
УТВЕРЖДЕНА

Решением Ученого совета
АНО ВО «Центральный университет»
«25» мая 2026 г.
Протокол № 2

**Рабочая программа дисциплины (модуля)
«Алгоритмы»**

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Управление командами
разработки

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2026

**Москва
2026**

Содержание

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

1. Краткая характеристика дисциплины (модуля)

Рабочая программа дисциплины (модуля) «Алгоритмы» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования – магистратура по направлению 02.04.01 Математика и компьютерные науки, профиль Управление командами разработки, утвержденный приказом Министерства науки и высшего образования Российской Федерации № 810 от 23.08.2017 года.

Изучение дисциплины (модуля) «Алгоритмы» позволяет заложить фундаментальную основу для эффективной разработки программного обеспечения, анализа данных и решения задач в области искусственного интеллекта, биоинформатики, системного программирования и веб-разработки. Дисциплина (модуль) формирует алгоритмическое мышление, необходимое для создания производительных, масштабируемых и корректных решений в условиях реальных вычислительных ограничений.

Место дисциплины (модуля) в структуре образовательной программы

Настоящая дисциплина (модуль) включена в учебный план по программе подготовки магистратуры по направлению 02.04.01 Математика и компьютерные науки, профиль Управление командами разработки и входит в вариативную часть Блока 1, формируемую участниками образовательных отношений как дисциплина по выбору.

Дисциплина (модуль) доступна к изучению на 1 и 2 курсе во 2 и 3 семестре на выбор.

Цель изучения дисциплины (модуля): формирование у обучающихся глубокого понимания принципов проектирования, анализа и оптимизации алгоритмов, а также развитие навыков эффективного решения сложных вычислительных задач с использованием фундаментальных методов и структур данных.

Задачи изучения дисциплины (модуля):

- изучить основные принципы анализа алгоритмов: временная и пространственная сложность, оценка в лучшем, среднем и худшем случаях, амортизационный анализ;
- освоить базовые и продвинутые алгоритмы поиска, сортировки, обработки строк, работы с графами и деревьями;
- научиться применять парадигмы алгоритмического программирования: рекурсию, динамическое программирование, жадные методы;
- овладеть методами хеширования, построения и использования хеш-таблиц с разрешением коллизий;
- развить навыки реализации и анализа графовых алгоритмов: обходы, кратчайшие пути, топологическая сортировка, компоненты связности;
- научиться выбирать и адаптировать оптимальные структуры данных и алгоритмы для решения прикладных задач в условиях ограничений по времени и памяти.

В результате освоения дисциплины (модуля) обучающийся должен:

знать:

- понятие и принципы анализа алгоритмов: временная и пространственная сложность, основные типы сложности ($O(1)$, $O(n)$, $O(\log n)$, $O(n^2)$), их примеры и подходы к оценке (лучший, худший случай, амортизационный анализ);
- базовые алгоритмы поиска и сортировки: бинарный поиск, сортировки вставками, слиянием и кучей, их особенности, требования и сложность;
- рекурсия: её компоненты, ограничения, дерево вызовов, применение в задачах;
- принципы и преимущества динамического программирования: разбиение задач на подзадачи, мемоизация, итеративный подход, примеры на задачах НОП и НВП;
- основы жадных алгоритмов: локальная оптимизация, их применение и ограничения;
- алгоритмы строкового поиска: расстояние Левенштейна, префикс-функция, алгоритм КМП, Рабина-Карпа, их применения и особенности;
- основы хеширования: хеш-функции, хеш-таблицы, методы разрешения коллизий, применение хеширования в задачах;

- теория графов: понятие вершин, рёбер, взвешенных графов, представление графов (матрица и список смежности), основы BFS и DFS, слабая и сильная связность, топологическая сортировка, компоненты сильной связности и конденсация графа;
- алгоритмы поиска кратчайшего пути: Дейкстры и Флойда-Уоршелла, их применение, особенности и оптимизация;
- основы работы с деревьями: структура бинарных деревьев поиска (BST), свойства, операции вставки, удаления, поиска, обходов (pre-order, in-order, post-order), сложность операций.

