

УТВЕРЖДЕНА

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

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

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Машинное обучение

Квалификация (степень) выпускника: магистр

Форма обучения: очная (с применением ДОТ)

Срок освоения программы: 2 года

Год набора: 2025

**Москва
2025**

Содержание

1. Краткая характеристика дисциплины (модуля)	3
2. Перечень планируемых результатов обучения.....	6
3. Тематический план.....	8
4. Содержание дисциплины (модуля).....	8
5. Учебно-методическое обеспечение	10
6. Материально-техническое обеспечение	10
7. Методические и оценочные материалы	12

1. Краткая характеристика дисциплины (модуля)

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

Изучение дисциплины (модуля) «Алгоритмы» предоставляет студентам фундаментальные знания о методах решения задач и оптимизации процессов, что является ключевым аспектом в программировании и разработке программного обеспечения. Эти навыки позволяют будущим специалистам эффективно анализировать, разрабатывать и внедрять алгоритмические решения, что существенно повышает их конкурентоспособность на рынке труда.

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 Машинное обучение и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 1 курсе во 2 семестре.

Цель изучения дисциплины (модуля): формирование умения разрабатывать, анализировать и эффективно применять алгоритмические методы для решения различных вычислительных задач.

Задачи изучения дисциплины (модуля):

- изучить основные принципы оценки эффективности алгоритмов по времени и памяти, включая различные типы сложности и методы анализа;
- освоить реализацию и модификацию базовых алгоритмов поиска и сортировки для решения задач с различными требованиями к данным;
- разработать навыки применения рекурсии в алгоритмах, включая оптимизацию для избежания избыточных вычислений;
- применить динамическое программирование для решения задач оптимизации путем разбиения на подзадачи и использования мемоизации;
- использовать жадные стратегии для решения оптимизационных задач, оценивая их преимущества и ограничения;
- реализовать алгоритмы для обработки и поиска в строках, включая вычисление расстояний и префикс-функций;
- освоить структуры данных, такие как графы, деревья и хеш-таблицы, для моделирования и анализа реальных задач, включая поиск путей и связности.

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;
- рекурсию: 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;
- теорию 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)
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-5.	Способен передавать результат решенных прикладных задач в виде конкретных рекомендаций, выраженных в терминах области машинного обучения	ПК-5.1.	Знает основные методы и подходы к формулированию рекомендаций на основе результатов решения прикладных задач, а также термины и концепции, специфичные для области машинного обучения
		ПК-5.2.	Умеет анализировать результаты решенных задач и формулировать четкие, конкретные рекомендации, адаптируя их к требованиям и ожиданиям целевой аудитории
		ПК-5.3.	Имеет практический опыт в разработке и представлении рекомендаций на основе анализа прикладных задач, включая участие в проектах, где результаты были успешно

			применены и оценены в контексте области машинного обучения
--	--	--	--

3. Тематический план

№ п/п	Наименование раздела дисциплины (модуля)	Трудоемкость, академические часы				ТКУ (текущий контроль успеваемости)
		Очная форма				
		Аудиторная работа		Контроль	Самостоятельная работа	
Лекции	Семинары (практические занятия)					
1	Сложность, итеративные сортировки и бинарный поиск	2	2		14	Домашние задания, Квиз
2	Рекурсия, рекурсивные сортировки и куча	2	2		15	Домашние задания, Контест
3	Динамическое программирование и задачи о подпоследовательностях	2	2		14	Домашние задания, Квиз
4	Жадные алгоритмы и расстояние Левенштейна	2	2		15	Домашние задания, Контест
5	Поиск строки в тексте	2	2		15	Домашние задания, Проект
6	Хеширование	2	2		14	Домашние задания, Квиз
7	Графы	2	2		15	Домашние задания, Контест
8	Обход графа в глубину. Связность и сортировка	2	2		14	Домашние задания, Квиз
9	Кратчайший путь в графе	2	2		15	Домашние задания, Контест
10	Деревья	2	2		15	Домашние задания, Проект
	<i>Зачет с оценкой</i>			4		
	Итого:	20	20	4	146	
	<i>Объем дисциплины (модуля) (в ак. ч.)</i>	190				
	<i>Объем дисциплины (модуля) (в зач. ед.)</i>	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. 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. 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. 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. Рафгарден, Т. Совершенный алгоритм. Жадные алгоритмы и динамическое программирование : практическое руководство / Т. Рафгарден. - Санкт-Петербург : Питер, 2020. - 256 с. - (Серия «Библиотека программиста»). - ISBN 978-5-4461-1445-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/175612>.

