

**УТВЕРЖДЕНА**

Решением Ученого совета  
АНО ВО «Центральный университет»  
«24» июня 2025 г.  
Протокол №2

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

**Направление подготовки:** 38.03.05 Бизнес-информатика

**Направленность (профиль) подготовки:** Бизнес-аналитика

**Квалификация (степень) выпускника:** бакалавр

**Форма обучения:** очная

**Срок освоения программы:** 4 года

**Год набора:** 2025

**Москва  
2025**

## Содержание

<b>1. Краткая характеристика дисциплины (модуля)</b> .....	<b>3</b>
<b>2. Перечень планируемых результатов обучения</b> .....	<b>5</b>
<b>3. Тематический план</b> .....	<b>7</b>
<b>4. Содержание дисциплины (модуля)</b> .....	<b>8</b>
<b>5. Учебно-методическое обеспечение</b> .....	<b>9</b>
<b>6. Материально-техническое обеспечение</b> .....	<b>9</b>
<b>7. Методические и оценочные материалы</b> .....	<b>11</b>

## 1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Алгоритмы и структуры данных 2» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по специальности 38.03.05 Бизнес-информатика, профиль Бизнес-аналитика, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 838 от 29.07.2020 года.

Изучение дисциплины (модуля) «Алгоритмы и структуры данных 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 38.03.05 Бизнес-информатика, профиль Бизнес-аналитика и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений как дисциплина по выбору.

Дисциплина (модуль) изучается на 2 курсе в 4 семестре, доступна для изучения после успешного освоения любой из дисциплины (модуля): «Алгоритмы и структуры данных 1».

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

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

- формирование знаний основных понятий алгоритмов, графов;
- формирование умений анализировать, реализовывать эффективные алгоритмы в различных контекстах, разрабатывать и внедрять структуры данных для оптимального хранения и обработки информации;
- формирование умения применять алгоритмы для решения прикладных задач.

### **В результате освоения дисциплины (модуля) студент должен:**

#### ***знать:***

- асимптотический анализ алгоритмов
- классические алгоритмы поиска и сортировки
- основные структуры данных
- основные алгоритмы на графах
- основные динамические алгоритмы

#### ***уметь:***

- применять асимптотический анализ для оценки алгоритмической эффективности
- реализовывать эффективные алгоритмы в различных контекстах.
- 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.

#### ***владеть:***

- навыком анализа сложности алгоритмов и выбора наиболее подходящих решений;
- навыком алгоритмического мышления для решения комплексных задач;
- навыком оптимизации существующих алгоритмов для повышения их производительности;
- навыком применения алгоритмов для решения прикладных задач;

- навыком эффективного использования алгоритмов поиска и сортировки в различных компьютерных приложениях;
- навыком разработки и применения алгоритмов графов для решения реальных задач.

## 2. Перечень планируемых результатов обучения

Компетенции, формируемые в результате освоения дисциплины (модуля) (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)
УК-1.	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1.	Знает методы поиска и анализа информации в области аналитики, основные принципы критической оценки источников информации и их релевантности.
		УК-1.2.	Умеет критически оценивать источники информации и синтезировать данные из различных источников для решения задач, применять системный подход к анализу и решению комплексных проблем
		УК-1.3.	Имеет практический опыт работы с современными инструментами и технологиями для обработки информации, формулировании и структурировании задач на основе полученной информации
ОПК-2.	Способен проводить исследование и анализ рынка информационных систем и информационно-коммуникационных технологий, выбирать рациональные решения для управления бизнесом	ОПК-2.1.	Знает основные тенденции и характеристики рынка информационных систем и информационно-коммуникационных технологий
		ОПК-2.2.	Умеет проводить исследование и анализ рыночной информации для оценки потребностей бизнеса и выбора оптимальных решений
		ОПК-2.3.	Имеет практический опыт в разработке и внедрении стратегий управления бизнесом на основе анализа рынка информационных технологий
ПК-2.	Способен использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования для решения задач профессиональной деятельности	ПК-2.1.	Знает основные математические методы и инструментальные средства, применяемые для обработки и анализа информации
		ПК-2.2.	Умеет эффективно использовать математический аппарат для систематизации данных и решения профессиональных задач
		ПК-2.3.	Имеет практический опыт работы с инструментами анализа информации в рамках исследовательских проектов
ПК-3.	Способен готовить научно-	ПК-3.1.	Знает требования и стандарты

