быстрой сортировки (QuickSort).
8. Как работает метод вставок (Insertion Sort)?
9. Когда целесообразно применять рекурсию?
10. Почему рекурсивные решения иногда приводят к проблемам производительности?
11. Приведите пример классической рекурсивной задачи.
12. Чем отличаются прямые и косвенные рекурсии?
13. Что такое абстрактный тип данных (АТД)? Зачем нужны АТД?
14. Перечислите распространенные примеры АТД.
15. Как реализуются списки в рамках концепции АТД?
16. Приведите реализацию списка с использованием массива.
17. Чем отличается односвязный список от двусвязного?
18. Что понимается под списком с динамическим выделением памяти?
19. Объясните особенности односвязного списка.
20. Что представляет собой очередь с приоритетом?
21. В чём заключается отличие реализаций очередей на массивах и списках?
22. Каковы базовые операции над стеком?
23. Как реализуется стек на статическом массиве?
24. В чём особенность стека на динамической структуре данных?
25. Расскажите о реализации очереди с приоритетом.
26. Что такое дек (deque)? Приведите его применение.
27. Что такое бинарная куча (binary heap)?
28. В чём различие между min-кучей и max-кучей?
29. Что значит поддерживать свойство кучи (heap property)?
30. Опишите шаги процедуры увеличения ключа (increase-key) в бинарной куче.

31. Каково назначение биномиальной кучи (binomial heap)?
32. Чем различаются биномиальная куча и левосторонняя куча (leftist heap)?
33. Что представляет собой косая куча (skew heap)?
34. Объясните различия между операциями слияния (merge) в разных видах куч.
35. Можно ли эффективно реализовать очередь с приоритетом на основе двоичной кучи?
36. Что такое хэширование и как оно связано с хэш-таблицей?
37. Что такое коллизия и какие методы её разрешения существуют?
38. Как организовать прямую адресацию в хэш-таблице?
39. Какие критерии хорошей хэш-функции?
40. Какие проблемы возникают при плохих хэш-функциях?
41. Что такое равномерное распределение значений хэш-функции?
42. В чём отличие открытых адресов от закрытых адресов в разрешении коллизий?
43. Чем полезны хэш-коды и как они влияют на безопасность данных?
44. Какие функции контроля целостности данных обеспечиваются средствами хэширования?
45. Чем отличается симметричное шифрование от асимметричного?
46. Что такое цифровая подпись и как она защищает данные?
47. Что такое система открытого ключа и как она организована?
48. Как организованы поисковые индексы и что такое обратные индексы?
49. Какие технологии используют поисковые системы для повышения скорости выдачи результатов?
50. Что такое последовательное хэширование и где оно применяется?
51. В чём особенность двойного хэширования?
52. Какие виды хэширования известны и где они используются?
53. Что такое универсальное хэширование и почему оно популярно?
54. Что такое согласованное хэширование и когда оно полезно?
55. В чём состоят потенциальные угрозы безопасности при неправильном выборе хэш-функции?
56. Что такое корень дерева и лист узла?
57. Как определить высоту дерева и диаметр дерева?
58. В чём отличие полных деревьев от почти-полных деревьев?
59. Какая операция необходима для поддержания баланса в дереве?
60. Какими свойствами обладают бинарные деревья поиска (BST)?
61. Что такое сбалансированное дерево и зачем оно нужно?
62. Какие варианты обхода дерева существуют (in-order, pre-order, post-order)?
63. Как реализован класс бинарного дерева поиска на языке C++?
64. В чём отличия бинарного дерева поиска от AVL-дерева?
65. Как обрабатываются дубликаты в бинарном дереве поиска?
66. Что такое очередь с приоритетом и как она реализована?
67. Какие методы применяются для поддержания упорядоченности в структуре данных с приоритетами?
68. Что такое внутреннее представление AVL-дерева?
69. Какие правила сохранения баланса соблюдаются в AVL-дереве?
70. Как осуществляется включение новой вершины в AVL-дерево?
71. Как изменяется высота дерева после включения нового узла?
72. Какие вращения необходимы для восстановления баланса в AVL-дереве?
73. В чём разница между левым и правым вращениями?
74. Как происходит удаление вершины из AVL-дерева?
75. Какие изменения происходят в дереве после удаления вершины?

1.3.4.1 Оценочные материалы для заочной формы обучения (ЗФО) (Курс 3, Сессия 3)

Модуль 1 (50 баллов):

Модуль содержит 2 лекционных занятия и 2 лабораторных работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 10. За выполненные и защищенные лабораторные занятия студент получает максимум 20 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы (ОПК-8):

