

**УТВЕРЖДЕНА**

Решением Ученого совета  
АНО ВО «Центральный университет»  
«07» марта 2024 г.  
Протокол №1

**Рабочая программа дисциплины (модуля)  
«Алгоритмы и структуры данных. Часть 2»**

Направление подготовки: 02.03.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Искусственный интеллект

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Срок освоения программы: 4 года

Год набора: 2024

**Москва  
2024**

## Содержание

<b>1. Краткая характеристика дисциплины (модуля)</b> .....	3
<b>2. Перечень планируемых результатов обучения</b> .....	4
<b>3. Тематический план</b> .....	4
<b>4. Содержание дисциплины (модуля)</b> .....	5
<b>5. Учебно-методическое обеспечение</b> .....	6
<b>6. Материально-техническое обеспечение</b> .....	6
<b>7. Методические и оценочные материалы</b> .....	8

## 1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Алгоритмы и структуры данных. Часть 2» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 02.03.01 Математика и компьютерные науки, профиль Искусственный интеллект, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 807 от 23.08.2017 года.

Изучение дисциплины (модуля) «Алгоритмы и структуры данных. Часть 2» формирует у студентов умение разрабатывать и создавать оптимизированные решения, что критически важно для производительности программного обеспечения. Кроме того, эти знания являются основой для дальнейшего изучения более сложных тем в области компьютерных наук и программирования.

### **Место дисциплины (модуля) в структуре образовательной программы**

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки бакалавриата по направлению 02.03.01 Математика и компьютерные науки, профиль Искусственный интеллект и входит в обязательную часть Блока 1, как дисциплина (модуль) по выбору.

Дисциплина (модуль) изучается на 2 курсе в 4 семестре.

Дисциплина (модуль) «Алгоритмы и структуры данных. Часть 2» доступна после успешного освоения дисциплины (модуля) «Алгоритмы и структуры данных. Часть 1».

**Цель изучения дисциплины (модуля):** заключается в формировании у студентов навыков проектирования и анализа эффективных алгоритмов и структур данных для решения различных задач программирования.

### **Задачи изучения дисциплины (модуля):**

— формирование знаний про асимптотический анализ алгоритмов, классические алгоритмы поиска и сортировки, основные структуры данных, основные алгоритмы на графах, основные динамические алгоритмы;

— формирование умений применять асимптотический анализ для оценки алгоритмической эффективности, реализовывать эффективные алгоритмы в различных контекстах;

— формирование умений разрабатывать и внедрять структуры данных для оптимального хранения и обработки информации, реализовывать алгоритмы графов для нахождения наименьших путей, циклов и потоков, разрабатывать решения задач с помощью динамического программирования;

— формирование навыка анализировать сложность алгоритмов и выбирать наиболее подходящие решения;

— формирование навыка развивать алгоритмическое мышление для решения комплексных задач, оптимизировать существующие алгоритмы для повышения их производительности;

— формирование умения применять алгоритмы для решения прикладных задач, эффективное использование алгоритмов поиска и сортировки в различных компьютерных приложениях, разрабатывать и применять алгоритмы графов для решения реальных задач.

## 2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) при проведении учебных занятий в форме контактной работы обучающихся с педагогическими работниками Университета и в форме самостоятельной работы обучающихся:

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
ОПК-4.	Способен находить, анализировать, реализовывать программно и использовать на практике математические алгоритмы, в том числе с применением современных вычислительных систем	ОПК-4.1.	Знает базовые основы современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
		ОПК-4.2.	Умеет использовать этот математический аппарат в профессиональной деятельности
		ОПК-4.3.	Имеет практический опыт применения современного математического аппарата, связанного с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности
ОПК-6.	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК-6.1.	Знает алгоритмы разработки, компьютерные программы, а также алгоритмы вычислительной математики в области искусственного интеллекта
		ОПК-6.2.	Умеет разрабатывать математические программные продукты и комплексы с использованием современных технологий программирования в области искусственного интеллекта
		ОПК-6.3.	Имеет практический опыт разработки интеллектуальных информационных систем для визуализации результатов исследований в области искусственного интеллекта
ПК-3.	Способен применять методы математического и алгоритмического моделирования для решения как теоретических, так и практических задач в рамках профессиональной деятельности	ПК-3.1.	Знает основные методы математического и алгоритмического моделирования, а также их применение для решения теоретических и прикладных задач
		ПК-3.2.	Умеет применять методы математического и алгоритмического моделирования для анализа и решения различных задач в области математики и компьютерных наук
		ПК-3.3.	Имеет опыт использования методов математического и алгоритмического моделирования при решении теоретических и прикладных задач в профессиональной деятельности

