

УТВЕРЖДЕНА

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

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

Направление подготовки: 02.04.01 Математика и компьютерные науки

Направленность (профиль) подготовки: Backend-разработка

Квалификация (степень) выпускника: магистр

Форма обучения: очная

Срок освоения программы: 2 года

Год набора: 2024

**Москва
2024**

Содержание

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

1. Краткая характеристика дисциплины (модуля)

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

Изучение дисциплины (модуля) «Алгоритмы Java» формирует у студентов знания алгоритмов и структур данных, которые являются основой для разработки оптимизированных и производительных приложений, что критически важно в условиях современных требований к программному обеспечению. Освоение этих понятий также развивает логическое мышление и способности к решению сложных задач.

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 Backend-разработка и входит в обязательную часть Блока 1.

Дисциплина (модуль) изучается на 1 курсе в 1 семестре.

Цель изучения дисциплины (модуля): формирование у студентов знания основных алгоритмических концепций и структур данных с использованием языка Java для эффективного решения задач программирования.

Задачи изучения дисциплины (модуля):

- формирование знания принципов работы контейнеров в стандартной библиотеке любого языка программирования, в частности, Java;
- формирование знания различных вариантов решения задачи о сортировке, о видах разрешения коллизий;
- формирование умения использовать подходящие инструменты из стандартной библиотеки языка программирования для реализации алгоритмов;
- формирование умения оценивать сложность и время работы произвольного алгоритма;
- формирование навыка оценивать различные уже существующие решения алгоритмических задач и, при необходимости, улучшать их;
- формирование навыка выбирать подходящие оптимальные алгоритмы для решения прикладных задач.

2. Перечень планируемых результатов обучения

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

Компетенция	Содержание компетенции	Индикатор компетенции	Перечень планируемых результатов обучения по дисциплине (модулю)
ОПК-3.	Способен самостоятельно создавать прикладные программные средства на основе современных информационных технологий и сетевых ресурсов, в том числе отечественного производства	ОПК-3.1.	Знает основные принципы программирования, архитектуры программного обеспечения и современные языки программирования, а также особенности отечественных информационных технологий и сетевых ресурсов
		ОПК-3.2.	Умеет разрабатывать прикладные программные средства, используя современные инструменты и технологии, а также интегрировать их с сетевыми ресурсами для решения конкретных задач
		ОПК-3.3.	Имеет практический опыт разработки программных средств, используемых при построении математических моделей в естественных науках
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)
		<i>Очная форма</i>				
		Аудиторная работа		Контр оль	Самостоя тельная работа	
Лекции	Семинары (Практическ ие занятия)					
1	Асимптотический анализ. Рекуррентные соотношения	2	2		11	Домашнее задание
2	Бинарный поиск. Префиксные суммы	2	2		11	Домашнее задание
3	Задача сортировки, Теорема о сортировках. Алгоритм сортировки слиянием	2	2		11	Домашнее задание
4	Комбинаторика: перестановки, биномиальные коэффициенты	2	2		11	Домашнее задание Тест
5	Амортизационный анализ: метод потенциалов и метод бухгалтерского учета	2	2		11	Домашнее задание
6	Понятие вероятности. Дискретная теория вероятности. Условная вероятность	2	2		11	Домашнее задание
7	Структура данных бинарная пирамида. Алгоритм пирамидальной сортировки	2	2		11	Домашнее задание Тест
8	Списки: односвязные и двусвязные. Последовательные контейнеры-адаптеры: стек, очередь, двусторонняя очередь	2	2		11	Домашнее задание
9	Случайные величины. Характеристики случайных величин: математическое ожидание и дисперсия. Основные дискретные распределения: биномиальное, геометрическое, пуассоновское	2	2		11	Домашнее задание
10	Рандомизированные алгоритмы. Вероятностная сложность. Алгоритмы быстрой сортировки и поиска порядковой статистики	2	2		11	Домашнее задание Тест

11	Деревья поиска: базовые операции. Красно-черное дерево: теорема о высоте, операция вставки	2	2		11	Домашнее задание
12	Декартово дерево поиска по явному и неявному ключам	2	2		11	Домашнее задание
13	Понятие хеш-функции. Универсальные и k-независимые семейства хеш-функций. Ассоциативные контейнеры	2	2		11	Домашнее задание
14	Гипотеза простого равномерного хеширования. Анализ разрешения коллизий методом цепочек. Алгоритм FKS для статической хеш-таблицы	2	2		11	Домашнее задание
15	Разрешение коллизий методом открытой адресации. Хеширование кукушкой	2	2		10	Домашнее задание
	<i>Зачет с оценкой</i>			4		
	Итого:	30	30	4	164	
Объем дисциплины (модуля) (в ак. ч.)		228				
Объем дисциплины (модуля) (в зач. ед.)		6				