уметь:

- оценивать временную и пространственную сложность алгоритмов, анализировать их эффективность;
- реализовывать и модифицировать алгоритмы поиска и сортировки (бинарный поиск, сортировки вставками, слиянием, кучей);
- создавать рекурсивные алгоритмы, выявлять избыточные вычисления и оптимизировать их;
- реализовывать алгоритмы динамического программирования (мемоизация, итеративный подход) для задач, связанных с оптимизацией, например, НОП и НВП;
- решать задачи оптимизации с использованием жадных алгоритмов, анализировать их эффективность;
- разрабатывать алгоритмы строкового поиска (КМП, Рабина-Карпа), восстанавливать пути преобразований (например, расстояние Левенштейна);
- использовать хеш-функции для создания хеш-таблиц и решения задач на основе хеширования;
- реализовывать алгоритмы обхода графов (BFS, DFS), находить компоненты связности, топологическую сортировку, компоненты сильной связности (алгоритм Тарьяна);
- использовать алгоритмы поиска кратчайшего пути (Дейкстра, Флойд-Уоршелл) для анализа сетей, маршрутизации, планирования;
- работать с бинарными деревьями поиска (BST): вставка, удаление, поиск, обходы, балансировка.

владеть:

- навыком анализа алгоритмов с точки зрения эффективности и выбора подходящих подходов для оптимизации времени и памяти;
- навыком решения сложных прикладных задач, включая оптимизацию, маршрутизацию, обработку строк и графов;
- навыком оптимизации алгоритмов и структур данных для работы с большими объёмами данных;
- навыком разработки алгоритмов с использованием динамического программирования, жадных методов и методов строкового поиска для прикладных задач;
- навыком адаптации алгоритмов и выбора оптимальных структур данных (графы, деревья, хеш-таблицы) для специфических задач;
- навыком анализа графовых структур и выявления их особенностей, таких как связность, наличие циклов, порядок выполнения задач;
- навыком проектирования и разработки приложений, использующих структуры данных, такие как деревья, графы, хеш-таблицы, для эффективной организации и обработки данных.

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)
УК-6.	Способен определять и реализовывать приоритеты собственной деятельности и способы ее совершенствования на основе самооценки	УК-6.1.	Знает основные методы самооценки и анализа своей деятельности, а также принципы управления временем и целеполагания
		УК-6.2.	Умеет ставить реалистичные и достижимые цели, определять приоритеты в своей деятельности, а также разрабатывать и внедрять планы по совершенствованию своих навыков и компетенций на основе полученной самооценки
		УК-6.3.	Имеет практический опыт применения методов самооценки в своей профессиональной деятельности, включая участие в тренингах, семинарах и проектах, направленных на развитие личной эффективности и профессионального роста
ОПК-1.	Способен находить, формулировать и решать актуальные и значимые проблемы прикладной и компьютерной математики	ОПК-1.1.	Знает основные методы и подходы к решению задач прикладной и компьютерной математики, включая алгоритмы, математическое моделирование и теорию оптимизации, а также современные инструменты и технологии, используемые в этой области
		ОПК-1.2.	Умеет анализировать и формулировать математические задачи, применять соответствующие методы и алгоритмы для их решения, а также интерпретировать и представлять результаты в понятной и доступной форме
		ОПК-1.3.	Имеет практический опыт работы над проектами или исследованиями в области прикладной и компьютерной математики, включая участие в конкурсах, олимпиадах или научных публикациях, где были решены актуальные и значимые задачи
ПК-1.	Способен определять общие формы и закономерности области управления командами разработки	ПК-1.1.	Знает основные теоретические концепции и принципы, относящиеся к области управления командами разработки, а также ключевые закономерности и модели, которые помогают в анализе и интерпретации данных
		ПК-1.2.	Умеет проводить систематический анализ области управления командами разработки, выявлять и формулировать

			общие закономерности и тенденции, а также применять методы исследования для получения новых знаний и понимания
		ПК-1.3.	Имеет практический опыт работы в области управления командами разработки, включая участие в научных проектах, исследованиях или практических заданиях, где были выявлены и описаны общие формы и закономерности
ПК-3.	Способен решать задачи профессиональной деятельности, формулировать результат, увидеть следствия полученного результата	ПК-3.1.	Знает основные принципы и методы решения задач профессиональной деятельности, а также способы формулирования и представления результатов, включая анализ последствий и их значимость в контексте проекта
		ПК-3.2.	Умеет применять математические и компьютерные методы для решения конкретных задач, формулировать четкие и обоснованные результаты, а также анализировать их последствия для дальнейших действий и решений
		ПК-3.3.	Имеет практический опыт в решении профессиональных задач, включая участие в проектах, где были получены результаты и проанализированы их следствия, что способствовало принятию обоснованных решений
ПК-6	Способен разрабатывать программное обеспечение для решения прикладных задач в сфере профессиональной деятельности	ПК-6.1.	Знает основные языки программирования, методы разработки программного обеспечения, а также принципы проектирования и архитектуры программных систем, применяемых в конкретной предметной области
		ПК-6.2.	Умеет анализировать прикладные задачи, разрабатывать алгоритмы и реализовывать их в виде программного обеспечения, используя современные инструменты и технологии, а также проводить тестирование и отладку созданных решений
		ПК-6.3.	Имеет практический опыт разработки программного обеспечения в рамках реальных проектов, включая участие в командах, где были успешно реализованы решения для конкретных прикладных задач в сфере профессиональной деятельности