	технические отчеты, презентации, научные публикации по результатам выполненных исследований		оформления научно-технических отчетов, презентаций и публикаций
		ПК-3.2.	Умеет структурировать и представлять результаты исследований в ясной и доступной форме
		ПК-3.3.	Имеет практический опыт подготовки и публикации научных материалов, отражающих результаты выполненных исследований
ПК-8	Способен под руководством специалиста более высокой категории осуществлять планирование и организацию проектной деятельности на основе стандартов управления проектами	ПК-8.1.	Знает принципы и стандарты управления проектами
		ПК-8.2.	Умеет разрабатывать планы и организовывать проектную деятельность в соответствии с установленными стандартами
		ПК-8.3.	Имеет практический опыт участия в проектной работе, включая планирование и координацию задач

### 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	Консультации					
1	Графы	6	6	1		24	Домашнее задание Подготовка к семинару
2	Деревья поиска	6	6	1		24	Домашнее задание Контрольная работа
3	Хеширование	6	6	2		24	Домашнее задание Подготовка к семинару задание
4	Строковые алгоритмы	6	6	1		24	Коллоквиум
5	Динамическое программирование	6	6	1		24	Домашнее задание Подготовка к семинару
	<i>Экзамен</i>				4		
	<i>Итого:</i>	<i>30</i>	<i>30</i>	<i>6</i>	<i>4</i>	<i>120</i>	
	<i>Объем дисциплины (модуля) (в ак. ч.)</i>	<i>190</i>					
	<i>Объем дисциплины (модуля) (в зач. ед.)</i>	<i>5</i>					

#### 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 острого дерева
2	Деревья поиска	Деревья. Терминология деревьев. Сбалансированные деревья. AVL-дерево. Красно-черное дерево
3	Хеширование	Понятие хэширование. 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. 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>.

2. 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>.

### **Дополнительная литература:**

3. 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>.

## 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 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 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:

<b>Десятибалльная оценка</b>	<b>Пятибалльная оценка</b>	<b>Общая характеристика результата обучения по дисциплине (модулю)</b>
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, 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.
9	Отлично	
8	Отлично	
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
6	Хорошо	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		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.
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 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
Коллоквиум	30%	1	Устные ответы на вопросы, список которых известен студенту заранее
Экзамен	20%	1	Письменная или устная работа над заданием, направленным на проверку полученных знаний и навыков по дисциплине

**Формула расчёта итоговой оценки по дисциплине (модулю) «Алгоритмы и структуры данных 2»:** « $0,15 \times \text{Домашние задания} + 0,15 \times \text{Аудиторная работа} + 0,2 \times \text{Контрольная работа} + 0,3 \times \text{Коллоквиум} + 0,2 \times \text{Экзамен}$ ».

**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 связанного 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 вопросы для подготовки к семинару

### Понятие алгоритма. Оценка трудоемкости алгоритма. Асимптотика

- Что такое алгоритм? Какие основные свойства алгоритма?
- Какие способы измерения времени выполнения алгоритма существуют?

- Что такое временная сложность алгоритма?
- Что такое пространственная сложность алгоритма?
- Объясните понятия «худший», «лучший» и «средний» случаи в оценке алгоритмов.
- Что такое асимптотическая нотация  $O$ ,  $\Omega$ ,  $\Theta$ ? Приведите определения.
- Как интерпретировать запись  $T(n) = O(n^2)$ ?
- В чем разница между  $O(n)$ ,  $o(n)$ ,  $\Omega(n)$ ,  $\omega(n)$ ,  $\Theta(n)$ ?
- Как использовать асимптотические оценки для сравнения алгоритмов?
- Приведите пример алгоритма и оцените его временную сложность.