4. Содержание дисциплины (модуля)

№п/п	Наименование раздела дисциплины (модуля)	Содержание дисциплины (модуля) по темам
1	Асимптотический анализ. Рекуррентные соотношения	O-большое, \Omega, \Theta. Асимптотика. Рекуррентные соотношения. *Мастер теорема.
2	Бинарный поиск. Префиксные суммы	Базовые алгоритмы: бинарный поиск, префиксные суммы и прочее
3	Задача сортировки, Теорема о сортировках. Алгоритм сортировки слиянием	Сортировки вплоть до MergeSort.
4	Комбинаторика: перестановки, биномиальные коэффициенты	Основы комбинаторики
5	Амортизационный анализ: метод потенциалов и метод бухгалтерского учета	Амортизационный анализ, асимптотика динамического массива
6	Понятие вероятности. Дискретная теория вероятности. Условная вероятность	Дискретный теорвер
7	Структура данных бинарная пирамида. Алгоритм пирамидальной сортировки	Куча, HeapSort. Построение кучи за линейное время. Алгоритм пирамидальной сортировки
8	Списки: односвязные и двусвязные. Последовательные контейнеры-адаптеры: стек, очередь, двусторонняя очередь	Связные списки. Последовательные контейнеры-адаптеры: стек, очередь, двусторонняя очередь

9	Случайные величины. Характеристики случайных величин: математическое ожидание и дисперсия. Основные дискретные распределения: биномиальное, геометрическое, пуассоновское	Матожидание и дисперсия. *Неравенство Маркова. Основные дискретные распределения: биномиальное, геометрическое, пуассоновское
10	Рандомизированные алгоритмы. Вероятностная сложность. Алгоритмы быстрой сортировки и поиска порядковой статистики	Вероятностная сложность. Quicksort, quick select. Алгоритмы быстрой сортировки и поиска порядковой статистики
11	Деревья поиска: базовые операции. Красно-черное дерево: теорема о высоте, операция вставки	Деревья поиска. Красно-черное дерево: теорема о высоте, операция вставки
12	Декартово дерево поиска по явному и неявному ключам	Деревья поиска (декартово дерево + неявный ключ)
13	Понятие хеш-функции. Универсальные и k-независимые семейства хеш-функций. Ассоциативные контейнеры	Хеширование. Универсальные и k-независимые семейства хеш-функций. Ассоциативные контейнеры
14	Гипотеза простого равномерного хеширования. Анализ разрешения коллизий методом цепочек. Алгоритм FKS для статической хеш-таблицы	Хеш-таблицы методом цепочек. Анализ разрешения коллизий методом цепочек. Алгоритм FKS для статической хеш-таблицы
15	Разрешение коллизий методом открытой адресации. Хеширование кукушкой	Хеш-таблицы часть 2. Разрешение коллизий методом открытой адресации. Хеширование кукушкой

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. Кубенский, А. А. Функциональное программирование : учебник и практикум для вузов / А. А. Кубенский. — Москва : Издательство Юрайт, 2025. — 348 с. — (Высшее образование). — ISBN 978-5-9916-9242-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561074>.

Дополнительная литература:

1. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560815>.

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;
- 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 к 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	Катастрофы, стихийные бедствия, аварии, эпидемии. Солнечная и геомагнитная активность. /ежедневный обзор	http://www.disasters.chat.ru
2	Каталог по безопасности жизнедеятельности	http://www.eun.chat.ru
3	Научная электронная библиотека elibrary.ru библиотека	https://elibrary.ru/defaultx.asp
4	База данных для IT-специалистов	https://habr.com
5	База данных ScienceDirect	https://www.sciencedirect.com
6	Официальный сайт Министерства науки и высшего образования Российской Федерации	https://minobrnauki.gov.ru/
7	Федеральный портал «Российское образование»	https://www.edu.ru/
8	Информационная система "Единое окно доступа к образовательным ресурсам"	http://window.edu.ru/
9	Единая коллекция цифровых образовательных ресурсов	http://school-collection.edu.ru/
10	Федеральный центр информационно - образовательных ресурсов	http://fcior.edu.ru/
11	Сайт различных плагинов	https://maven.apache.org/plugins/
12	Maven central repository - хранилище библиотек и фреймворков	https://mvnrepository.com/repos/central

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
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 Java» в рамках текущего контроля успеваемости используются такие виды учебной работы, как лекции, семинары, домашние задания, тест, 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 заданий 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.

