Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 14.07.2026 16:42:53
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Московский политехнический университет»

УТВЕРЖДАЮ
Декан факультета
«Информационных технологий»
_____ / Д.Г. Демидов /
«26» февраля 2026 г.

Рабочая программа дисциплины
«БОЛЬШИЕ ДАННЫЕ»

Направление подготовки:
09.03.01 Информатика и вычислительная техника

Образовательная программа (профиль):
«Разработка инженерного программного обеспечения»

Год начала обучения:
2026

Уровень образования:
бакалавриат

Квалификация (степень) выпускника:
Бакалавр

Форма обучения:
очная

Москва, 2026

Разработчик(и):

к.ф-м.н, доцент  / А.В. Филимонов/

Согласовано:
Заведующий кафедрой «Прикладная информатика»,
к.э.н, доцент  С.В. Суворов/

Содержание

1	Цели, задачи и планируемые результаты обучения по дисциплине	4
2	Место дисциплины в структуре образовательной программы	5
3	Структура и содержание дисциплины	5
3.1	Виды учебной работы и трудоемкость	6
3.2	Тематический план изучения дисциплины	6
3.3	Содержание дисциплины	6
4	Учебно-методическое и информационное обеспечение	7
4.1	Основная литература	7
4.2	Дополнительная литература	7
4.3	Электронные образовательные ресурсы	7
5	Материально-техническое обеспечение	8
6	Методические рекомендации	8
6.1	Методические рекомендации для преподавателя по организации обучения	8
6.2	Методические указания для обучающихся по освоению дисциплины	8
7	Фонд оценочных средств	9
7.1	Методы контроля и оценивания результатов обучения	9
7.2	Шкала и критерии оценивания результатов обучения	9
7.3	Оценочные средства	16

1. Цели, задачи и планируемые результаты обучения по дисциплине

К основной цели освоения дисциплины «Большие данные» можно отнести формирование у студентов профессиональной компетенции в области разработки и использования систем обработки и анализа больших массивов данных, изучению теоретических основ построения и функционирования подобных систем.

К основным задачам дисциплины «Большие данные» относятся:

- Разработка методик автоматизации обработки больших данных.
- Изучить теоретические основы построения систем обработки и анализа больших данных.
- Изучить основные языковые и визуальные способы организации, отображения и манипулирования данными под управлением систем обработки и анализа больших данных.
- Изучить теоретические основы построения и функционирования подобных систем.

Обучение по дисциплине «Большие данные» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	**Индикаторы достижения компетенции**
ПК-1. Способен разрабатывать требования и проектировать программное обеспечение.	ИПК-1.1. Знает: Возможности существующей программно-технической архитектуры Возможности современных и перспективных средств разработки программных продуктов, технических средств Методологии разработки программного обеспечения и технологии программирования Методологии и технологии проектирования и использования баз данных Языки формализации функциональных спецификаций Методы и приемы формализации задач Методы и средства проектирования программного обеспечения Методы и средства проектирования программных интерфейсов Методы и средства проектирования баз данных Принципы построения и виды архитектуры компьютерного программного обеспечения; Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения *ИПК-1.2 Умеет:* Проводить сбор и систематизацию требований к компьютерному программному обеспечению

	проводить оценку и обоснование рекомендуемых решений; проводить оценку и обоснование рекомендуемых решений; использовать существующие типовые решения и шаблоны проектирования компьютерного программного обеспечения; применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов; применять существующие стандарты для разработки технической документации на компьютерное программное обеспечение. *ИПК-1.3 Владеет:* Инструментами и технологиями разработки требований и проектирования программного обеспечения; инструментами и технологиями разработки программного кода.

2. Место дисциплины в структуре образовательной программы

Дисциплина «Большие данные» относится к числу элективных дисциплин, основной профессиональной образовательной программы.

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Теория вероятностей и математическая статистика;
- Основы программирования;
- Базы данных;
- Веб-разработка;
- Тестирование программного обеспечения

3. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единиц, т.е. **144** академических часа, из них 72 часа – самостоятельная работа студентов. Все они осваиваются обучающимися в седьмом семестре.