### 3. Тематический план

№п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		<i>Очная форма</i>				
		Контактная работа		Контроль	Самостоятельная работа	
Лекции	Семинары					
1	Графы	6	6		23	Домашнее задание Контрольная работа
2	Деревья поиска	6	8		23	Домашнее задание Контрольная работа
3	Хеширование	6	8		23	Домашнее задание Подготовка к семинару
4	Строковые алгоритмы	6	8		23	Коллоквиум
5	Динамическое программирование	8	8		24	Домашнее задание Подготовка к семинару
	<i>Экзамен</i>			4		
	<b>Итого:</b>	<b>32</b>	<b>38</b>	<b>4</b>	<b>116</b>	
	<b>Объем дисциплины (модуля) (в ак. ч.)</b>	<b>190</b>				
	<b>Объем дисциплины (модуля) (в зач. ед.)</b>	<b>5</b>				

### 4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Графы	Графы. Основные понятия и определения. Способы задания графов. Алгоритмы на графах. Поиск в глубину. Поиск в ширину. Алгоритмы на основе обходов. Кратчайшие пути. Система непересекающихся множеств. Алгоритмы построения остовного дерева.
2	Деревья поиска	Деревья. Терминология деревьев. Сбалансированные деревья. AVL-дерево. Красно-черное дерево.
3	Хеширование	Понятие хэширования. Метод цепочек. Решение коллизий.
4	Строковые алгоритмы	Z-функция. Префикс функция.
5	Динамическое программирование	Двумерная динамика. Работа со строками. Строки.

## 5. Учебно-методическое обеспечение

Университет располагает полным набором лицензионного и свободно распространяемого программного обеспечения, включая продукты отечественного производства.

Каждый студент в течение всего периода обучения получает индивидуальный неограниченный доступ к электронно-библиотечной системе и электронной информационно-образовательной среде университета. Эти системы предоставляют возможность доступа к ресурсам из любой точки, где есть подключение к сети Интернет, как на территории университета, так и за его пределами.

Студентам обеспечен удаленный доступ к современным профессиональным базам данных и информационным справочным системам.

### *Основная литература:*

1. Меджедович, Д. Алгоритмы и структуры для массивных наборов данных : практическое руководство / Д. Меджедович, Э. Тахирович ; пер. с англ. А. В. Логунова. – Москва : ДМК Пресс, 2024. - 342 с. – ISBN 978-5-93700-250-1. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2205044>.

2. Орлов, С. А. Теория и практика языков программирования : учебник для вузов / С. А. Орлов. - Санкт-Петербург : Питер, 2021. - 688 с. - (Стандарт 3-го поколения). - ISBN 978-5-4461-9740-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1857045>.

3. Бхаргава, А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих : пособие / А. Бхаргава. - Санкт-Петербург : Питер, 2021. - 288 с. - (Серия «Библиотека программиста»). - ISBN 978-5-4461-0923-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1739631>.

4. Ахмад, И. 40 алгоритмов, которые должен знать каждый программист на Python : практическое руководство / И. Ахмад. - Санкт-Петербург : Питер, 2023. - 368 с. - (Библиотека программиста). - ISBN 978-5-4461-1908-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2122957>.

### *Дополнительная литература:*

1. Палий, И. А. Линейное программирование : учебник для вузов / И. А. Палий. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 175 с. — (Высшее образование). — ISBN 978-5-534-04716-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/563472>.

2. Методы оптимизации: теория и алгоритмы : учебник для вузов / А. А. Черняк, Ж. А. Черняк, Ю. М. Метельский, С. А. Богданович. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 354 с. — (Высшее образование). — ISBN 978-5-534-04103-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/559380>.

3. Сухарев, А. Г. Численные методы оптимизации : учебник и практикум для вузов / А. Г. Сухарев, А. В. Тимохов, В. В. Федоров. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2025. — 367 с. — (Высшее образование). — ISBN 978-5-534-17381-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/562179>.