Тест – особая форма проверки знаний. Проводится после освоения одной или нескольких тем и свидетельствует о качестве понимания основных понятий изучаемого материала. Тестовые задания составлены к ключевым понятиям, основным разделам, важным терминологическим категориям изучаемой дисциплины (модуля).

Для подготовки к тесту необходимо знать терминологический аппарат дисциплины (модуля), понимать смысл научных категорий и уметь их использовать в профессиональной лексике. Владение понятийным аппаратом, включённым в тестовые задания, позволяет преподавателю быстро проверить уровень понимания студентами важных методологических категорий.

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 Java».

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

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, 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	Зачтено	
8	Отлично	Зачтено	
7	Хорошо	Зачтено	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,
6	Хорошо	Зачтено	

Десятибалльная оценка	Пятибалльная оценка	Оценка за зачет	Общая характеристика результата обучения по дисциплине (модулю)
			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.
5	Удовлетворительно	Зачтено	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	Зачтено	
3	Не сдан	Не зачтено	Студент не овладел обязательным минимумом знаний по предмету и не может ответить на вопросы, даже если преподаватель задает дополнительные наводящие вопросы.
2	Не сдан	Не зачтено	
1	Не сдан	Не зачтено	

Дисциплина (модуль) «Алгоритмы Java» оценивается следующим образом:

Активность	Вес	Описание
Домашние задания	70%	Оцениваются по критериям. Можно набрать максимум 10 баллов за каждое из заданий.
Тест	30%	Ответы на вопросы по изученным темам

Формула расчёта итоговой оценки по дисциплине (модулю) «Алгоритмы Java»:
« $0,7 \times \text{среднее за домашние задания} + 0,3 \times \text{тест}$ ».

Текущий контроль успеваемости обучающихся по дисциплине (модулю)

Примерные домашние задания

Домашнее задание: Случайные величины и их характеристики

1. Определите математическое ожидание и дисперсию для биномиального распределения с параметрами $n=10$ и $p=0.5$.

2. Рассчитайте математическое ожидание и дисперсию для геометрического распределения с параметром $p=0.3$.

3. Найдите математическое ожидание и дисперсию пуассоновского распределения с параметром $\lambda=4$.

4. Примените неравенство Маркова к случайной величине XX с математическим ожиданием $E(X)=5$ и найдите верхнюю границу вероятности $P(X \geq 10)$.

5. Приведите примеры реальных задач, где можно применить биномиальное, геометрическое и пуассоновское распределения.

Домашнее задание: Рандомизированные алгоритмы

1. Определите понятие вероятностной сложности и приведите пример алгоритма с вероятностной сложностью.

2. Опишите алгоритм быстрой сортировки (Quicksort) и его среднюю временную сложность.

3. Приведите пример использования алгоритма поиска порядковой статистики (Quickselect) и объясните, как он работает.

4. Рассчитайте вероятностную сложность алгоритма Quickselect для поиска медианы в массиве из 7 элементов.

5. Объясните, как выбор опорного элемента влияет на производительность алгоритма Quicksort.

Домашнее задание: Деревья поиска

1. Опишите базовые операции, которые можно выполнять с деревьями поиска (вставка, удаление, поиск).

2. Объясните теорему о высоте красно-черного дерева и приведите пример дерева, соответствующего этой теореме.

3. Опишите процесс вставки нового узла в красно-черное дерево и укажите, какие операции могут потребоваться для поддержания свойств дерева.

4. Приведите пример, как изменится структура красно-черного дерева после вставки нескольких элементов.

5. Рассчитайте высоту красно-черного дерева, если в нем содержится 15 узлов.

Домашнее задание: Хеширование и анализ коллизий

1. Определите понятие хеш-функции и приведите примеры универсальных хеш-функций.

2. Объясните, что такое k -независимые семейства хеш-функций и как они используются для уменьшения коллизий.

3. Опишите методы разрешения коллизий в хеш-таблицах и приведите пример метода цепочек.

4. Объясните гипотезу простого равномерного хеширования и как она влияет на производительность хеш-таблицы.