1. Перечислите пять свойств красно чёрного дерева. Какое из них гарантирует сбалансированность?
2. При вставке узла в красно чёрное дерево могут возникнуть три случая нарушения свойств. Опишите каждый случай и действия по их исправлению.
3. Какие виды вращений используются в красно чёрных деревьях?
4. Приведите пример левого поворота вокруг узла.
5. Опишите процесс удаления узла из красно чёрного дерева.
6. Как обрабатываются случаи, когда удаляемый узел чёрный?
7. Что такое дерево промежутков?
8. Как оно организовано и для каких задач применяется?
9. Опишите алгоритм поиска всех интервалов, пересекающих произвольно заданный интервал, в дереве промежутков.
10. В чём ключевое отличие **SPLAY** дерева от других сбалансированных деревьев (**AVL**, красно чёрных)?
11. Какие операции в **SPLAY** дереве называют словарными?
12. Как реализуются операции в **SPLAY**-дереве?
13. Опишите процедуру **splay** (протаскивание) узла к корню.
14. Перечислите возможные случаи вращений (**zig**, **zig zig**, **zig zag**).
15. Как работают операции **split** и **merge** в **SPLAY** дереве? Для чего они нужны?
16. Опишите алгоритм вставки нового узла в **SPLAY** дерево.
17. Зачем в **SPLAY** деревьях нужны вспомогательные процедуры для работы с указателями на родителей? Как они поддерживаются?
18. Как выполняется операция **find** в **SPLAY** дереве? Что происходит с найденным узлом?
19. Сравните амортизированную сложность основных операций **SPLAY** деревьев с красно чёрными деревьями.
20. Что такое **B** дерево? Назовите его основные параметры (минимальную степень **t**, максимальное количество ключей в узле).
21. Опишите процесс вставки ключа в **B** дерево.
22. Когда и как происходит расщепление узлов?
23. Опишите процесс удаления ключа из **B** дерева. Какие случаи требуют слияния или заимствования ключей?
24. Почему **B** деревья широко применяются в базах данных и файловых системах?
25. Как в **B** дереве выполняется исправление потенциальной разбалансировки на этапе поиска удаляемого элемента?
26. Что такое биномиальное дерево **B_k**? Перечислите его свойства (количество узлов, высота, степень).
27. Какие основные операции поддерживает биномиальная куча?
28. Оцените их временную сложность.
29. Опишите алгоритм слияния (**merge**) двух биномиальных куч.
30. В чём преимущества биномиальной кучи перед обычной двоичной? В чём недостатки?

31. Что такое куча Фибоначчи? Какие операции в ней имеют лучшие амортизированные оценки по сравнению с биномиальной кучей?
32. Опишите структуру кучи Фибоначчи: корневой список, каскадная обрезка, пометки узлов.
33. Как выполняется операция `extract_min` в куче Фибоначчи? Какие шаги необходимы и как достигается амортизированная сложность $O(\log n)$?
34. Что такое фибоначчьево дерево? Как оно связано с кучей Фибоначчи?
35. Предложите сценарии, в которых использование `SPLAY` дерева предпочтительнее красно-чёрного, и наоборот. Аргументируйте ответ.

Лабораторная работа №1. Программная реализация красно-черных деревьев. Добавление и удаление вершин.

Контрольные вопросы (ОПК-8):

1. Какие свойства красно-черного дерева нарушаются при вставке нового узла?
2. Опишите алгоритм удаления узла из красно-черного дерева.
3. Какие дополнительные действия требуются, если удаляемый узел черный?
4. Как выполняется балансировка после удаления?
5. Как проверить корректность красно-черного дерева после вставки и удаления?
6. Что такое дерево промежутков (`interval tree`)?
7. Какая дополнительная информация хранится в каждом узле и как она используется при поиске пересекающихся интервалов?
8. Опишите процесс вставки интервала в дерево промежутков. Как поддерживается поле максимального правого конца в узлах при вставке и удалении?
9. Использовались рекурсивные или итеративные алгоритмы для операций балансировки?
10. Какие преимущества и недостатки каждого подхода вы можете выделить?

Лабораторная работа №2. Программная реализация `SPLAY`-деревьев. Построение двух корректных деревьев поиска, построенные на одном и том же множестве ключей друг от друга последовательностью поворотов.

Контрольные вопросы (ОПК-8):

1. Какие основные операции поддерживает `SPLAY`-дерево? Опишите алгоритм операции `splay` и её роль в работе дерева.
2. В чем отличие `SPLAY`-дерева от других сбалансированных деревьев (например, AVL или красно-черных) с точки зрения гарантий времени работы?
3. Как выполняется вставка нового ключа в `SPLAY`-дерево? Приведите последовательность шагов.
4. Какие возможны случаи вращений (`zig`, `zig-zig`, `zig-zag`) при выполнении операции `splay`? Приведите примеры для каждого случая.
5. Объясните, как реализуются операции `split` и `merge` в `SPLAY`-дереве. Для каких целей они используются?
6. Как в вашей реализации организовано хранение указателей на родительские узлы? Почему это важно для операции `splay`?
7. В задании требуется построить два корректных дерева поиска на одном и том же множестве ключей, которые можно преобразовать друг в друга последовательностью поворотов. Докажите, что для любых двух бинарных деревьев поиска с одинаковым набором ключей существует такая последовательность.
8. Опишите алгоритм преобразования одного дерева поиска в другое с помощью поворотов (вращений). Какие типы поворотов (левый, правый) используются?
9. Приведите пример двух деревьев поиска, построенных на множестве ключей $\{1,2,3,4,5\}$, и покажите последовательность поворотов, переводящую одно дерево в другое.
10. Какие сложности могут возникнуть при реализации преобразования деревьев с помощью поворотов? Как вы проверяли корректность вашей реализации?

Модуль 2 (50 баллов):

Модуль содержит 1 лекционное занятие и 2 лабораторные работы. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 10. За выполненные и защищенные лабораторные занятия студент получает максимум 20 баллов. Общее максимальное количество баллов за модуль 2 составляет 50 баллов.

Вопросы для контрольной работы (ОПК-8):