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, проекты, контесты, 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, уделяя особое внимание их применению и алгоритмам. Полезно решать практические задачи и примеры, чтобы закрепить теоретические знания. Также стоит ознакомиться с типичными вопросами и форматами заданий, чтобы лучше подготовиться к тестированию.

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

Для успешной подготовки к проекту: четко определите цели и задачи проекта, распределите роли и обязанности между участниками, а также установите сроки выполнения каждой части работы. Регулярно проводите встречи для обсуждения прогресса и решения возникающих вопросов.

Контекст – интерактивная платформа с заданиями разного уровня сложности и автоматической проверкой результатов.

Контекст позволяет оперативно оценивать усвоение материала и выявлять пробелы в знаниях через тесты и практические задачи. Такой формат способствует регулярной самопроверке и повышает мотивацию к изучению дисциплины (модуля).

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:

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

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			<p>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.</p>
7	Хорошо	Зачтено	Студент обладает знаниями предмета
6	Хорошо	Зачтено	<p>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.</p>
5	Удовлетворительно	Зачтено	Студент обладает базовыми знаниями по
4	Удовлетворительно	Зачтено	<p>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.</p>
3	Не сдан	Не зачтено	<p>Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные</p>

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			наводящие вопросы.

Дисциплина (модуль) «Алгоритмы» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	40%	Набор задач по темам недели
Проекты	40%	Исследовательская работа по дисциплине (модулю) и презентация результатов
Контесты	15%	Интерактивная платформа с заданиями разного уровня сложности и автоматической проверкой результатов
Квизы	5%	Решение интерактивных тестов

Формула расчёта итоговой оценки по дисциплине (модулю) «Алгоритмы»: $\langle 0,4 \times \text{среднее за домашние задания} + 0,4 \times \text{среднее за проекты} + 0,15 \times \text{среднее за контесты} + 0,05 \times \text{среднее за квизы} \rangle$.

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание 1.

Задание 1.

Дано три алгоритма:

- Алгоритм А имеет два вложенных цикла по n элементов.
- Алгоритм В — рекурсивный, разбивающий задачу пополам на каждом шаге, с константной работой на каждом уровне.
- Алгоритм С — один цикл по n элементам с вложенным циклом, который в худшем случае выполняется k раз (k — константа).

Определите асимптотическую временную сложность каждого алгоритма и обоснуйте ответ.

Задание 2.

Напишите рекурсивную функцию на любом языке программирования для вычисления n -го числа Фибоначчи. Затем оптимизируйте функцию с помощью мемоизации и сравните время выполнения для $n=40$.

Задание 3.

Реализуйте алгоритмы сортировки:

- сортировку вставками,
- сортировку слиянием.

Проведите сравнительный анализ их временной сложности на массивах размером 1000, 5000 и 10000 элементов, заполненных случайными числами.

Задание 4.

Реализуйте минимальную кучу (min-heap) с операциями вставки и извлечения минимального элемента. Проверьте корректность работы, вставив в кучу 15 случайных чисел и последовательно извлекая минимальные элементы.

Задание 5.

Реализуйте алгоритм пирамидальной сортировки (heapsort) на основе своей реализации кучи из предыдущего задания. Проверьте корректность сортировки массива из 20 случайных чисел.