### **Базовые алгоритмы. Мастер-теорема. Рекуррентные соотношения**

- Что такое рекуррентное соотношение?
- Как формулируется мастер-теорема?
- Какие случаи мастер-теоремы существуют?
- Приведите пример рекуррентного соотношения и решите его с помощью мастер-теоремы.
- Как определить асимптотическую сложность алгоритма на основе рекуррентного соотношения?
- Что такое метод подстановки для решения рекуррентных соотношений?
- Какие виды рекуррентных соотношений существуют (линейные, неравномерные и т.д.)?
- Как решать рекуррентные соотношения с помощью дерева рекурсии?
- В каких алгоритмах часто встречаются рекуррентные соотношения?
- Приведите пример алгоритма, для которого мастер-теорема не применима.

### **Алгоритмы поиска. Линейные алгоритмы**

- Что такое линейный поиск?
- Какова временная сложность линейного поиска?
- В каких случаях линейный поиск предпочтительнее бинарного?
- Как реализовать алгоритм линейного поиска?
- Что такое поиск с барьером?
- Какие преимущества и недостатки у линейного поиска?
- Как искать минимальный или максимальный элемент в массиве?
- Что такое поиск по критерию?
- Как изменить линейный поиск для работы с неупорядоченными структурами?
- Приведите пример задачи, где линейный поиск является оптимальным решением.

### **Алгоритмы сортировки. Теория чисел. Алгоритм Евклида**

- Как работает алгоритм Евклида для нахождения НОД?
- Что такое решето Эратосфена и как оно используется для поиска простых чисел?
- Какие методы факторизации чисел вы знаете?
- Как реализуется бинарный поиск? В чем его сложность?
- В чем отличие целочисленного бинарного поиска от вещественного бинарного поиска по ответу?
- Как работает сортировка слиянием? Какова её временная сложность?
- Что такое быстрая сортировка? Как она работает?
- Опишите принцип пирамидальной (heap) сортировки.
- Что такое метод двух указателей и где он применяется?
- Объясните алгоритмы сортировки событий и использование скользящего окна.

## Структуры данных

- Как классифицируются структуры данных?
- Что такое стек? Какие операции поддерживает?
- Что такое очередь и очередь с приоритетом?
- Что такое дека? Как она отличается от очереди?
- Какие операции поддерживают множества и словари?
- Как устроен односвязный линейный список?
- В чем отличие циклического списка от обычного?
- Что такое двусвязный список? Какие преимущества он даёт?
- Как реализовать основные операции (вставка, удаление) в списках?
- В каких задачах наиболее эффективны списочные структуры?

## Динамическое программирование (одномерная динамика)

- Что такое динамическое программирование?
- В чем отличие динамического программирования от жадных алгоритмов?
- Как решается задача НВП (наибольшей возрастающей подпоследовательности)?
- Как решается задача НОП (наибольшей общей подпоследовательности)?
- Что такое оптимальная подструктура?
- Как строится таблица для динамического программирования?
- Какова временная и пространственная сложность алгоритмов динамического программирования?
- Какие основные этапы решения задачи динамическим программированием?
- Приведите пример задачи, решаемой одномерной динамикой.
- Как оптимизировать по памяти решения динамического программирования?

## Графы

- Что такое граф? Какие виды графов существуют?
- Как задаются графы (матрица смежности, список смежности и др.)?
- Как работает алгоритм поиска в глубину (DFS)?
- Как работает алгоритм поиска в ширину (BFS)?
- Какие задачи решаются с помощью обходов графа?
- Что такое кратчайший путь? Какие алгоритмы его нахождения вы знаете?
- Что такое система непересекающихся множеств? Для чего она используется?
- Как построить остовное дерево? Какие алгоритмы для этого существуют?
- В чем отличие минимального остовного дерева от минимального остовного леса?
- Приведите пример применения алгоритмов на графах в реальных задачах.