1. Дайте определение графа. Чем ориентированный граф отличается от неориентированного?
2. Что такое степень вершины? Приведите формулу суммы степеней.
3. Какие графы называют связными? Приведите пример несвязного графа.
4. Что такое цикломатическое число графа? Как оно вычисляется?
5. Как найти максимальное остовное дерево взвешенного графа?
6. Опишите структуру данных «система непересекающихся множеств» (DSU).
7. Какие операции она поддерживает?
8. В чём разница между реализацией DSU на массиве и на лесе (с эвристиками)?
9. Что такое частично упорядоченное множество?
10. Приведите пример отношения порядка.
11. Какие операции можно выполнять над частично упорядоченным множеством?
12. Опишите алгоритм поиска в ширину (BFS). Для каких задач он применяется?
13. Чем поиск в глубину (DFS) отличается от BFS? Приведите пример использования DFS.
14. Что такое минимальное покрывающее дерево (MST)? Назовите два классических алгоритма его построения.
15. Как работает алгоритм Краскала для MST?
16. Как работает алгоритм Прима для MST?
17. Для решения каких задач применяют алгоритмы поиска кратчайшего пути на взвешенном графе?
18. Опишите алгоритм Дейкстры. Какое ограничение на веса рёбер у него есть?
19. Для каких графов используют алгоритм Беллмана-Форда?
20. Что такое цифровой поиск? В чём его отличие от обычного бинарного поиска?
21. Опишите структуру цифрового дерева (trie) для набора строк.
22. Что такое хеширование с открытой адресацией?
23. Перечислите основные методы разрешения коллизий.
24. Какие требования предъявляются к хеш-функции для открытой адресации?
25. Как выполняется поиск элемента в хеш-таблице с открытой адресацией?
26. Какие типы данных можно хешировать?
27. Приведите примеры хеш-функций для чисел и строк.
28. Что такое обобщение цифрового поиска?
29. В каких структурах оно используется?
30. Как можно комбинировать цифровой поиск с поиском в последовательных таблицах?
31. Приведите пример задачи, где эффективно использовать сочетание цифрового поиска и хеширования.
32. Что такое «лес» в контексте DSU? Как он ускоряет операции?
33. Как вычислить цикломатическое число для связного графа?
34. В чём заключается эвристика «сжатие пути» в DSU?
35. Как найти компоненты связности графа с помощью BFS или DSU?

Лабораторная работа №3. Задачи на графах. Построение матрицы смежности. Построение матрицы инцидентности.

Контрольные вопросы (ОПК-8):

1. Как в вашей программе задаётся граф: список рёбер, матрица смежности или иной формат?
2. Опишите алгоритм построения матрицы смежности по списку рёбер.
3. Какие размерности имеет матрица инцидентности для неориентированного графа?
4. Как отличаются матрицы инцидентности для ориентированного и неориентированного графов?
5. Как реализовано заполнение матрицы инцидентности: что записывается в ячейку при инцидентности вершины и ребра?
6. Какие структуры данных выбраны для хранения матриц (список списков, `map`, массив)?
7. Как проверяется корректность ввода графа перед построением матриц?
8. Какие операции над графом вы реализовали с использованием матрицы смежности (поиск соседей, проверка связности)?
9. Как выводится матрица смежности и матрица инцидентности на экран (форматирование)?
10. Какие ограничения по размеру графа накладывает выбранное представление?

Лабораторная работа №4. Реализация с использованием списков. Построение списков связи. Построение перечней рёбер.

Контрольные вопросы (ОПК-8):

1. Как организовать список смежности на основе массива списков?
2. Как добавить ребро в список смежности для ориентированного графа?
3. Как удалить ребро из списка смежности?
4. Как проверить наличие ребра между двумя вершинами в списке смежности?
5. Как построить перечень рёбер по готовому списку смежности?
6. Как по перечню рёбер восстановить список смежности?
7. Как оценить память, занимаемую списком смежности и перечнем рёбер?
8. Как реализовать обход графа в глубину, используя только список смежности?
9. Как учесть кратные рёбра при построении списков связи?
10. Какие операции с графом эффективнее выполнять на списках смежности, а какие – на перечне рёбер?

1.3.4.2 Вопросы, выносимые на экзамен по дисциплине «Структуры и алгоритмы обработки данных» (ОПК-8)

1. Что такое сложность алгоритма?
2. Как оценивается временная сложность алгоритма?
3. Назовите наиболее известные алгоритмы сортировки.
4. Опишите алгоритм быстрой сортировки (QuickSort).
5. Когда целесообразно применять рекурсию?
6. Что такое абстрактный тип данных (АТД)? Зачем нужны АТД?
7. Чем отличается односвязный список от двусвязного?
8. Что представляет собой список с динамическим выделением памяти?
9. Каковы базовые операции над стеком?
10. Как реализуется очередь с приоритетом?
11. Что такое бинарная куча (binary heap)?
12. В чём различие между `min`-кучей и `max`-кучей?
13. Что такое хэширование и как оно связано с хэш-таблицей?
14. Что такое коллизия и какие методы её разрешения существуют?
15. Какие критерии хорошей хэш-функции?
16. Что такое корень дерева и лист узла?