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостояте льная работа	
		Лекции	Семинары (практичес кие занятия)			
1	Сложность, итеративные сортировки и бинарный поиск	2	2		13	Домашнее задание Квиз
2	Рекурсия, рекурсивные сортировки и куча	2	2		13	Домашнее задание Квиз
3	Динамическое программирование и задачи о подпоследовательностях	2	2		13	Домашнее задание Квиз
4	Жадные алгоритмы и расстояние Левенштейна	3	3		14	Домашнее задание Квиз
5	Поиск строки в тексте	2	2	2	13	Контрольная работа
6	Хеширование	3	3		14	Домашнее задание Квиз
7	Графы	2	2		13	Домашнее задание Квиз
8	Обход графа в глубину. Связность и сортировка	3	3		14	Домашнее задание Квиз
9	Кратчайший путь в графе	2	2		13	Домашнее задание Квиз
10	Деревья	3	3	2	14	Контрольная работа
	Зачет с оценкой			4		Проект
Итого:		24	24	8	134	
Объем дисциплины (модуля) (в ак. ч.)		190				
Объем дисциплины (модуля) (в зач. ед.)		5				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Сложность, итеративные сортировки и бинарный поиск	Асимптотическая сложность алгоритмов Бинарный поиск Сортировка бинарными вставками
2	Рекурсия, рекурсивные сортировки и куча	Рекурсия Сортировка слиянием Куча и сортировка кучей
3	Динамическое программирование и задачи о подпоследовательностях	Динамическое программирование Задачи о подпоследовательностях
4	Жадные алгоритмы и расстояние Левенштейна	Жадные алгоритмы Расстояние Левенштейна

5	Поиск строки в тексте	Префикс-функция Алгоритм Кнута-Морриса-Пратта
6	Хеширование	Хеширование Хеш-таблицы Алгоритм Рабина-Карпа
7	Графы	Графы Представление графа Обход графа в ширину
8	Обход графа в глубину. Связность и сортировка	Обход в глубину Топологическая сортировка Связность графов
9	Кратчайший путь в графе	Алгоритм Дейкстры Алгоритм Флойда-Уоршелла
10	Деревья	Деревья Операции на BST Обходы дерева

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. 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>.

2. Шкодина, Т. А. Алгоритмы и структуры данных в Python. Лабораторный практикум : учебное пособие / Т. А. Шкодина. - Ростов-на-Дону : Издательско-полиграфический комплекс РГЭУ (РИНХ), 2022. - 80 с. - ISBN 978-5-7972-3037-3. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2213221>.

3. Лонца, А. Алгоритмы обучения с подкреплением на Python. Описание и разработка алгоритмов искусственного интеллекта : практическое руководство / А. Лонцапер ; пер. с англ. А. А. Слинкина. - Москва : ДМК Пресс, 2020. - 288 с. - ISBN 978-5-97060-855-5. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2260350>.

4. Истратова, Е. Е. Алгоритмы и структуры данных : учебное пособие / Е. Е. Истратова, И. Н. Томилов, А. А. Якименко. - Новосибирск : Изд-во НГТУ, 2025. - 68 с. - ISBN 978-5-7782-5502-9. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2244205>.

Дополнительная литература:

1. Алгоритмы. Руководство по разработке / С. Скиена; пер. с англ. под ред. А. А. Зубарева. — Москва: ДМК Пресс, 2024. — 464 с

2. Искусство программирования. Том 1. Основные алгоритмы / Д. Е. Кнут; пер. с англ. под ред. А. Л. Френкеля. — 3-е изд., перераб. и доп. — Москва: Вильямс, 2023. — 704 с.

3. Грокаем алгоритмы: Иллюстрированное пособие для программистов и любопытствующих / Дж. Грокот; пер. с англ. под ред. И. В. Смирнова. — Москва: МИФ, 2024. — 288 с.