Домашнее задание 2.

Задание 1.

Напишите программу, которая находит длину наибольшей общей подпоследовательности (НОП) для двух строк. Например, для строк "AGGTAB" и "GXTXAYB" НОП равна 4 ("GTAB"). Обоснуйте сложность вашего алгоритма и объясните, как вы используете динамическое программирование для решения этой задачи.

Задание 2.

Реализуйте алгоритм, который находит минимальное количество монет, необходимых для размена заданной суммы, используя монеты определенных номиналов (например, 1, 5, 10, 25). Проведите тестирование вашего алгоритма для различных сумм и наборов монет, а также проанализируйте его временную и пространственную сложность.

Задание 3.

Реализуйте жадный алгоритм для решения задачи о рюкзаке с дробными предметами. Дано множество предметов с весом и стоимостью, необходимо максимизировать общую стоимость, не превышая заданный вес рюкзака. Проведите тестирование на примерах и сравните результаты с результатами, полученными с использованием динамического программирования.

Задание 4.

Напишите программу для вычисления расстояния Левенштейна между двумя строками. Объясните, как работает ваш алгоритм, и проведите анализ его временной сложности. Дополнительно разработайте функцию, которая будет возвращать самую короткую последовательность операций (вставка, удаление, замена), необходимых для преобразования одной строки в другую.

Задание 5.

Используйте расстояние Левенштейна для реализации функции, которая будет находить наиболее похожие строки из списка на заданную строку. Например, если у вас есть список ["кот", "котик", "собака", "пес"], и вы ищете строку "кот", функция должна вернуть "кот" и "котик". Проведите тестирование на различных входных данных и проанализируйте производительность вашего алгоритма.

Домашнее задание 3.

Задание 1.

Напишите программу для поиска подстроки в строке с использованием наивного метода. Проведите тестирование на различных строках и подстроках, измерив время выполнения для разных размеров входных данных. Обоснуйте временную сложность вашего алгоритма.

Задание 2.

Реализуйте алгоритм Кнута — Морриса — Пратта для поиска подстроки в строке. Объясните, как работает алгоритм, и как он использует предварительное вычисление префикс-функции. Проведите тестирование на примерах и сравните время выполнения с наивным методом.

Задание 3.

Реализуйте функцию для хеширования строк с использованием простого хеш-функции (например, хеширование на основе суммы ASCII-кодов символов). Создайте структуру данных, которая будет хранить строки и их хеши, и реализуйте операции вставки и поиска. Проведите тестирование на коллизиях и объясните, как можно улучшить вашу хеш-функцию для уменьшения вероятности коллизий.

Задание 4.

Реализуйте алгоритм поиска подстроки в строке с использованием метода Рабина — Карпа, который основан на хешировании. Объясните, как работает ваш алгоритм, и как он обрабатывает коллизии. Проведите тестирование на различных строках и подстроках, сравнив результат с алгоритмом КМП.

Задание 5.

Расширьте ваш алгоритм Кнута — Морриса — Пратта, чтобы он мог находить все вхождения подстроки в строке, а не только первое. Создайте функцию, которая будет возвращать список всех индексов вхождений подстроки. Проведите тестирование на различных примерах и проанализируйте производительность вашего алгоритма.

Примерные задания для квизов

Квиз 1.

Вопрос 1.

Какова основная идея динамического программирования?

- A. Разделить задачу на подзадачи и решать их независимо.
- B. Использовать рекурсию для решения задачи.
- C. Хранить результаты подзадач для предотвращения повторных вычислений.
- D. Использовать жадные методы для нахождения оптимального решения.

Ответ: C.

Вопрос 2.

Что такое наибольшая общая подпоследовательность (НОП)?