17. Какими свойствами обладают бинарные деревья поиска (BST)?
18. Какие варианты обхода дерева существуют (in-order, pre-order, post-order)?
19. Что такое сбалансированное дерево и зачем оно нужно?
20. В чём отличия бинарного дерева поиска от AVL-дерева?
21. Какие вращения необходимы для восстановления баланса в AVL-дереве?
22. Перечислите пять свойств красно-чёрного дерева. Какое из них гарантирует сбалансированность?
23. При вставке узла в красно-чёрное дерево могут возникнуть три случая нарушения свойств. Опишите каждый случай и действия по их исправлению.
24. Какие виды вращений используются в красно-чёрных деревьях? Приведите пример левого поворота.
25. Опишите процесс удаления узла из красно-чёрного дерева. Как обрабатываются случаи, когда удаляемый узел чёрный?
26. Почему красно-чёрные деревья гарантируют высоту $O(\log n)$, в отличие от обычных BST?
27. В каких сценариях красно-чёрное дерево предпочтительнее AVL-дерева, и наоборот?
28. Опишите процедуру splay (протаскивание) узла к корню.
29. Что такое B-дерево? Назовите его основные параметры (минимальную степень t , максимальное количество ключей в узле).
30. Почему B-деревья широко применяются в базах данных и файловых системах?
31. Что такое биномиальное дерево B_k ? Перечислите его свойства.
32. Опишите алгоритм слияния (merge) двух биномиальных куч.
33. Что такое куча Фибоначчи? Какие операции в ней имеют лучшие амортизированные оценки по сравнению с биномиальной кучей?
34. Дайте определение графа. Чем ориентированный граф отличается от неориентированного?
35. Что такое степень вершины? Приведите формулу суммы степеней.
36. Какие графы называют связными? Приведите пример несвязного графа.
37. Опишите структуру данных «система непересекающихся множеств» (DSU). Какие операции она поддерживает?
38. В чём разница между реализацией DSU на массиве и на лесе (с эвристиками)?
39. Опишите алгоритм поиска в ширину (BFS). Для каких задач он применяется?
40. Чем поиск в глубину (DFS) отличается от BFS? Приведите пример использования DFS.
41. Что такое минимальное покрывающее дерево (MST)? Назовите два классических алгоритма его построения.
42. Как работает алгоритм Краскала для MST?
43. Как работает алгоритм Прима для MST?
44. Опишите алгоритм Дейкстры. Какое ограничение на веса рёбер у него есть?
45. Для каких графов используют алгоритм Беллмана-Форда?
46. Что такое цифровой поиск? В чём его отличие от обычного бинарного поиска?
47. Опишите структуру цифрового дерева (trie) для набора строк.
48. Что такое хеширование с открытой адресацией? Перечислите основные методы разрешения коллизий.
49. Как организовать список смежности для графа?
50. Как по списку смежности построить перечень рёбер?
51. Как реализовать построение остовного дерева обходом в глубину (или ширину)?
52. Как реализовать алгоритм Флойда-Уоршелла для матрицы кратчайших путей?
53. Что такое многоуровневые высечки при обработке словаря?
54. Как обеспечить откат (undo) последней высечки в словаре?
55. Сравните амортизированную сложность основных операций SPLAY-деревьев с красно-чёрными деревьями.

Модуль «Контрольная работа» (100 баллов):

Темы заданий контрольной работы:

1. Алгоритмы сортировки элементов массива и исследование их эффективности.
2. Статические структуры данных. Линейный список. Массив.
3. Полустатические структуры данных. Стеки. Очереди.
4. Динамические структуры данных. Деревья. Графы.

Для выполнения контрольной работы обучающиеся получают в библиотеке печатный экземпляр или скачивают в ЛК электронное издание учебно-методического пособия: «Методические указания по выполнению контрольной работы по дисциплине «Структуры и алгоритмы обработки данных».

Модуль посвящен выполнению задания на контрольную работу. Максимальное количество баллов за выполненную и защищенную контрольную работу составляет 100.

Контрольная работа выполняется в процессе освоения компетенции (ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения), а итоговая оценка за её защиту характеризует заключительный этап их освоения.

Критерии оценивания контрольной работы приведены в таблице.

Таблица – Критерии оценивания контрольной работы

№п/п	Оцениваемые параметры	Оценка в баллах
1.	Корректность реализации программного кода:	40
	– Программа компилируется и выполняется без ошибок.	10
	– Алгоритмы сортировки работают правильно на разных входных данных.	10
	– Операции со статическими, полустатическими и динамическими структурами реализованы без логических ошибок.	10
	– Обработка граничных случаев (пустой массив, стек/очередь, дерево/граф без вершин и т.д.).	10
2.	Эффективность и выбор алгоритмов:	30
	– Использование подходящих алгоритмов сортировки с учётом размера данных.	8
	– Оптимальная реализация стека/очереди (без лишних копирований, с учётом переполнения/опустошения).	8
	– Выбор между статическим и динамическим представлением данных обоснован.	6
	– Алгоритмы на графах и деревьях имеют заявленную сложность.	8
3.	Оформление результатов контрольной работы:	15
	– контрольная работа оформлена в полном соответствии с ЕСКД;	15
	– контрольная работа оформлена с незначительными отклонениями от ЕСКД;	10
	– оформление контрольной работы не соответствует требованиям ЕСКД.	5
4.	Качество ответов на вопросы в процессе защиты контрольной работы:	15
	– Студент может объяснить выбор структур данных и алгоритмов.	5
	– Отвечает на вопросы о сложности операций, преимуществах/недостатках реализованных структур.	5
	– Вносит изменения в код по требованию преподавателя.	5
	– Допускает грубые ошибки при ответе или не может четко ответить на вопросы.	0

Студент, набравший по контрольной работе:

- от 0 до 40 баллов, получает оценку «неудовлетворительно»;
- от 41 до 60 баллов, получает оценку «удовлетворительно»;
- от 61 до 80 баллов, получает оценку «хорошо»;
- от 81 до 100 баллов, получает оценку «отлично».

2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

2.1 Порядок и методика проведения текущего контроля и промежуточной аттестации

Предусмотрены следующие виды контроля:

- текущий контроль по каждому модулю – в форме написания контрольной работы для оценки теоретических знаний;
- текущий контроль по каждому модулю – в форме отчетов по лабораторным работам и практическим занятиям для оценки практических навыков;
- промежуточная аттестация по дисциплине – в форме зачета или экзамена.

Текущий контроль успеваемости в форме защиты лабораторных и практических проводится по окончании каждой лабораторной работы (практического занятия) в форме индивидуального собеседования по выполненным исследованиям или расчетам. Проводится с целью контроля закрепления теоретической части материала и степени отработки студентом практических навыков исследования на аппаратуре.

Контрольные работы выполняются в виде короткого письменного ответа на один вопрос, изученный на предыдущей лекции в начале каждой последующей лекции. Ответ на вопрос дается в течение 5-10 минут. Таким образом, после лекционного курса каждого модуля формируется общая оценка за теоретические знания.

С целью повышения качества обучения за счет побуждения студентов к активной текущей учебной работе, четкого и оперативного контроля всего хода учебного процесса, снижения роли случайных и субъективных факторов при оценивании учебной деятельности студентов в образовательном процессе реализована модульно-рейтинговая система.

Правила ее использования прописаны в «Положении об МРС».