## Деревья поиска

- Что такое дерево? Какие виды деревьев существуют?
- Что такое бинарное дерево поиска (BST)?
- Какие операции поддерживает BST?
- Что такое сбалансированное дерево? Зачем нужно балансировать дерево?
- Как устроено AVL-дерево? Какие операции поддерживаются?
- Как происходит балансировка в AVL-дереве?
- Что такое красно-черное дерево? Основные свойства?
- Как происходит вставка и удаление в красно-черном дереве?
- Сравните AVL-дерево и красно-черное дерево по производительности и сложности.
- Где применяются сбалансированные деревья в практике?

## Хеширование

- Что такое хеширование? Для чего оно используется?
- Что такое хеш-функция? Какие требования к ней предъявляются?
- Какие способы разрешения коллизий вы знаете?
- Как работает метод цепочек?
- Какие преимущества и недостатки метода цепочек?
- Что такое открытая адресация? Какие её виды существуют?
- Как выбрать размер хеш-таблицы?
- Что такое хеш-таблица с динамическим расширением?
- Как влияет качество хеш-функции на производительность?
- Приведите пример практического применения хеширования.

## Строковые алгоритмы

- Что такое Z-функция и как она вычисляется?
- Как используется Z-функция для поиска подстроки?
- Что такое префикс-функция (функция  $\pi$ )?
- Как вычисляется префикс-функция?
- Как префикс-функция используется в алгоритме Кнута-Морриса-Пратта?
- В чем преимущество КМП по сравнению с наивным поиском?
- Какие задачи решаются с помощью Z-функции и префикс-функции?
- Как реализовать алгоритм поиска всех вхождений подстроки в строку?
- Что такое суффиксный массив и как он связан с префикс-функцией?
- Приведите пример задачи на обработку строк с использованием этих алгоритмов.

## Динамическое программирование (двумерная динамика, работа со строками)

- В чем особенности двумерного динамического программирования?
- Как решается задача поиска наибольшей общей подпоследовательности (НОП)?
- Как решается задача редактирования строк (редакционное расстояние)?
- Что такое таблица переходов в динамическом программировании по строкам?
- Как реализовать алгоритм для задачи о рюкзаке с двумерной динамикой?
- Как оптимизировать память при решении двумерных задач?
- Какие типичные ошибки допускаются при реализации двумерной динамики?
- Приведите пример задачи на динамическое программирование со строками.
- Как использовать динамическое программирование для подсчёта количества способов?
- В чем преимущества и ограничения динамического программирования при работе со строками?

## Примерные домашние задания

### Домашнее задание по теме: «Алгоритмы и их свойства»

1. Определите, что такое алгоритм, и приведите три примера различных алгоритмов из повседневной жизни.
2. Назовите и объясните три основных свойства алгоритмов.
3. Сравните детерминированные и недетерминированные алгоритмы. Приведите примеры каждого типа.
4. Опишите, как можно измерить время выполнения алгоритма. Какие инструменты могут быть использованы для этой цели?
5. Объясните, что такое временная сложность алгоритма и как она может повлиять на выбор алгоритма для решения задачи.

### Домашнее задание по теме: «Асимптотические нотации и рекуррентные отношения»

1. Что такое асимптотические нотации? Приведите примеры "О большого", "Ω" и "Θ".
2. Объясните, как интерпретировать нотацию "О большого" на примере алгоритма сортировки.
3. Что такое мастер-теорема и в каких случаях она используется? Приведите пример.
4. Решите следующее рекуррентное соотношение:  $T(n)=2T(n/2)+n$ . Используйте мастер-теорему для нахождения временной сложности.
5. Приведите пример рекурсивного алгоритма и объясните его временную сложность.