## 6. Материально-техническое обеспечение

Университет располагает материально-технической базой, соответствующей действующим противопожарным правилам и нормам и обеспечивающей проведение всех видов дисциплинарной и междисциплинарной подготовки, практической и научно-исследовательской работ обучающихся, предусмотренных учебным планом.

Помещения, которые представляют собой учебные аудитории для проведения занятий лекционного типа, занятий семинарского (практического) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы и помещения для хранения и профилактического обслуживания учебного оборудования. Помещения укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Изучение дисциплины (модуля) обеспечивается в учебных аудиториях, оснащенных:

- столами и стульями;
- компьютерной техникой;
- механическими калькуляторами;
- специализированным оборудованием, включая демонстрационное оборудование.

Помещения для самостоятельной работы обучающихся, в том числе приспособленные для использования инвалидами и лицами с ограниченными возможностями здоровья, оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

Обучающимся предоставляется доступ (в том числе удаленный) к ресурсам информационно-телекоммуникационной сети «Интернет», электронным ресурсам (в том числе электронным библиотечным системам, современным профессиональным базам данных и информационным справочным системам):

№	Наименование портала (издания, курса, документа)	Ссылка
1.	Научная электронная библиотека elibrary.ru библиотека	<a href="https://elibrary.ru/defaultx.asp">https://elibrary.ru/defaultx.asp</a>
2.	База данных для IT-специалистов	<a href="https://habr.com">https://habr.com</a>
3.	База данных ScienceDirect	<a href="https://www.sciencedirect.com">https://www.sciencedirect.com</a>
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	<a href="https://minobrnauki.gov.ru/">https://minobrnauki.gov.ru/</a>
5.	Федеральный портал «Российское образование»	<a href="https://www.edu.ru/">https://www.edu.ru/</a>
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	<a href="http://window.edu.ru/">http://window.edu.ru/</a>
7.	Единая коллекция цифровых образовательных ресурсов	<a href="http://school-collection.edu.ru/">http://school-collection.edu.ru/</a>
8.	Федеральный центр информационно - образовательных ресурсов	<a href="http://fcior.edu.ru/">http://fcior.edu.ru/</a>

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), в том числе комплект лицензионного программного обеспечения, современные профессиональные базы данных и информационные справочные системы:

Наименование ПО	Производство	Лицензионное / свободно распространяемое
<b>Операционные системы:</b>		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
<b>Браузеры:</b>		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
<b>Офисные приложения:</b>		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое

Adobe Acrobat Reader	зарубежное	свободно распространяемое
<b>Программное обеспечение для планирования и учета времени:</b>		
Toggle app	зарубежное	свободно распространяемое
<b>Системы управления проектами:</b>		
Microsoft Imagine (Project)	зарубежное	лицензионное
<b>Системы управления базами данных:</b>		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
<b>Системы резервного копирования (backup):</b>		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
<b>Справочно-правовые системы:</b>		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
<b>Средства антивирусной защиты:</b>		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
<b>Среды разработки:</b>		
Visual Studio Code	зарубежное	свободно распространяемое
Bash (Unix shell)	зарубежное	свободно распространяемое
Anaconda	зарубежное	свободно распространяемое
Robotic Operating System	зарубежное	свободно распространяемое
CopelliaSim	зарубежное	свободно распространяемое
Google Colaboratory	зарубежное	свободно распространяемое
<b>Пакеты программных средств и библиотек:</b>		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
<b>Системы управления библиографической информацией:</b>		
Zotero	зарубежное	свободно распространяемое
<b>Сервисы и службы:</b>		
Bind	зарубежное	свободно распространяемое
Docker	зарубежное	свободно распространяемое

## 7. Методические и оценочные материалы

### Методические указания для обучающихся по освоению дисциплины (модуля)

В процессе изучения дисциплины (модуля) «Алгоритмы и структуры данных. Часть 2» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекция, семинары и домашние задания, коллоквиум, контрольная работа, а также различные виды самостоятельной работы обучающихся по заданию преподавателя, направленные на развитие навыков профессиональной лексики, закрепление практических профессиональных компетенций, поощрение инициатив.

*Лекция* – систематическое, последовательное, монологическое изложение преподавателем учебного материала, как правило, теоретического характера.