4. Алгоритмы и структуры данных на Python / Quantum Technologies; пер. с англ. под ред. Гринчик Н. — Питер, 2024. — 368 с.

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	https://elibrary.ru/defaultx.asp
2.	База данных для IT-специалистов	https://habr.com
3.	База данных ScienceDirect	https://www.sciencedirect.com
4.	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
5.	Федеральный портал «Российское образование»	https://www.edu.ru/
6.	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
7.	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
8.	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/

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
Операционные системы:		
Microsoft Imagine (Windows Client, Server)	зарубежное	лицензионное
Браузеры:		
Яндекс.Браузер	отечественное	свободно распространяемое
Google Chrome	зарубежное	свободно распространяемое
Офисные приложения:		
Microsoft Imagine (Visio, OneNote)	зарубежное	лицензионное
TeXstudio	зарубежное	свободно распространяемое
Adobe Acrobat Reader	зарубежное	свободно распространяемое
Программное обеспечение для планирования и учета времени:		
Toggle app	зарубежное	свободно распространяемое
Системы управления проектами:		
Microsoft Imagine (Project)	зарубежное	лицензионное

Системы управления базами данных:		
Microsoft Imagine (SQL Server)	зарубежное	лицензионное
Системы резервного копирования (backup):		
Acronis Backup Advanced for HyperV	зарубежное	лицензионное
Справочно-правовые системы:		
КонсультантПлюс: справочно-правовая система	отечественное	лицензионное
Средства антивирусной защиты:		
Kaspersky Endpoint Security для бизнеса Стандартный Russian Edition	отечественное	лицензионное
Среды разработки:		
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
Пакеты программных средств и библиотек:		
AutoPsy	зарубежное	свободно распространяемое
Interactive Disassembler (IDA)	зарубежное	свободно распространяемое
Системы управления библиографической информацией:		
Zotero	зарубежное	свободно распространяемое
Сервисы и службы:		
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» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, квизы, контрольные работы, проект, 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.

Семинар – это форма учебной деятельности, проводимая в учебном заведении под руководством преподавателя, где студенты активно участвуют в обсуждениях, практических заданиях и других формах взаимодействия.

Для успешной подготовки к семинару рекомендуется заранее ознакомиться с темой занятия и основными материалами, чтобы иметь возможность активно участвовать в обсуждении. Также полезно подготовить вопросы и идеи для обсуждения, что поможет глубже понять материал и продемонстрировать заинтересованность.

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.

Квиз – это интерактивное тестирование, направленное на проверку знаний и понимания изучаемого материала.

Для успешной подготовки к квизу рекомендуется внимательно изучить основные понятия и методы, изучаемые на курсе. Полезно решать практические задачи и примеры, чтобы закрепить теоретические знания. Также стоит ознакомиться с типичными вопросами и форматами заданий, чтобы лучше подготовиться к тестированию.

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 и продемонстрировать навыки их практического применения.

Проект – исследовательская работа по дисциплине (модулю) и презентация результатов.

Для успешной подготовки к проекту рекомендуется: четко определить цели и задачи проекта; составить план работы, разбив проект на этапы с указанием сроков выполнения каждого из них; использовать разнообразные источники информации и инструменты для исследования темы; регулярно проверять прогресс и вносить коррективы в план, если это необходимо.

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»

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 **зачета с оценкой**, 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, если иное не указано в силлабусе дисциплины:

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.
9	Отлично	
8	Отлично	

Десятибалльная оценка	Пятибалльная оценка	Общая характеристика результата обучения по дисциплине (модулю)
		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.
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, 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.

Дисциплина (модуль) «Алгоритмы» оценивается следующим образом:

Активность	Вес	Количество	Описание
Домашние задания	40%	10	Набор задач по темам недели
Контрольные работы	15%	2	Письменная работа с набором задач, которые нужно решить за ограниченное время
Квизы	5%	10	Интерактивное тестирование, направленное на проверку знаний и понимания изучаемого материала
Зачет с оценкой	40%	1	Проект – исследовательская работа по дисциплине (модулю) и презентация результатов

Формула расчёта итоговой оценки по дисциплине (модулю) «Алгоритмы»: $0,4 \times \text{Среднее за домашние задания} + 0,15 \times \text{Среднее за контрольные работы} + 0,05 \times \text{Среднее за квизы} + 0,4 \times \text{Зачет с оценкой}$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1. Анализ и реализация сортировок

1. Реализуйте алгоритмы сортировки: вставками, слиянием и кучей на языке Python или C++.
2. Для каждого алгоритма:
 - измерьте время выполнения на массивах разного размера (100, 1000, 10 000 элементов),
 - сравните с теоретической сложностью.
3. Постройте график зависимости времени от размера массива.
4. Напишите вывод: в каких случаях каждый алгоритм наиболее эффективен.
5. Оформите код, графики и пояснения в виде отчёта.