Набранные обучающимся в течение текущего контроля баллы могут быть переведены в оценку (по дисциплине предусмотрен зачет на 2 курсе):

- «не зачтено» - от 0 до 40 от максимального количества баллов;
- «зачтено» - от 41 до 100 максимального количества баллов.

Набранные обучающимся в течение текущего контроля баллы могут быть переведены в оценку (по дисциплине предусмотрен экзамен на 3 курсе):

- «неудовлетворительно» - от 0 до 40 от максимального количества баллов;
- «удовлетворительно» - от 41 до 60 максимального количества баллов;
- «хорошо» - от 61 до 80 максимального количества баллов;
- «отлично» - от 81 до 100 максимального количества баллов.

Соотношения максимального количества баллов, полученных студентом по блокам модулей, показаны в Таблицах.

Таблица - Распределение баллов по блокам модулей дисциплины «Структуры и алгоритмы обработки данных» для очной формы обучения (ОФО)

Модуль	Всего баллов (Максимальное значение)	Теоретический блок (Распределение баллов по средствам оценивания)	Практический блок (Распределение баллов по средствам оценивания)
Курс 2, Семестр 4			
Модуль 1	50	15	35=15+20
Модуль 2	50	15	35=14+21
Зачет	100	100	
Курс 3, Семестр 5			
Модуль 1	50	10	40=20+20
Модуль 2	50	10	40=20+20
Экзамен	100	100	

Таблица - Распределение баллов по блокам модулей дисциплины «Структуры и алгоритмы обработки данных» для заочной формы обучения (ЗФО)

Модуль	Всего баллов (Максимальное значение)	Теоретический блок (Распределение баллов по средствам оценивания)	Практический блок (Распределение баллов по средствам оценивания)
Курс 2, Сессия 3			
Модуль 1	50	20	30=15+15
Курс 3, Сессия 2			
Модуль 2	50	10	40=20+20
Зачет	100	100	
Курс 3, Сессия 3			
Модуль 1	50	10	40=20+20
Модуль 2	50	10	40=20+20
Экзамен	100	100	

Как правило, теоретический блок оценивается по результатам контрольной работы. Практический блок оценивается по результатам выполнения лабораторных работ и заданий на практических занятиях.

На зачете и экзамене производится оценка тех компетенций, которые должны быть в той или иной форме освоены в процессе изучения. Рекомендуется формировать вопросы в экзаменационных билетах/в билетах к зачету таким образом, чтобы преподаватель смог оценить все компетенции данной дисциплины.

2.2 Методика проведения зачета и экзамена в группах очной и заочной форм

Экзамен и зачет по дисциплине «Структуры и алгоритмы обработки данных» у студентов очной и заочной форм обучения проводится письменно по классической методике. Вопросы сгруппированы в билетах. В каждом билете содержится по 2-3 вопроса, скомпонованные таким образом, чтобы преподаватель смог оценить все компетенции данной дисциплины. Количество билетов должно быть не менее числа студентов в группе. Студенты готовят письменные ответы на вопросы.

3 Промежуточная аттестация

Промежуточная аттестация проводится в форме зачета:

- для очной формы обучения (ОФО) на втором курсе в 4 семестре;
- для заочной формы обучения (ЗФО) на третьем курсе, сессия 2.

Зачет проводится по расписанию экзаменационной сессии в письменном виде. Количество вопросов в билете к зачету – 2. Проверка ответов и объявление результатов производится в день зачета. Результаты аттестации заносятся в зачетную ведомость и зачетную книжку студента.

Студенты, не прошедшие промежуточную аттестацию по графику сессии, должны ликвидировать задолженность в установленном порядке.


Промежуточная аттестация проводится в форме экзамена:

- для очной формы обучения (ОФО) на третьем курсе в 5 семестре;
- для заочной формы обучения (ЗФО) на третьем курсе, сессия 3.

Экзамен проводится по расписанию экзаменационной сессии в письменном виде. Количество вопросов в экзаменационном билете – 3. Проверка ответов и объявление результатов производится в день экзамена. Результаты аттестации заносятся в экзаменационную ведомость и зачетную книжку студента.

Студенты, не прошедшие промежуточную аттестацию по графику сессии, должны ликвидировать задолженность в установленном порядке.

Образец билета к зачету

	<p>МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ Северо-Кавказский филиал ордена Трудового Красного Знамени федерального государственного бюджетного образовательного учреждения высшего образования «Московский технический университет связи и информатики»</p>	<p>Утверждаю Зав. кафедрой «ИВТ» _____ Соколов С.В. «___» _____ 20__ г.</p>
<p>Направление подготовки: 09.03.01. Информатика и вычислительная техника Курсы: 2 Дисциплина: Структуры и алгоритмы обработки данных</p>		
<p style="text-align: center;">Билет №1</p> <p>1. Что такое абстрактный тип данных (АТД)? Зачем нужны АТД?</p> <p>2. Приведите реализацию списка с использованием массива.</p> <p style="text-align: right;">ст. преподаватель кафедры ИВТ _____ Осмоловская Н.С. «___» _____ 20__ г.</p>		

Образец экзаменационного билета



МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,
СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Северо-Кавказский филиал ордена Трудового Красного Знамени
федерального государственного бюджетного образовательного
учреждения высшего образования
«Московский технический университет связи и информатики»

Утверждаю
Зав. кафедрой «ИВТ»
_____ Соколов С.В.
« ____ » _____ 20 ____ г.

Направление подготовки: **09.03.01. Информатика и вычислительная техника**

Курсы: 2

Дисциплина: Структуры и алгоритмы обработки данных

Билет №1

1. Перечислите пять свойств красно-чёрного дерева. Какое из них гарантирует сбалансированность?
2. Что такое минимальное покрывающее дерево (MST)? Назовите два классических алгоритма его построения.
3. Реализовать построение остовного дерева обходом в глубину (или ширину).