В процессе лекций рекомендуется вести конспект лекций: кратко и схематично фиксировать основные идеи, выводы и обобщения лекции; выделять важные мысли, ключевые слова и термины. Необходимо отметить вопросы или материалы, которые вызывают затруднения, и попытаться найти ответы в рекомендованной литературе. Если разобраться в материале не удастся, следует сформулировать вопрос и задать его преподавателю на консультации или во время семинарского (практического) занятия.

*Участие в семинаре (аудиторная работа)* – активная работа студента на семинаре, его ответы на вопросы преподавателя и участие в дискуссии.

Для успешного участия в семинаре студентам рекомендуется заранее ознакомиться с

темой обсуждения, прочитать необходимые материалы и подготовить вопросы. Важно активно слушать и вовлекаться в дискуссию, высказывая свои мнения и аргументируя их. При ответах на вопросы преподавателя стоит быть уверенным, четким и логичным, опираясь на изученный материал. Также полезно поддерживать диалог с однокурсниками, чтобы обогатить обсуждение и расширить свои знания.

*Коллоквиум* – устные ответы на вопросы, список которых известен студенту заранее,

В процессе подготовки к коллоквиуму необходимо проанализировать учебные материалы, ознакомившись с лекциями, учебниками и дополнительными источниками, акцентируя внимание на ключевых темах. Рекомендуется создать структурированные конспекты, выделяя основные идеи, термины и формулы.

*Контрольная работа* – письменная работа с набором задач, которые нужно решить за ограниченное время.

Цель контрольной работы - получить специальные знания по одной или нескольким темам дисциплины (модуля) и продемонстрировать навыки их практического применения.

*Домашнее задание* – набор задач по темам недели.

При работе над домашними заданиями важно внимательно ознакомиться с требованиями и сроками выполнения. Рекомендуется разбивать задания на этапы, чтобы избежать перегрузки и лучше усвоить материал. Использовать различные источники информации, включая учебники и онлайн-ресурсы, для более глубокого понимания темы.

*Самостоятельная работа* – работа студентов, направленная на углубленное изучение отдельных тем и вопросов учебной дисциплины (модуля).

В процессе самостоятельной работы студенты взаимодействуют с рекомендованными материалами при минимальном участии преподавателя. Задачи студента включают работу с конспектами лекций (обработка текста), повторное изучение учебных материалов планов и тезисов ответов, изучение дополнительных тем, выполнение учебно-исследовательских заданий и другое.

### **Система оценивания результатов обучения по дисциплине (модулю)**

**Критерии получения уровня и оценивания сформированности компетенций по дисциплине (модулю) «Алгоритмы и структуры данных. Часть 2»**

Оценивание уровня учебных достижений, обучающихся по дисциплине (модулю), осуществляется в виде текущего контроля успеваемости и промежуточной аттестации.

**Промежуточная аттестация** по дисциплине (модулю) осуществляется в форме **экзамена**, при этом проводится оценка компетенций, сформированных по дисциплине.

Для оценивания текущего контроля успеваемости и промежуточной аттестации используется десятибалльная шкала оценивания, которая соотносится с традиционной пятибалльной шкалой следующим образом:

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
10	Отлично	Студент полностью владеет знаниями, изложенными в рабочей программе, и глубоко осмысляет дисциплину. Он самостоятельно и логически последовательно отвечает на все вопросы, акцентируя внимание на наиболее важном. Умеет анализировать, сравнивать, классифицировать, обобщать,
9	Отлично	
8	Отлично	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		конкретизировать и систематизировать изученный материал, выделяя ключевые моменты и устанавливая причинно-следственные связи. Четко формулирует ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты дисциплины (модуля) с практическими задачами.
7	Хорошо	Студент обладает знаниями предмета почти в полном объеме рабочей программы и самостоятельно, логически последовательно и всесторонне отвечает на все вопросы, акцентируя внимание на наиболее значимых моментах. Он умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал, выделяя его ключевые аспекты и устанавливая причинно-следственные связи. Формулирует свои ответы, уверенно интерпретирует результаты анализов и других исследований, а также решает сложные ситуационные задачи. Студент хорошо знаком с методами исследования, необходимыми для практической деятельности, и умеет связывать теоретические аспекты предмета с практическими задачами.
6	Хорошо	
5	Удовлетворительно	Студент обладает базовыми знаниями по дисциплине (модулю), но испытывает трудности при самостоятельных ответах и использует неточные формулировки. В ходе ответов он допускает ошибки, касающиеся сути вопросов. Студент способен решать только самые простые задачи и владеет лишь минимальным набором методов исследования.
4	Удовлетворительно	
3	Не сдан	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	
1	Не сдан	