- A. Подпоследовательность, которая встречается в обеих строках в том же порядке.
- B. Подпоследовательность, которая является общей для двух строк, но не обязательно в том же порядке.
- C. Подпоследовательность, которая содержит все символы обеих строк.
- D. Подпоследовательность, которая является максимальной по длине.

Ответ: A.

Вопрос 3.

Какой из следующих алгоритмов является жадным для решения задачи о рюкзаке с дробными предметами?

- A. Алгоритм Дейкстры
- B. Алгоритм Кнута — Морриса — Пратта
- C. Жадный алгоритм, выбирающий предметы с наибольшей стоимостью на единицу веса
- D. Динамическое программирование

Ответ: C.

Вопрос 4.

Какое из следующих утверждений верно для расстояния Левенштейна?

- A. Оно всегда равно нулю.
- B. Оно измеряет количество операций вставки, удаления и замены, необходимых для преобразования одной строки в другую.
- C. Оно всегда меньше длины самой длинной строки.
- D. Оно используется только для поиска подстрок.

Ответ: B.

Вопрос 5.

Какова временная сложность алгоритма для вычисления расстояния Левенштейна с использованием динамического программирования?

- A. $O(n)$
- B. $O(m + n)$
- C. $O(m * n)$
- D. $O(\log n)$

Ответ: C.

Квиз 2.

Вопрос 1.

Какой из следующих методов является наивным способом поиска подстроки в строке?

- A. Сравнить каждый символ строки с первым символом подстроки.
- B. Использовать префикс-функцию.
- C. Применять хеширование для быстрого поиска.
- D. Использовать бинарный поиск.

Ответ: A.

Вопрос 2.

Какова основная идея алгоритма Кнута — Морриса — Пратта?

- A. Использовать жадный подход.
- B. Применять рекурсию.
- C. Вычислять префикс-функцию для оптимизации поиска.
- D. Хранить все возможные подстроки в хеш-таблице.

Ответ: C.

Вопрос 3.

Какой из следующих методов используется для уменьшения вероятности коллизий в хеш-таблицах?

- A. Линейное пробирование
- B. Метод цепочек
- C. Квадратное пробирование
- D. Все вышеперечисленные методы

Ответ: D.

Вопрос 4.

Какое значение имеет префикс-функция в алгоритме Кнута — Морриса — Пратта?

- A. Определяет длину совпадающих префиксов и суффиксов.
- B. Помогает в хешировании строк.
- C. Подсчитывает количество вхождений подстроки.
- D. Не имеет значения в данном алгоритме.

Ответ: A.

Вопрос 5.

Что происходит, когда две разные строки имеют одинаковое хеш-значение?

- A. Это нормально и не вызывает проблем.
- B. Это называется коллизией.
- C. Это приводит к ошибке.
- D. Хеш-таблица автоматически увеличивается.

Ответ: B.

Квиз 3.

Вопрос 1.

Какой основной принцип лежит в основе алгоритма обхода в ширину (BFS)?

- A. Посещение узлов по мере их добавления в стек.
- B. Посещение всех соседей узла перед переходом к следующему узлу.
- C. Использование рекурсии для обхода графа.
- D. Сравнение весов рёбер для выбора узлов.

Ответ: В.

Вопрос 2.

Какова основная цель алгоритма Дейкстры?

- А. Найти кратчайший путь от одной вершины ко всем остальным в графе с отрицательными весами.
- В. Найти кратчайший путь от одной вершины ко всем остальным в графе с неотрицательными весами.
- С. Найти все возможные пути в графе.
- Д. Определить, существует ли цикл в графе.

Ответ: В.

Вопрос 3.

Какой тип графов может обрабатывать алгоритм Флойда — Уоршелла?

- А. Только ориентированные графы.
- В. Только неориентированные графы.
- С. Графы с отрицательными весами.
- Д. Графы без рёбер.

Ответ: С.

Вопрос 4.

Какое основное свойство алгоритма Беллмана — Форда?