_____ ст. преподаватель кафедры ИВТ _____ Осмоловская Н.С.
« ____ » _____ 20 ____ г.

Один комплект отпечатанных билетов к зачету (экзамену), подписанных преподавателем кафедры и утвержденных заведующим кафедрой хранится у заведующего кафедрой, другой комплект – у преподавателя, ведущего дисциплину.

4 Тестовые задания для проведения оценки сформированности компетенций (ОПК-8)

Данная дисциплина формирует у обучающегося ОПК-8 компетенцию, объективно оценить которую можно при ответе студента на тест или при выполнении практического или лабораторного задания.

Один комплект тестовых вопросов с указанием правильных ответов хранится у заведующего кафедрой, другой комплект – у преподавателя, ведущего дисциплину.

Тестовые задания позволяют оценить уровень сформированности компетенции ОПК-8.

Задания распределены по блокам в соответствии с уровнем сложности. Каждый блок содержит номер задания, и текст задания.

Базовый уровень содержит примерно 50 % заданий.

Он формируется из заданий с выбором одного или нескольких верных ответов из предложенных или воспроизведения фактического материала (терминология, факты, классификация, параметры и др.);

Повышенный уровень в среднем составляет 35 % заданий.

В нем используются задания на сопоставление, сравнение, установление последовательности;

Высокий уровень в среднем составляют – 15 % заданий.

В задании используются: решения нетиповых задач, алгоритмы, доказательства, задания с развернутым ответом – определения, перечисление, изображение схемы, структуры и др.

Количество заданий по каждой компетенции равно 20. Всего 40 заданий.

Ключи к ответам представлены в отдельной таблице.

ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

БЛОК А (базовый уровень) – задание с выбором одного или нескольких верных ответов из предложенных.

Инструкция по тестам Блока А: Прочитайте текст и выберите один или несколько правильных ответов.

№ задания	Тексты заданий
1	Какой из следующих вариантов правильно обращается к первому элементу массива указателей? 1. ptrArray(0) 2. *ptrArray[0] 3. &ptrArray[0] 4. ptrArray[0]
2	Какое из представленных ниже утверждений НЕ верно для двунаправленных списков? 1. значения указателей некоторых элементов могут быть пустыми 2. каждый элемент имеет два указателя на соседние элементы 3. переходы между указателями возможны в двух направлениях 4. указатель последнего элемента показывает на первый элемент списка
3	Укажите возможный способ реализации стека: 1. в виде трехмерного массива 2. в виде двумерного массива 3. в виде линейного однонаправленного списка 4. в виде циклического однонаправленного списка
4	Для структуры данных дерево НЕ верны следующие утверждения: 1. существует узел, в который не входит ни одной дуги 2. не существует вершин, из которых не выходит ни одной дуги 3. в каждую вершину входит одна дуга 4. существуют вершины, в которые входят две или более дуг
5	В структуре данных дерево вершина со всеми ее потомками называется? 1. узел 2. поддерево 3. корень 4. очередь
6	Степень дерева определяется: 1. максимальной степенью вершины дерева 2. усредненной степенью вершины дерева 3. степенью корня 4. суммой вершин
7	При динамической реализации двоичного дерева в виде списка каждый элемент дерева имеет: 1. одно поле данных и одно поле указателя 2. одно поле данных и два поля указателей 3. два поля данных и одно поле указателя 4. два поля данных и два поля указателей
8	Какая сортировка называется иначе сортировкой с уменьшающимися расстояниями? 1. сортировка подсчетом

№ задания	Тексты заданий
	2. сортировка включением 3. сортировка Шелла 4. древесная сортировка
9	Временная сложность алгоритма бинарного (двоичного) поиска составляет: 1. $O(n)$ 2. $O(n \log n)$ 3. $O(\log n)$ 4. $O(1)$
10	Укажите виды хеширования данных: 1. хеширование с открытой адресацией 2. хеширование с линейной выборкой 3. хеширование с цепочками (метод цепочек) 4. хеширование с бинарным деревом

БЛОК Б (повышенный уровень) – задание закрытого типа на установление соответствия
Инструкция: Прочитайте текст и установите соответствие.

№ задания	Тексты заданий
11	Установите соответствие между структурами данных и их основным назначением: 1. Множества 2. Хэш-таблица 3. Двоичное дерево поиска 4. Граф <ul style="list-style-type: none"> А) быстрый доступ к данным по ключу (например, кэш результатов вычислений). Б) поддержка упорядоченных данных с динамическими операциями (например, телефонный справочник с возможностью вывода в отсортированном порядке). В) моделирование связей между объектами (например, поиск кратчайшего пути в транспортной сети). Г) проверка уникальности элементов (например, хранение идентификаторов посетителей сайта за день).
12	Установите соответствие между способом реализации графа и его описанием: 1. Матрица смежности 2. Список смежности 3. Перечень рёбер 4. Матрица инцидентности <ul style="list-style-type: none"> А) Матрица размера $V \times E$, где отмечается принадлежность вершины ребру (для ориентированных графов – с указанием направления). Б) Список (или массив) всех рёбер, каждое задаётся парой (или тройкой с весом) вершин. В) Квадратная матрица, где элемент $[i][j]$ указывает на наличие (и вес) ребра между вершинами i и j. Г) Массив списков: для каждой вершины хранятся соседние вершины (и веса).
13	Установите соответствие между алгоритмом на графах и его назначением: 1. Алгоритм Дейкстры 2. Алгоритм Крускала 3. Алгоритм Прима

№ задания	Тексты заданий
	<p>4. Алгоритм Флойда</p> <p>А) Построение минимального остовного дерева (MST) путём добавления рёбер с минимальным весом, избегая циклов (с использованием DSU).</p> <p>Б) Поиск кратчайших путей между всеми парами вершин (динамическое программирование).</p> <p>В) Построение минимального остовного дерева (MST) путём постепенного наращивания дерева от начальной вершины.</p> <p>Г) Поиск кратчайших путей от одной вершины до всех остальных в графе с неотрицательными весами рёбер.</p>
14	<p>Установите соответствие между обозначением сложности алгоритма и её названием (или описанием).</p> <p>1. $O(1)$</p> <p>2. $O(\log n)$</p> <p>3. $O(n)$</p> <p>4. $O(n^2)$</p> <p>А) Линейная сложность</p> <p>Б) Квадратичная сложность</p> <p>В) Константная сложность</p> <p>Г) Логарифмическая сложность</p>

БЛОК Б (повышенный уровень) – задание закрытого типа на установление последовательности.