Домашнее задание 2. Динамическое программирование и строки

1. Реализуйте алгоритм вычисления расстояния Левенштейна между двумя строками с восстановлением последовательности операций (вставка, удаление, замена).
2. Примените его для сравнения пар строк:
 - "алгоритм" и "логарифм".
3. Реализуйте поиск наибольшей общей подпоследовательности (НОП) для двух последовательностей.
4. Оцените временную и пространственную сложность обоих алгоритмов.
5. Оформите результаты с пояснением логики заполнения таблицы.

Домашнее задание 3. Графовые алгоритмы

1. Дан неориентированный граф (задан списком смежности).
2. Реализуйте:
 - обход в ширину (BFS) и определите расстояния от стартовой вершины,
 - обход в глубину (DFS) и определите количество компонент связности.
3. Для ориентированного ациклического графа выполните топологическую сортировку.
4. Постройте визуальное представление графа (можно вручную или с помощью библиотеки).
5. Оформите код, результаты и анализ сложности.

Примерные задания для контрольной работы

Задание 1. Бинарный поиск

Напишите алгоритм бинарного поиска в отсортированном массиве. Примените его для поиска элемента 14 в массиве: [3, 7, 9, 11, 14, 18, 22, 25]

Опишите каждый шаг: какие значения принимают левая и правая границы, как вычисляется середина, когда находится искомый элемент.

Задание 2. Сортировка слиянием

Опишите принцип работы алгоритма сортировки слиянием. Выполните сортировку массива [38, 27, 43, 3, 9, 82, 10] по шагам:

1. Разбейте массив на подмассивы до единичных элементов.
2. Покажите процесс слияния пар отсортированных подмассивов.
3. Запишите окончательный отсортированный результат.

Задание 3. Динамическое программирование (НОП)

Найдите длину наибольшей общей подпоследовательности (НОП) для строк: X = "AGGTAB" и Y = "GXTXAYB"

Постройте таблицу динамического программирования (без рисования — опишите логику заполнения) и восстановите саму подпоследовательность, указав путь от конца к началу.

Задание 4. Жадный алгоритм (задача о выборе заявок)

Даны интервалы времени выполнения задач:

(1, 4), (3, 5), (0, 6), (5, 7), (8, 9), (5, 9)

Используя жадный алгоритм (по принципу наименьшего времени окончания), выберите максимальное количество непересекающихся задач.

Опишите порядок выбора и запишите итоговый набор.

Задание 5. Расстояние Левенштейна

Вычислите расстояние Левенштейна между строками:

"kitten" и "sitting"

Опишите, как можно получить вторую строку из первой, и укажите минимальное количество операций (вставка, удаление, замена).

Задание 6. Алгоритм Кнута-Морриса-Пратта (КМП)

Для строки $P = \text{"ABABC"}$ постройте префикс-функцию (без таблицы — в виде списка значений).

Объясните, как используется префикс-функция при поиске подстроки в тексте, и укажите её назначение.

Задание 7. Бинарное дерево поиска (BST)

Даны значения: 8, 3, 10, 1, 6, 14, 4, 7, 13.

1. Постройте бинарное дерево поиска, вставляя элементы по порядку.
2. Выполните три обхода: pre-order, in-order, post-order — и запишите последовательности.
3. Объясните, какое свойство сохраняется в BST и как оно используется при поиске.

Примерные задания для квиза**Вопрос 1.**

Какова временная сложность бинарного поиска в отсортированном массиве?

1. $O(n)$
2. $O(\log n)$
3. $O(n \log n)$
4. $O(1)$

Правильный ответ: 2

Вопрос 2.

Какой алгоритм сортировки имеет наилучшую среднюю временную сложность $O(n \log n)$ и использует метод «разделяй и властвуй»?

1. Сортировка вставками
2. Сортировка слиянием
3. Сортировка выбором
4. Пузырьковая сортировка

Правильный ответ: 2

Вопрос 3.

Какой алгоритм используется для нахождения кратчайших путей от одной вершины ко всем остальным в графе с неотрицательными весами?