- А. Он работает только с неориентированными графами.
- В. Он может обнаруживать отрицательные циклы в графе.
- С. Он всегда находит кратчайший путь в графе с неотрицательными весами.
- Д. Он требует использования стека для хранения узлов.

Ответ: В.

Вопрос 5.

Какой из следующих алгоритмов наиболее подходит для нахождения кратчайших путей во взвешенном графе с отрицательными весами?

- А. Алгоритм Дейкстры
- В. Алгоритм обхода в ширину (BFS)
- С. Алгоритм Флойда — Уоршелла
- Д. Алгоритм Беллмана — Форда

Ответ: D.

Примерное описание к проекту

Цели проекта:

1. Изучить и проанализировать различные алгоритмы и структуры данных, их асимптотическую сложность и применение.
2. Реализовать несколько алгоритмов, включая сортировки, жадные алгоритмы и динамическое программирование.
3. Решить практические задачи, связанные с использованием этих алгоритмов и структур данных.

Задания:

1. Асимптотическая сложность и рекурсия:

- Изучите основные понятия асимптотической сложности (O , Θ , Ω).
- Реализуйте рекурсивные функции для вычисления факториала и чисел Фибоначчи.
- Проанализируйте временную сложность каждой из функций.

2. Сортировки и структура данных куча:

- Реализуйте алгоритмы сортировки: пузырьковую, быструю и сортировку слиянием.
- Реализуйте структуру данных "куча" (минимальная и максимальная).
- Напишите алгоритм сортировки на основе кучи (Heap Sort) и проанализируйте его сложность.

3. Динамическое программирование:

- Изучите методы динамического программирования и их применение к задачам.
- Реализуйте алгоритм для нахождения длины наибольшей общей подпоследовательности (LCS).
- Решите задачу о размене монет с использованием динамического программирования.

4. Жадные алгоритмы и задача о рюкзаке:

- Изучите жадные алгоритмы и их применение.
- Реализуйте жадный алгоритм для решения задачи о рюкзаке.
- Сравните эффективность жадного алгоритма с динамическим программированием для этой задачи.

5. Расстояние Левенштейна:

- Изучите алгоритм вычисления расстояния Левенштейна между двумя строками.
- Реализуйте этот алгоритм и проанализируйте его временную и пространственную сложность.
- Примените алгоритм для решения задачи исправления опечаток в текстах.

Ожидаемые результаты:

- Создание документации, описывающей каждый реализованный алгоритм, его сложность и примеры использования.
- Подготовка презентации, в которой будут представлены основные выводы и результаты работы над проектом.
- Разработка тестов для проверки корректности и эффективности реализованных алгоритмов.

Дополнительные рекомендации:

- Используйте графические представления для иллюстрации работы алгоритмов, например, визуализации сортировок или динамического программирования.
- Рассмотрите возможность использования языков программирования, таких как Python или Java, для реализации алгоритмов.

Примерные задания для контестов

Задание 1.

Напишите функцию, которая принимает строку и подстроку, и использует алгоритм Бойера — Мура для поиска подстроки в строке. Выведите индексы, на которых начинается подстрока.

Задание 2.

Реализуйте алгоритм КМП для поиска подстроки в строке. Напишите тесты, которые проверяют корректность работы алгоритма на различных примерах.

Задание 3.

Реализуйте хеш-таблицу с использованием метода открытой адресации. Реализуйте операции вставки, поиска и удаления элементов. Проверьте работу хеш-таблицы на наборе данных.

Задание 4.

Напишите функцию, которая выполняет обход в ширину на заданном графе (представленном в виде списка смежности). Выведите порядок посещения вершин, начиная с заданной начальной вершины.

Задание 5.

Реализуйте алгоритм обхода в глубину для графа, представленного в виде матрицы смежности. Выведите порядок посещения вершин, начиная с заданной начальной вершины.

Задание 6.