Виды учебных занятий по дисциплине: лекции – 1 часа в неделю (всего 18 часов), лабораторные занятия – **2** час в неделю (всего 54 часов). Формы контроля – экзамен.

3.1. Виды учебной работы и трудоемкость

№ п/п	**Вид учебной работы**	**Количество часов**	**Семестры**	
			Семестр	**Неделя семестра**
1	**Аудиторные занятия**	**72**	7	1-15
	В том числе:			
1.1	Лекции	18	7	1-15
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	54	7	1-15
2	**Самостоятельная работа**	**72**	7	1-15

3	**Промежуточная аттестация**		4	16-17
	Экзамен			
	Итого:	**144**		

3.2. Тематический план изучения дисциплины

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Математические основы анализа алгоритмов	35	4		13		18
2	Проекты «Выпуклая оболочка» и «Изображение проекции полиэдра»	37	5		14		18
3	Реализация структур данных	35	4		13		18
4	Сортировки	37	5		14		18
Итого		**144**	**18**		**54**		**72**

3.3 Содержание дисциплины

Математические основы анализа алгоритмов

Асимптотические обозначения и сравнение скоростей роста функций. Стандартные функции и обозначения. Суммирование и оценки сумм. Применение различных математических методов для анализа сложности алгоритмов. Рекуррентные соотношения. Метод подстановки. Метод итераций. Общая теорема о рекуррентных соотношениях и примеры её применения.

Проекты «Выпуклая оболочка» и «Изображение проекции полиэдра»

Графический интерфейс проекта "Выпуклая оболочка". Модули и потоки в языке Ruby. Модификация эталонного кода проекта "Выпуклая оболочка" в соответствии с индивидуальными заданиями. Проект "Изображение проекции полиэдра". Задание полиэдра. Изображение проекции полиэдра без удаления невидимых линий. Тени и просветы. Одномерные координаты. Тень от грани. Пересечение отрезка с полупространством. Оптимизация эталонного кода проекта "Изображение проекции полиэдра": отделение фазы вычислений от фазы изображения, удаление дубликатов рёбер, предкомпиляция граней, учёт особенностей взаимного расположения рёбер и граней, гнездование граней. Модификация эталонного кода проекта "Изображение проекции полиэдра" в соответствии с индивидуальными заданиями.

Реализация структур данных

Непрерывные и ссылочные реализации. Непрерывная реализация ограниченного стека и очереди, двух и трёх стеков, ограниченных в совокупности, на базе вектора. Оценки эффективности. Ссылочные реализации списков. Различные способы реализации динамического множества (линейный и последовательный поиск, таблица с прямой адресацией и битовая реализация). Хеш-таблицы и хеш-функции. Оценки эффективности. Деревья и основные определения, связанные с ними. Представление корневых деревьев (двоичных и сильно ветвящихся). Двоичные деревья поиска и операции над ними. Построение идеально сбалансированного дерева с заданным числом узлов и его обход (три варианта). Оценка высоты случайного двоичного дерева поиска. Анализ поиска с включением по дереву и выводы из него. Представление об идеально сбалансированных, АВЛ и красно-чёрных двоичных деревьях поиска. Их сравнительные характеристики и области применения.

Сортировки

Внутренние сортировки и их основные характеристики. Сортировка простыми включениями и анализ её сложности. Сортировки простым выбором и простым обменом. Простейшие модификации пузырьковой сортировки. Анализ сложности. Сортировка слиянием. Оценки её ёмкостной и временной эффективности с помощью теоремы о рекуррентных оценках. Идея пирамидальной сортировки (сортировки с помощью кучи). Реализация пирамиды на массиве. Итоговый алгоритм и анализ эффективности пирамидальной сортировки. Быстрая сортировка. Рекурсивная и итерационная версии. Оценка глубины стека. Сравнение быстрой и пирамидальной сортировок. Внешние сортировки. Простое, естественное и многопутевое слияние. Многофазная сортировка и начальное распределение серий. Оценки эффективности и методы повышение скорости работы внешних сортировок. Теорема о нижней границе числа обменов для произвольного алгоритма внутренней сортировки. Примеры альтернативных более быстрых алгоритмов (сортировка подсчётом и цифровая сортировка). Применение теоремы о рекуррентных оценках для получения оценок временной эффективности пирамидальной и быстрой (в худшем, лучшем и среднем случае) сортировок.