1. Алгоритм Флойда-Уоршелла
2. Алгоритм Краскала
3. Алгоритм Дейкстры
4. Алгоритм Прима

Правильный ответ: 3

Вопрос 4.

Что делает префикс-функция в алгоритме Кнута-Морриса-Пратта?

1. Считает количество вхождений подстроки
2. Определяет длину самого длинного собственного суффикса, совпадающего с префиксом

3. Строит хеш строки
4. Находит медиану подстроки

Правильный ответ: 2

Вопрос 5.

Какой обход бинарного дерева посещает вершину после рекурсивной обработки левого и правого поддеревьев?

1. Pre-order
2. In-order
3. Post-order
4. Level-order

Правильный ответ: 3

Примерное описание задания и критерии оценивания к проекту

Описание проектного задания: разработайте приложение для анализа биологической последовательности (например, ДНК или белка) с использованием алгоритмов.

Задачи проекта:

- Поиск всех вхождений подпоследовательности (реализация алгоритма Рабина-Карпа или КМП).
- Вычисление расстояния Левенштейна между двумя последовательностями.
- Построение дерева поиска для хранения частот встречаемости k-меров (подстрок длины k).
- Визуализация результатов (таблицы, графики).
- Формат: программа на Python/C++ с интерфейсом командной строки или Jupyter-ноутбук с пояснениями.

Критерии оценивания проекта:

- Реализация алгоритмов — корректность и эффективность реализованных алгоритмов (поиск, Левенштейн, дерево).
- Использование структур данных — обоснованный выбор (хеш-таблицы, деревья, графы) и корректная реализация.
- Анализ сложности — оценка временной и пространственной сложности для ключевых функций.
- Функциональность и тестирование — программа работает на разных входных данных, есть тесты.
- Оформление и документация — понятный код, комментарии, отчёт с описанием, выводами.
- Креативность и прикладная направленность — задача имеет биологическую/практическую мотивацию.
- Презентация и защита — ясность изложения, ответы на вопросы, демонстрация работы.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Как обозначается порядок роста времени выполнения алгоритма при увеличении входных данных?	асимптотическая сложность	УК-6
2.	Какую временную сложность имеет бинарный поиск в отсортированном массиве?	$O(\log n)$	УК-6
3.	Как называется алгоритм поиска элемента путём последовательного деления интервала пополам?	бинарный поиск	УК-6

4.	Какой принцип лежит в основе сортировки бинарными вставками?	вставка с использованием бинарного поиска	УК-6
5.	Как называется способ определения результата задачи через решение её подзадач того же типа?	рекурсия	ОПК-1
6.	Какую сложность имеет сортировка слиянием в худшем случае?	$O(n \log n)$	ОПК-1
7.	Как называется структура данных в виде почти полного бинарного дерева с упорядоченными приоритетами?	куча	ОПК-1
8.	Какой алгоритм сортировки использует структуру кучи для упорядочивания элементов?	сортировка кучей	ОПК-1
9.	Какой метод применяется для решения задач с перекрывающимися подзадачами и запоминанием результатов?	динамическое программирование	ПК-1
10.	Какая задача требует нахождения самой длинной общей подпоследовательности двух строк?	задача LCS	ПК-1
11.	Какой подход выбирает локально оптимальное решение на каждом шаге?	жадный алгоритм	ПК-1
12.	Какая метрика определяет минимальное количество операций вставки, удаления и замены для преобразования строки?	расстояние Левенштейна	ПК-1
13.	Как называется массив, хранящий длины наибольших собственных префиксов строки, совпадающих с её суффиксами?	префикс-функция	ПК-3
14.	Какой алгоритм поиска подстроки использует префикс-функцию для ускорения сопоставления?	алгоритм Кнута–Морриса–Пратта	ПК-3
15.	Как называется структура данных для хранения пар «ключ–значение» с быстрым доступом?	хеш-таблица	ПК-3
16.	Какой алгоритм поиска подстроки использует хеширование для сравнения строк?	алгоритм Рабина–Карпа	ПК-3
17.	Как называется структура, состоящая из вершин и рёбер?	граф	ПК-6
18.	Какой алгоритм используется для обхода графа по уровням?	обход в ширину	ПК-6
19.	Как называется алгоритм упорядочивания вершин ориентированного ациклического графа?	топологическая сортировка	ПК-6
20.	Какой алгоритм находит кратчайшие пути от одной вершины до всех остальных при неотрицательных весах?	алгоритм Дейкстры	ПК-6