Дисциплина (модуль) «Алгоритмы и структуры данных. Часть 2» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	15%	11	Набор задач по темам недели
Аудиторная работа	15%	11	Активное участие в семинарах: ответы на вопросы преподавателя и участие в дискуссии
Контрольная работа	20%	2	Письменная работа с набором задач, которые нужно решить за ограниченное время

Активность	Вес	Количество	Описание
Коллоквиум	25%	1	Устные ответы на вопросы, список которых известен студенту заранее
Экзамен	25%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине (модулю)

**Формула расчёта итоговой оценки по дисциплине (модулю) «Алгоритмы и структуры данных. Часть 2»:** « $0,15 \times \text{Домашние задания} + 0,15 \times \text{Аудиторная работа} + 0,2 \times \text{Контрольная работа} + 0,25 \times \text{Коллоквиум} + 0,25 \times \text{Экзамен}$ ».

**Текущий контроль успеваемости обучающихся по дисциплине (модулю)**

**Примерный список определений для коллоквиума**

#### **Классификация структур данных**

1. **Структуры данных** - определение и классификация.
2. **Линейные структуры данных** - описание и примеры.
3. **Нелинейные структуры данных** - описание и примеры.
4. **Статические и динамические структуры данных** - различия и примеры.
5. **Абстрактные типы данных (АТД)** - определение и примеры.
6. **Комплексные структуры данных** - определение и примеры.

#### **Стеки**

7. **Стек** - определение и основные операции (push, pop).
8. **Применение стека** - примеры использования (обратная польская запись, отмена действий).
9. **Алгоритм проверки сбалансированности скобок** - описание и реализация.
10. **Реализация стека с помощью массива** - описание и анализ.
11. **Реализация стека с помощью связного списка** - описание и анализ.

#### **Очереди**

12. **Очередь** - определение и основные операции (enqueue, dequeue).
13. **Кольцевая очередь** - определение и реализация.
14. **Приоритетная очередь** - определение и применение.
15. **Очередь с двумя концами (дека)** - определение и операции.

#### **Множества и словари**

16. **Множество** - определение и основные операции (объединение, пересечение).
17. **Словарь (ассоциативный массив)** - определение и применение.
18. **Хеш-таблица** - определение, функции хеширования и коллизии.
19. **Алгоритм поиска в множестве** - различные подходы и их сложность.
20. **Динамические множества** - реализация и применение.

#### **Односвязные линейные списки**

21. **Односвязный список** - определение и основные операции (добавление, удаление).
22. **Реверсирование односвязного списка** - алгоритм и сложность.
23. **Поиск элемента в односвязном списке** - алгоритм и сложность.
24. **Слияние двух отсортированных списков** - алгоритм и реализация.
25. **Удаление дубликатов из односвязного списка** - алгоритм и реализация.

#### **Циклические списки**

26. **Циклический односвязный список** - определение и операции.
27. **Поиск элемента в циклическом списке** - алгоритм и сложность.
28. **Слияние циклических списков** - алгоритм и реализация.

#### **Двусвязные линейные списки**

29. **Двусвязный список** - определение и основные операции.

30. **Преимущества и недостатки двусвязного списка по сравнению с односвязным** - анализ.
31. **Реверсирование двусвязного списка** - алгоритм и сложность.
32. **Удаление элемента из двусвязного списка** - алгоритм и реализация.

#### **Общие темы**

33. **Сравнение структур данных** - когда использовать каждую из них.
34. **Сложность операций** - временная и пространственная сложность для различных структур.
35. **Реализация структуры данных в различных языках программирования** - примеры.
36. **Применение структур данных в алгоритмах** - примеры использования.
37. **Алгоритмы на графах** - связь с линейными списками.
38. **Структуры данных для реализации очередей и стеков** - различные подходы.
39. **Динамическое выделение памяти** - работа с указателями.
40. **Обзор популярных библиотек для работы со структурами данных** - примеры и применение.