5. Опишите алгоритм FKS для статической хеш-таблицы и объясните, как он работает.

Примерные задания к тестам

Тест: Структура данных бинарная пирамида и алгоритм пирамидальной сортировки

1. Что такое бинарная пирамида (heap)?

- a) Дерево поиска
- b) Полное бинарное дерево, удовлетворяющее свойству пирамиды
- c) Связный список
- d) Хеш-таблица

2. Какое свойство выполняет бинарная пирамида?

- a) Значение в родительском узле всегда меньше значений в дочерних
- b) Значение в родительском узле всегда больше или равно значений в дочерних
- c) Узлы упорядочены по алфавиту
- d) Нет ограничений на значения узлов

3. Какова временная сложность операции вставки в бинарную пирамиду?
- a) $O(1)$
 - b) $O(\log N)$
 - c) $O(N)$
 - d) $O(N \log N)$
4. Что делает операция "просеивания вниз" (heapify) в пирамиде?
- a) Вставляет новый элемент
 - b) Восстанавливает свойство пирамиды после удаления корня
 - c) Удаляет последний элемент
 - d) Сортирует массив
5. Что из перечисленного является основным шагом алгоритма пирамидальной сортировки?
- a) Построение пирамиды из массива
 - b) Последовательное удаление максимума с восстановлением пирамиды
 - c) Сортировка вставками
 - d) Быстрая сортировка
6. Какова общая временная сложность пирамидальной сортировки?
- a) $O(N)$
 - b) $O(N \log N)$
 - c) $O(\log N)$
 - d) $O(N^2)$
7. В каком порядке элементы извлекаются из бинарной пирамиды при сортировке?
- a) От минимального к максимальному
 - b) От максимального к минимальному
 - c) В случайном порядке
 - d) По алфавиту
8. Что происходит с корнем пирамиды после его удаления?
- a) Корень заменяется последним элементом, затем происходит просеивание вниз
 - b) Корень просто удаляется без замены
 - c) Корень заменяется случайным элементом
 - d) Корень дублируется
9. Как можно представить бинарную пирамиду в памяти?
- a) В виде связного списка
 - b) В виде массива
 - c) В виде хеш-таблицы
 - d) В виде графа
10. Какой тип пирамиды используется для сортировки по возрастанию?
- a) Максимальная пирамида
 - b) Минимальная пирамида
 - c) Двухнаправленная пирамида
 - d) Не используется пирамида

Тест: Рандомизированные алгоритмы, вероятностная сложность, Quicksort и Quickselect

1. Что означает термин "рандомизированный алгоритм"?
- a) Алгоритм, использующий случайные числа для принятия решений

- b) Алгоритм, работающий только на случайных данных
- c) Алгоритм, который всегда выдает случайный результат
- d) Алгоритм с детерминированным поведением

2. Какова средняя временная сложность алгоритма Quicksort?

- a) $O(N)$
- b) $O(N \log N)$
- c) $O(N^2)$
- d) $O(\log N)$

3. Что такое "поиск порядковой статистики" (Quickselect)?

- a) Поиск максимального элемента
- b) Поиск k-го по величине элемента в массиве
- c) Сортировка массива
- d) Поиск элемента по индексу

4. Какой метод используется для выбора опорного элемента в Quicksort?

- a) Первый элемент
- b) Последний элемент
- c) Случайный элемент
- d) Средний элемент

5. Какова худшая временная сложность Quicksort?

- a) $O(n)$
- b) $O(N \log N)$
- c) $O(N^2)$
- d) $O(\log N)$

6. Что такое вероятностная сложность алгоритма?

- a) Время работы в худшем случае
- b) Среднее время работы по вероятностному распределению входных данных и случайных чисел
- c) Время работы в лучшем случае
- d) Количество операций сравнения

7. Какая из следующих характеристик относится к Quickselect?

- a) Сортирует весь массив
- b) Работает за $O(N)$ в среднем
- c) Работает за $O(N^2)$ в среднем
- d) Использует дополнительную память $O(N)$

8. Что происходит на этапе "разделения" (partition) в Quicksort?

- a) Массив разбивается на две части относительно опорного элемента
- b) Массив сортируется полностью
- c) Выбирается максимальный элемент
- d) Элементы перемешиваются случайным образом

9. Как влияет выбор опорного элемента на производительность Quicksort?

- a) Не влияет
- b) Чем ближе опорный элемент к медиане, тем лучше производительность
- c) Чем дальше от медианы, тем лучше производительность
- d) Выбор опорного элемента влияет только на память

10. Какой из следующих алгоритмов является рандомизированным?