Инструкция: Прочитайте текст и установите правильную последовательность.

№ задания	Тексты заданий
15	<p>Представлено двоичное дерево. Перечислите все его узлы при обходе снизу.</p> <pre> d ├── └── b f ├── └── a c e g </pre>
16	<p>Установите правильную последовательность действий при написании программы с деревом на C++.</p> <ol style="list-style-type: none"> 1. Определение структуры узла. 2. Реализация публичных методов-обёрток. 3. Написание функций для обхода дерева (прямой, симметричный, обратный) и вывода данных. 4. Реализация конструкторов и деструктора. 5. Реализация операций, специфичных для типа дерева (балансировка, вращения, поддержка свойств). 6. Реализация вспомогательных рекурсивных методов (вставка, поиск, удаление, обходы). 7. Создание класса дерева.
17	<p>Установите правильную последовательность пунктов при написании программы на C++:</p> <ol style="list-style-type: none"> 1. Определение основной функции main() 2. Заголовочный комментарий и директива include 3. Прототипы функций и объявлений классов

№ задания	Тексты заданий
	4. Реализация классов и методов 5. Тестирование и отладка 6. Подключение нужных библиотек 7. Определение класса и его методов

БЛОК В (высокий уровень) – задание открытого типа с развернутым ответом.

Инструкция: Прочитайте текст и запишите обоснованный ответ.

№ задания	Тексты заданий
18	Программа хранит узел дерева Фибоначчи и саму фибоначчьеву кучу. Расставьте строки программы в правильном порядке. 1. <code>typedef bool (*cmpPtr)(node*, node*);</code> 2. <code>struct node {</code> 3. <code>bool mark;</code> 4. <code>int degree;</code> 5. <code>int key;</code> 6. <code>node* left;</code> 7. <code>node* child;</code> 8. <code>node* right;</code> 9. <code>node* parent;</code> 10. <code>};</code> 11. <code>struct fib_heap {</code> 12. <code>node* min;</code> 13. <code>cmpPtr less;</code> 14. <code>int roots_amount;</code> 15. <code>}</code>
19	Какой алгоритм находит кратчайшие пути между всеми парами вершин графа? Алгоритм _____.
20	Ациклический неориентированный граф называется _____.

Система оценивания выполнения тестовых заданий по компетенции ОПК-8.

Номер задания	Указания по оцениванию	Результат оценивания (баллы, полученные за выполнение задания/характеристика правильности ответа)
Задания 1-10	Задания с выбором одного или нескольких верных ответов из предложенных считается верным, если правильно указана цифра или цифры, означающие верные ответы.	Совпадение с верным ответом оценивается 1 баллом; неверный ответ или его отсутствие - 0 баллов.
Задания 11-14	Задания закрытого типа на установление соответствия считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)	Полное совпадение с верным ответом оценивается 1 баллом; неверный ответ или его отсутствие - 0 баллов.

Задания 15-17	Задания закрытого типа на установление последовательности считается верным, если правильно указана вся последовательность цифр	Полное совпадение с верным ответом оценивается 1 баллом; если допущены ошибки или ответ отсутствует - 0 баллов.
Задания 18-20	Задание открытого типа с развернутым ответом считается верным, если ответ совпадает с эталонным по содержанию и полноте.	Полный правильный ответ на задание оценивается 3 баллами; если допущена одна ошибка/неточность/ответ правильный, но не полный - 1 балл, если допущено более одной ошибки/ответ неправильный/ ответ отсутствует- 0 баллов.

Методика проведения тестирования

Максимальное время выполнения теста - 60 минут.

Количество тестовых заданий в тесте не менее 60.

Для проведения контроля остаточных знаний компьютер формирует в случайном порядке 60 тестовых заданий.

При прохождении тестового контроля знаний студент должен выбрать один или несколько правильных ответов в каждом тестовом задании.

В случае правильного ответа на 49 и более тестовых заданий (81-100% правильных ответов) студент подтверждает освоение компетенций с уровнем «отлично». При ответе на 48-37 тестовых заданий (61-80% правильных ответов) студент подтверждает освоение компетенций с уровнем «хорошо». При ответе на 36-25 тестовых заданий (41-60% правильных ответов) студент подтверждает освоение компетенций с уровнем «удовлетворительно». При ответе менее чем на 25 тестовых заданий (0-40% правильных ответов) студент не подтверждает необходимый уровень знаний и оценивается «неудовлетворительно».

Методические рекомендации по освоению дисциплины

1. Методические рекомендации преподавателю

Перед началом изучения дисциплины преподаватель должен ознакомить студентов с рабочей программой и оценочными материалами по дисциплине, с видами учебной и самостоятельной работы, перечнем литературы и интернет-ресурсов, с формами текущей и промежуточной аттестации, с критериями оценки качества знаний для итоговой оценки по дисциплине.

При проведении лекций, преподаватель:

- 1) формулирует тему и цель занятия, объявляет учебные вопросы;
- 2) излагает основные теоретические положения;
- 3) с помощью технических средств обучения и/или под запись дает определения основных понятий, расчетных формул;
- 4) проводит примеры из отечественного и зарубежного опыта, дает текущие статистические данные для наглядного и образного представления изучаемого материала;
- 5) в конце занятия выдает вопросы для самостоятельного изучения.