### **Примерные задания для контрольных работ**

#### **Контрольная работа №1**

**Задание 1.** Определите, что такое алгоритм. Приведите примеры различных типов алгоритмов и их применения в реальной жизни.

**Задание 2.** Опишите основные способы измерения времени выполнения алгоритмов. Сравните их и объясните, в каких случаях каждый из методов наиболее уместен.

**Задание 3.** Объясните, что такое асимптотическая нотация. Приведите примеры использования нотаций  $O(n)$ ,  $\Theta(n)$  и  $\Omega(n)$  и объясните, что они означают.

**Задание 4.** Примените мастер-теорему для анализа временной сложности следующего рекуррентного соотношения:  $T(n) = 2T(n/2) + n$ . Укажите, какой случай мастер-теоремы вы использовали и приведите подробное решение.

**Задание 5.** Решите следующее рекуррентное соотношение с помощью метода подстановки:  $T(n) = T(n - 1) + n$ . Найдите асимптотику  $T(n)$ .

**Задание 6.** Напишите алгоритм для линейного поиска элемента в массиве и проанализируйте его временную сложность. Объясните, как вы пришли к этому результату.

**Задание 7.** Опишите алгоритм, который находит сумму всех элементов в односвязном списке. Какова временная сложность этого алгоритма?

**Задание 8.** Приведите пример рекуррентного соотношения, которое можно решить с помощью мастер-теоремы. Проанализируйте его временную сложность и объясните, какой случай теоремы был использован.

**Задание 9.** Объясните, как можно использовать линейные алгоритмы для решения задачи о нахождении максимального и минимального элементов в массиве. Напишите алгоритм и проанализируйте его сложность.

**Задание 10.** Приведите пример задачи, которая может быть решена с помощью рекурсии, и напишите соответствующий рекурсивный алгоритм. Проанализируйте временную сложность вашего решения.

#### **Контрольная работа №2**

**Задание 1.** Опишите алгоритм Евклида для нахождения наибольшего общего делителя (НОД) двух чисел. Приведите пример работы алгоритма на конкретных числах.

**Задание 2.** Реализуйте алгоритм Решета Эратосфена для нахождения всех простых чисел до  $N$ . Объясните, как работает этот алгоритм, и приведите пример его использования.

**Задание 3.** Объясните, что такое факторизация чисел. Напишите алгоритм для нахождения простых множителей заданного числа и проанализируйте его временную сложность.

**Задание 4.** Реализуйте рекурсивный алгоритм для бинарного поиска в отсортированном массиве. Приведите пример массива и элемента для поиска, а также объясните, как работает ваш алгоритм.

**Задание 5.** Напишите алгоритм для целочисленного бинарного поиска по ответу. Приведите пример задачи, где этот метод может быть применен, и объясните шаги алгоритма.

**Задание 6.** Приведите примеры базовых сортировок (пузырьковая, вставками, выбором) и проанализируйте их временную сложность в лучшем, среднем и худшем случаях.

**Задание 7.** Опишите алгоритм сортировки слиянием. Реализуйте его и проанализируйте временную и пространственную сложность.

**Задание 8.** Реализуйте алгоритм быстрой сортировки. Объясните, как выбирается опорный элемент и как это влияет на эффективность алгоритма.

**Задание 9.** Объясните метод двух указателей и приведите пример его использования для решения задачи, например, для нахождения пары чисел в массиве, сумма которых равна заданному значению.

**Задание 10.** Опишите алгоритм "скользящее окно" и приведите пример задачи, где он может быть применен, например, для нахождения максимальной суммы подмассива фиксированной длины. Реализуйте алгоритм и проанализируйте его сложность.

### **Примерные домашние задания**

#### **Домашнее задание по теме: «Графы»**

1. Реализуйте структуру данных для представления графа с использованием списка смежности и матрицы смежности. Напишите функции для добавления и удаления вершин и ребер, а также для вывода графа на экран.

2. Реализуйте алгоритмы обхода графа в глубину (DFS) и в ширину (BFS). Напишите программу, которая выводит порядок посещения вершин для заданного графа и начальной вершины. Проанализируйте временную сложность каждого алгоритма.