### Домашнее задание по теме: «Алгоритмы сортировки и бинарный поиск»

1. Опишите алгоритм сортировки слиянием. Какова его временная сложность в лучшем и худшем случае?
2. Объясните принцип работы быстрой сортировки. Как она отличается от сортировки слиянием?
3. Как работает бинарный поиск? В каких условиях он применяется?
4. Реализуйте алгоритм бинарного поиска на языке программирования по вашему выбору и протестируйте его на различных входных данных.
5. Объясните, как осуществляется целочисленный и вещественный бинарный поиск по ответу. Приведите примеры.

### Домашнее задание по теме: «Теория чисел и структуры данных»

1. Что такое алгоритм Евклида? Объясните, как он используется для нахождения НОД (наибольшего общего делителя) двух чисел.
2. Опишите, как работает решето Эратосфена для нахождения простых чисел. Какова его временная сложность?
3. Приведите примеры методов факторизации чисел и обсудите их эффективность.
4. Что такое метод двух указателей? Приведите пример его применения на практике.
5. Объясните, какие основные структуры данных вы знаете, и приведите примеры их использования в алгоритмах.

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

№ п/п	Задание	Ответ	Компетенция
1	Укажите метод поиска информации для анализа асимптотической сложности алгоритмов.	Литературный обзор (варианты: literature review, обзор литературы)	УК-1
2	Укажите способ синтеза данных из источников для оценки трудоемкости поиска.	Кросс-валидация (варианты: cross-validation, перекрёстная проверка)	УК-1
3	Укажите инструмент для обработки текстов в анализе рекуррентных соотношений.	Wolfram Mathematica (варианты: wolfram mathematica, математика вольфрам)	УК-1
4	Укажите технологию для структурирования задач в линейных алгоритмах поиска.	Elasticsearch (варианты: elasticsearch, эластик серч)	УК-1
5	Укажите тенденцию рынка информационных систем в оптимизации сортировок.	Автоматизация (варианты: automation, автоматизированные алгоритмы)	ОПК-2

6	Укажите метод анализа рыночных потребностей для оценки эффективности структур данных.	SWOT (варианты: swot-анализ, swot)	ОПК-2
7	Укажите стратегию внедрения на основе анализа рынка ИКТ в деревьях поиска.	Agile (варианты: agile, гибкая методология)	ОПК-2
8	Укажите инструмент для внедрения стратегий управления бизнесом в графовых алгоритмах.	Kanban (варианты: kanban, канбан)	ОПК-2
9	Укажите математический метод для анализа данных в динамическом программировании.	Рекуррентные соотношения (варианты: recurrence relations, рекуррентные)	ПК-2
10	Укажите инструмент для систематизации данных в хешировании.	Python (варианты: python, питон)	ПК-2
11	Укажите опыт применения математического аппарата в проектах по строковым алгоритмам.	Z-функция (варианты: z-function, z-функция)	ПК-2
12	Укажите инструмент анализа информации в исследованиях графов.	NetworkX (варианты: networkx, нетворккс)	ПК-2
13	Укажите стандарт оформления отчетов по результатам экспериментов с сортировками.	APA (варианты: apa, апа)	ПК-3
14	Укажите инструмент для визуализации данных в презентациях по структурам данных.	Tableau (варианты: tableau, табло)	ПК-3
15	Укажите платформу для публикации статей о динамическом программировании.	ACM Digital Library (варианты: acm digital library, цифровая библиотека acm)	ПК-3
16	Укажите формат файла для хранения научных отчетов в алгоритмах.	LaTeX (варианты: latex, латех)	ПК-3
17	Укажите методологию управления проектами в разработке систем на графах.	Waterfall (варианты: waterfall, водопад)	ПК-8
18	Укажите руководство для стандартизации процессов в проектах по деревьям поиска.	ISO 21500 (варианты: iso 21500, изо 21500)	ПК-8
19	Укажите инструмент для планирования этапов в проектах по хешированию.	Asana (варианты: asana, асана)	ПК-8
20	Укажите метод координации команды в разработке алгоритмов на строках.	Slack (варианты: slack, слэк)	ПК-8