На занятиях лабораторного цикла следует обратить внимание на соответствие выбираемых студентом средств выполнения решаемым в работе задачам.

Каждая лабораторная работа должна быть оформлена и защищена в соответствии с требованиями. Защита производится после оформления отчета по работе.

Во время выполнения заданий в учебной аудитории студент может консультироваться с преподавателем, определять наиболее эффективные методы решения поставленных задач. Если какая-то часть задания остается не выполненной, студент может продолжить её выполнение во время внеаудиторной самостоятельной работы.

Для оценки полученных знаний и освоения учебного материала по каждому разделу и в целом по дисциплине преподаватель использует формы текущего контроля и промежуточной аттестации знаний обучающихся.

2. Методические рекомендации студентам

Приступая к изучению новой учебной дисциплины, студенты должны ознакомиться с рабочей программой, оценочными материалами, учебной, научной и методической литературой, имеющейся в ЭИОС СКФ МТУСИ,

Глубина усвоения дисциплины зависит от активной и систематической работы студента на лекциях и практических занятиях, а также в ходе самостоятельной работы, по изучению рекомендованной литературы.

2.1 Методические указания для обучающихся по лекционным занятиям

Важно сосредоточить внимание на содержании лекции. Это поможет лучше воспринимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине.

Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющей материал

прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы.

Конспектирование читаемых лекций и их последующая доработка способствует более глубокому усвоению знаний, и поэтому являются важной формой учебной деятельности студентов.

2.2 Методические указания для обучающихся по подготовке к лабораторным работам

Лабораторные занятия по любой учебной дисциплине проводятся в соответствии с рабочей программой этой учебной дисциплины в установленные расписанием часы.

Инструктаж по технике безопасности, правилам пожарной безопасности и правилам внутреннего распорядка в лаборатории проводится преподавателем на первом лабораторном занятии. На этом же занятии студентам сообщаются:

- программа всего предстоящего лабораторного цикла;
- условия взаимодействия студентов с преподавателем в процессе выполнения лабораторных работ;
- условия контроля самостоятельной работы студентов, включая правила оформления отчетов по лабораторным работам и их последующей защиты;
- другая необходимая информация.

Организация лабораторных работ включает:

- самостоятельную внеаудиторную подготовку студента к выполнению каждой отдельной лабораторной работы в соответствии с ее программой в рамках часов, выделенных на самостоятельную проработку материала в программе дисциплины;
- входной контроль преподавателем степени подготовленности каждого студента к выполнению лабораторных работ;
- выполнение программы лабораторных работ в полном объеме;
- оформление отчета и его защиту каждым студентом в установленные сроки;
- формирование преподавателем рейтингов каждого из студентов по результатам выполнения и защиты им отдельных лабораторных работ и их циклов (используется Модульно-рейтинговая система).

Оценка качества выполнения лабораторных работ каждым студентом производится преподавателем отдельно за подготовку к работе, ее выполнение и защиту.

2.3 Методические указания для обучающихся по подготовке к практическим занятиям

Целью практических занятий является закрепление теоретических знаний, полученных при изучении дисциплины.

При подготовке к практическому занятию целесообразно выполнить следующие рекомендации:

- изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т.д.;
- при необходимости доработать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.

При выполнении практических занятий основным методом обучения является самостоятельная работа студента под управлением преподавателя. На них пополняются

теоретические знания студентов, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение студентов к будущей профессиональной деятельности.

Оценка выполненной работы осуществляется преподавателем комплексно: по результатам выполнения заданий, устному сообщению и оформлению работы.

После подведения итогов занятия студент обязан устранить недостатки, отмеченные преподавателем при оценке его работы.

Методические указания по выполнению лабораторных и практических работ по дисциплине «Структуры и алгоритмы обработки данных» для студентов очной формы обучения, размещенные на сайте СКФ МТУСИ в разделе «Методические материалы».

2.4 Методические указания для обучающихся по подготовке к контрольной работе

В ходе выполнения контрольной работы студент решает следующие задачи:

1. Реализация различных алгоритмов сортировки.
2. Реализация базовых операций над линейным списком на основе массива и обработка массива с заданными условиями.
3. Решение задач, требующих использования стека или очереди.
4. Реализация бинарного дерева поиска.

Порядок выполнения контрольной приведен в методических материалах по её выполнению. Критерии оценки контрольной работы приведены в оценочных материалах дисциплины.

3. Методические рекомендации для обучающихся по самостоятельной работе

Самостоятельная работа студентов является составной частью учебной работы и имеет целью закрепление и углубление полученных знаний и навыков, поиск и приобретение новых знаний, в том числе с использованием автоматизированных обучающих курсов (систем), а также выполнение учебных заданий, подготовку к предстоящим занятиям, зачетам и экзаменам.

Самостоятельная работа во внеаудиторное время включает в себя:

- повторение лекционного материала;
- подготовки к лабораторным работам (практическим занятиям);
- изучения учебной и научной литературы;
- решения задач, выданных на практических занятиях;
- подготовки к тестированию;
- выделение наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями кафедры на их еженедельных консультациях.
- проведение самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов, написания рефератов по отдельным вопросам изучаемой дисциплины.

Перед выполнением внеаудиторной самостоятельной работы преподаватель проводит инструктаж (консультацию) с определением цели задания, его содержания, сроков выполнения, основных требований к результатам работы, критериев оценки, форм контроля и перечня источников и литературы.

Обычно постановку задачи обучаемым на проведение самостоятельной работы преподаватель осуществляет на одном из занятий, предшествующем данному.

Методику самостоятельной работы все обучаемые выбирают индивидуально.