3. Реализуйте алгоритм Дейкстры для поиска кратчайшего пути в взвешенном графе без отрицательных ребер. Напишите программу, которая находит кратчайший путь от заданной начальной вершины до всех остальных вершин графа.

4. Напишите программу для обнаружения циклов в ориентированном и неориентированном графах. Используйте алгоритм обхода в глубину с отслеживанием состояния вершин. Проверьте работу программы на нескольких примерах.

5. Реализуйте алгоритмы Крускала и Прима для поиска минимального остовного дерева в связном взвешенном неориентированном графе. Сравните эффективность обоих алгоритмов на различных тестовых данных.

#### **Домашнее задание по теме: «Деревья поиска»**

1. Создайте класс для бинарного дерева поиска (BST), который поддерживает операции вставки, удаления и поиска элементов. Реализуйте методы для обхода дерева (инордер, префиксный и постфиксный) и напишите тесты для проверки корректности работы вашего класса.

2. Исследуйте различные методы балансировки бинарных деревьев поиска, такие как AVL-деревья или красно-черные деревья. Реализуйте один из этих методов и сравните его с обычным бинарным деревом поиска по времени выполнения операций вставки, удаления и поиска на различных наборах данных.

3. Напишите программу, которая находит  $k$ -й по величине элемент в бинарном дереве поиска. Используйте обход дерева для решения этой задачи и проанализируйте сложность вашего алгоритма. Подготовьте отчет о времени выполнения для различных значений  $k$ .

4. Реализуйте функцию, которая проверяет, является ли заданное дерево бинарным деревом поиска. Используйте рекурсивный подход для проверки правильности расположения узлов. Напишите тесты с различными деревьями, чтобы продемонстрировать работу вашей функции.

5. Исследуйте практическое применение деревьев поиска в реальных задачах, таких как реализация словарей или индексов в базах данных. Напишите краткое исследование о том, как деревья поиска помогают в организации и быстром доступе к данным, и приведите примеры их использования в программировании.

### **Домашнее задание по теме: «Хеширование»**

1. Создайте класс для хеш-таблицы, который поддерживает операции вставки, удаления и поиска элементов. Используйте метод цепочечного хеширования для обработки коллизий. Напишите тесты для проверки корректности работы вашего класса.

2. Исследуйте производительность хеш-таблицы с различными хеш-функциями. Реализуйте несколько различных хеш-функций и сравните их эффективность по времени выполнения операций (вставка, поиск, удаление) на различных наборах данных. Подготовьте отчет с графиками и выводами.

3. Реализуйте хеш-таблицу с открытой адресацией для разрешения коллизий. Используйте линейное или квадратичное пробирование. Проведите эксперименты с разными размерами таблицы и количеством вставляемых элементов, чтобы оценить, как это влияет на производительность.

4. Напишите программу, которая использует хеширование для хранения и поиска строк. Реализуйте хеш-функцию, которая учитывает длину строки и её символы. Проведите тесты, чтобы продемонстрировать, как ваша реализация справляется с коллизиями и насколько эффективно выполняет поиск.

5. Исследуйте практическое применение хеширования в реальных задачах, таких как хранение паролей или работа с кэшами. Напишите краткое исследование о том, как хеширование помогает обеспечивать безопасность и эффективность, и приведите примеры его использования в программировании.

### **Примерны вопросы для семинаров**

1. Что такое алгоритм? Опишите основные характеристики, которые должен иметь алгоритм.

2. Объясните разницу между временной и пространственной сложностью алгоритма. Приведите примеры.

3. Опишите различия между массивами и связными списками. В каких ситуациях предпочтительнее использовать каждую из этих структур данных?

4. Что такое бинарное дерево? Опишите основные операции, которые можно выполнять с бинарным деревом (например, вставка, удаление, обход).

5. Объясните, что такое ориентированный и неориентированный граф. Приведите примеры применения графов в реальных задачах.

6. Опишите алгоритм Дейкстры для поиска кратчайшего пути в графе. Какие ограничения у этого алгоритма?

7. Сравните алгоритмы сортировки "пузырьком" и "быстрой сортировки". Какова их временная сложность в лучшем и худшем случае?

8. Что такое хэш-таблица? Каковы основные операции с хэш-таблицами и как они обеспечивают эффективный доступ к данным?

9. Опишите различия между стеком и очередью. Приведите примеры, где каждая из этих структур данных может быть использована.