Напишите функцию, реализующую алгоритм Дейкстры для нахождения кратчайшего пути от одной вершины ко всем остальным в графе с неотрицательными весами. Выведите

кратчайшие расстояния до всех вершин.

Задание 7.

Реализуйте алгоритм Флойда — Уоршелла для нахождения кратчайших путей между всеми парами вершин в графе. Выведите матрицу кратчайших расстояний.

Задание 8.

Напишите функцию, реализующую алгоритм Беллмана — Форда для нахождения кратчайших путей в графе с возможными отрицательными весами. Проверьте алгоритм на графе с отрицательными циклами.

Задание 9.

Напишите программу, которая сравнивает время выполнения алгоритмов Дейкстры и Флойда — Уоршелла на различных графах (с различным количеством вершин и рёбер). Выведите результаты сравнения.

Задание 10.

Реализуйте алгоритм, который использует BFS для нахождения кратчайшего пути в графе с равными весами рёбер. Проверьте его на примере лабиринта, представленном в виде матрицы.

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Какой из перечисленных алгоритмов сортировки обладает асимптотически самым большим временем работы в среднем случае? А. Сортировка слиянием Б. Быстрая сортировка В. Пирамидальная сортировка Г. Сортировка вставками	Г	ОПК-1
2.	Для поиска значения элемента в массиве в Python можно использовать метод <code>index()</code> . Какую оптимизацию можно выполнить, если известно, что массив отсортирован?	Бинарный поиск/бинарный поиск	ОПК-1
3.	Какой алгоритм позволяет узнать связность графа компьютерной или социальной сети? Ответ дайте аббревиатурой.	DFS/BFS/dfs/bfs	ПК-5
4.	Какой подход к решению алгоритмических задач используется для ускорения вычисления чисел Фибоначчи?	Динамическое программирование/динамическое программирование / кэширование	ПК-5
5.	Какую временную сложность имеет алгоритм бинарного поиска в худшем случае?	$O(\log n)$	ОПК-1
6.	Порекомендуйте алгоритм для топологической сортировки зависимостей в моделях машинного обучения.	Обход в глубину	ПК-5
7.	Чему равна пространственная сложность сортировки слиянием для массива из n элементов?	$O(n)$	ОПК-1
8.	Порекомендуйте алгоритм для вычисления расстояния между строками в задачах обработки текста.	Левенштейн	ПК-5

9.	Какое максимальное количество рекурсивных вызовов может произойти при вычислении числа Фибоначчи рекурсивно для $n=5$?	15	ОПК-1
10.	Порекомендуйте метод для обнаружения связанных компонент в графах социальных сетей.	Обход в ширину	ПК-5
11.	Какую сложность имеет динамическое программирование для задачи о наибольшей общей подпоследовательности в худшем случае?	$O(n \cdot m)$	ОПК-1
12.	Порекомендуйте структуру для хранения уникальных элементов с быстрым доступом в больших данных.	Хеш-таблица	ПК-5
13.	Какое минимальное количество монет нужно использовать жадным алгоритмом для сдачи 63 копеек монетами достоинством 1, 2, 5, 10, 25?	6	ОПК-1
14.	Порекомендуйте метод для обработки строковых данных в задачах машинного обучения, таких как поиск паттернов.	Рабина-Карпа	ПК-5
15.	Чему равно расстояние Левенштейна между строками "abc" и "ab"?	1	ОПК-1
16.	Порекомендуйте алгоритм для поиска кратчайшего пути в графе без отрицательных весов.	Дейкстра	ПК-5
17.	Какую временную сложность имеет алгоритм Рабина-Карпа в худшем случае для поиска подстроки?	$O(n \cdot m)$	ОПК-1
18.	Порекомендуйте структуру данных для быстрого поиска элементов в отсортированном массиве.	Бинарное дерево поиска	ПК-5
19.	Сколько вершин может содержать полный граф с 4 вершинами?	6	ОПК-1