- a) Сортировка вставками
- b) Quicksort с выбором случайного опорного элемента
- c) Сортировка слиянием
- d) Сортировка пузырьком

Задания для промежуточной аттестации по дисциплине (модулю)

№ п/п	Задание	Ответ	Компетенция
1.	Выберете из нижеперечисленных контейнеров ассоциативные: А. Deque Б. PriorityQueue В. HashMap Г. ArrayList	В	ОПК-3
2.	Какой из нижеперечисленных контейнеров позволяет проверять наличие элемента в нем за константное в среднем время: А. HashSet Б. TreeSet В. ArrayList Г. PriorityQueue	А	ОПК-3
3.	Выберите из нижеперечисленных алгоритмов сортировки тот, который работает за $O(N \log N)$ в худшем случае А. HeapSort Б. BubbleSort В. InsertionSort	А	ПК-6
4.	Рассмотрим задачу проверки наличия элемента в неизменяемом наборе элементов. Алгоритм А: отсортируем набор с помощью сортировки слиянием, запрос проверки наличия элемента в наборе осуществим бинарным поиском Алгоритм Б: построим TreeSet на элементах, запрос проверки осуществим стандартными методами TreeSet Алгоритм В: построим HashSet на элементах, запрос проверки осуществим стандартными методами HashSet. Для каждого из алгоритмов выберите один из трех вариантов времени подготовки данных в O -нотации (ответ запишите в виде последовательности чисел, где 121 означает, что для алгоритма А выбран первый вариант, для алгоритма Б - второй, для алгоритма В - первый). 1. $O(N)$ 2. $O(N)$ в среднем 3. $O(N \log N)$	332	ОПК-3
5.	Рассмотрим задачу проверки наличия элемента в неизменяемом наборе элементов. Алгоритм: построим TreeSet на элементах, запрос проверки осуществим стандартными запросами TreeSet Какая дополнительная память потребуется, если N - число элементов в контейнере? (ответ запишите в O -нотации).	$O(N)$	ОПК-3

6.	Рассмотрим семейство ассоциативных контейнеров, построенных на основе хеш-таблиц. Какое матожидание времени проверки наличия элемента при использовании метода цепочек для разрешения коллизий с универсальным семейством хеш-функций, если N - число элементов в контейнере. <i>Ответ запишите в O-нотации</i>	$O(1)/O(1)$ в среднем	ПК-6
7.	Рассмотрим семейство ассоциативных контейнеров, построенных на основе самобалансирующихся деревьев поиска. Какое время проверки наличия элемента, если N - число элементов в контейнере. <i>Ответ запишите в O-нотации</i>	$O(\log N)$	ПК-6
8.	Какой высоты в худшем случае может достичь бинарное дерево поиска без балансировки? <i>Ответ запишите в O-нотации</i>	$O(N)$	ПК-6
9.	Рассмотрим процедуру Partition, используемую в алгоритме QuickSort. Укажите время работы и потребляемую дополнительную память этой процедуры. <i>Ответ запишите в O-нотации через запятую, сначала время, затем дополнительная память без пробелов</i>	$O(N), O(1)$	ПК-6
10.	Рассмотрим рекуррентное соотношение на время работы некоторого алгоритма: $T(N) = 2 * T(N / 2) + O(N)$ В качестве базы рекурсии использовать $T(k) = 0$ при $k < 1$. Решите рекуррентное соотношение и укажите асимптотику роста $T(N)$. <i>Ответ запишите в O-нотации</i>	$O(N \log N)$	ПК-6
11.	Рассмотрим рекуррентное соотношение на время работы некоторого алгоритма: $T(N) = T(N / 2) + O(1)$ В качестве базы рекурсии использовать $T(k) = 0$ при $k < 1$. Решите рекуррентное соотношение и укажите асимптотику роста $T(N)$. <i>Ответ запишите в O-нотации</i>	$O(\log N)$	ПК-6
12.	Рассмотрим некоторый алгоритм, который требует структуры данных с следующим интерфейсом: - Добавить элемент в структуру - Извлечь/удалить минимальный элемент Все запросы должны выполняться за $O(\log N)$, где N - число элементов в структуре. Предъявите пример такой структуры из стандартной библиотеки Java.	PriorityQueue/TreeSet / priorityqueue/treeset / Priorityqueue/Treeset	ПК-6