10. Опишите алгоритм Крускала для нахождения минимального остовного дерева. Каковы его основные шаги и сложность?
11. Что такое алгоритм? Перечислите основные свойства, которые должен иметь алгоритм.
12. Объясните разницу между временной и пространственной сложностью. Как они измеряются?
13. В чем разница между статическими и динамическими структурами данных? Приведите примеры.
14. Каковы основные преимущества и недостатки массивов по сравнению со связными списками?
15. Что такое бинарное дерево поиска? Каковы его основные свойства?
16. Определите, что такое граф. Какие способы представления графов вы знаете?
17. Объясните, в чем разница между обходом в глубину (DFS) и обходом в ширину (BFS). В каких случаях лучше использовать каждый из методов?
18. Каковы основные принципы работы алгоритма "быстрой сортировки"? Какова его временная сложность в лучшем и худшем случае?
19. Что такое хэш-таблица и как она работает? Каковы основные операции с хэш-таблицами?
20. Опишите основные операции, которые можно выполнять со стеком и очередью. Приведите примеры их применения.
21. Как работает алгоритм Дейкстры? В каких случаях он может не сработать?
22. Опишите алгоритм Крускала для нахождения минимального остовного дерева. Какие структуры данных могут быть использованы для его реализации?
23. Опишите алгоритм сортировки слиянием. Какова его временная сложность?
24. Что такое динамическое программирование? Приведите пример задачи, которую можно решить с помощью этого метода.
25. Каковы основные различия между линейным и бинарным поиском? В каких случаях каждый из них применяется?
26. Как можно оптимизировать поиск в массиве, если он отсортирован? Опишите алгоритм бинарного поиска.
27. Объясните, как решается задача о рюкзаке с помощью динамического программирования.
28. Что такое сбалансированное дерево? Приведите примеры сбалансированных деревьев и их преимущества.
29. Как можно использовать алгоритм Флойда-Уоршелла для нахождения кратчайших путей во взвешенном графе?
30. Что такое trie (префиксное дерево)? Каковы его основные применения?

#### Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Структура данных представляет собой: а) набор правил и ограничений, определяющих связи между отдельными элементами и группами данных б) набор правил и ограничений, определяющих связи между отдельными элементами данных в) набор правил и ограничений, определяющих связи между отдельными группами данных г) некоторую иерархию данных	б	ОПК-6
2.	Линейный список, в котором доступен только последний элемент, называется: а) стеком б) очередью в) деком г) массивом	а	ОПК-4

	д) кольцом		
3.	Структура данных, работа с элементами которой организована по принципу FIFO (первый пришел - первый ушел) это: а) Стек б) Дек в) Очередь г) Список	в	ОПК-6
4.	В чём особенности очереди? а) открыта с обеих сторон; б) открыта с одной стороны на вставку и удаление; в) доступен любой элемент.	а	ОПК-4
5.	Каково правило выборки элемента из стека? а) первый элемент; б) последний элемент; (*последний добавленный) в) любой элемент.	б	ПК-3
6.	Какой алгоритм используется для нахождения наибольшего общего делителя (НОД) двух чисел?	Алгоритм Евклида	ОПК-4
7.	Как называется метод, который позволяет находить все простые числа до заданного числа $n$ ?	Решето Эратосфена	ОПК-4
8.	Какой алгоритм сортировки имеет среднюю временную сложность $O(n \log n)$ и использует метод "разделяй и властвуй"?	Быстрая сортировка	ОПК-6
9.	Какой метод используется для поиска элемента в отсортированном массиве с временной сложностью $O(\log n)$ ?	Бинарный поиск	ПК-3
10.	Как называется алгоритм, который разбивает массив на две части и объединяет их в отсортированном порядке?	Сортировка слиянием	ПК-3
11.	Какой метод сортировки использует структуру данных "куча" для упорядочивания элементов?	Пирамидальная сортировка	ПК-3
12.	Какой метод используется для нахождения пар, которые удовлетворяют определённому условию, перемещая два указателя по массиву?	Метод двух указателей	ПК-3
13.	Как называется алгоритм, который позволяет находить подмассивы с максимальной суммой?	Скользящее окно	ПК-3
14.	Какой алгоритм используется для разложения числа на множители, особенно для больших чисел?	Факторизация	ОПК-6