4. Учебно-методическое и информационное обеспечение

4.1. Основная литература

1. Мейер, Б. Инструменты, алгоритмы и структуры данных : учебное пособие / Б. Мейер. — 2-е изд. — Москва : ИНТУИТ, 2016. — 542 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/100603 (дата обращения: 25.09.2023). — Режим доступа: для авториз. пользователей.

2. Сыромятников, В. П. Структуры и алгоритмы обработки данных: Практикум : учебное пособие / В. П. Сыромятников. — Москва : РТУ МИРЭА, 2020. — 244 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/163915 (дата обращения: 25.09.2023). — Режим доступа: для авториз. пользователей.

4.2. Дополнительная литература

1. Апанасевич, С. А. Структуры и алгоритмы обработки данных. Линейные структуры : учебное пособие / С. А. Апанасевич. — Санкт-Петербург : Лань, 2022. — 136 с. — ISBN 978-5-8114-3366-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/206261 (дата обращения: 25.09.2023). — Режим доступа: для авториз. пользователей.
2. Макшанов, А. В. Большие данные. Big Data / А. В. Макшанов, А. Е. Журавлев, Л. Н. Тындыкарь. — 3-е изд., стер. — Санкт-Петербург : Лань, 2023. — 188 с. — ISBN 978-5-507-46866-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/322664 (дата обращения: 25.09.2023). — Режим доступа: для авториз. пользователей.
3. Нурматова, Е. В. Управление большими базами данных и высоконагруженными системами : учебное пособие / Е. В. Нурматова, Р. Ф. Халабия, Л. В. Бунина. — Москва : РТУ МИРЭА, 2019. — 120 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/171496 (дата обращения: 25.09.2023). — Режим доступа: для авториз. пользователей.

4.3 Электронные образовательные ресурсы

Электронный образовательный ресурс Технологии обработки больших данных: https://online.mospolytech.ru/course/view.php?id=12091

5. Материально-техническое обеспечение

5.1 Требования к оборудованию и помещению для занятий

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены современным компьютером с подключенным к нему проектором на настенный экран, или иным аналогичным по функциональному назначению оборудованием.

5.2 Требования к программному обеспечению

Для выполнения лабораторных работ и самостоятельной работы необходимо следующее программное обеспечение:

1. Microsoft Windows.
2. Notepad++.
3. Microsoft office.
4. Веб-браузер, Chrome.

6. Методические рекомендации

6.1. Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

3. При организации и проведения экзаменов в практико-ориентированной форме следует использовать утвержденные кафедрой Методические рекомендации.

6.2. Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи с учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия, лабораторные работы.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к проведению и обрабатывают результаты лабораторных работ, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;
- контроль со стороны преподавателей (текущий и промежуточный).

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

7. Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- проверка домашних заданий;
- проверка готовности студентов к проведению лабораторных работ;
- проверка выполненных лабораторных работ;
- проверка курсовых работ;
- проведение экзамена.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	Критерии оценивания			
	2	3	4	5
ПК-1. Способен разрабатывать требования и проектировать программное обеспечение.				
ПК-1.1. Знать: Возможности существующей программно-технической архитектуры. Возможности современных и перспективных средств разработки программных продуктов, технических средств. Методологии разработки программного обеспечения и технологии программирования. Методологии и технологии проектирования и использования баз данных. Языки формализации	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, возможности существующей программно-технической архитектуры. Возможности современных и перспективных средств разработки программных продуктов, технических средств. Методологии разработки программного обеспечения и технологии	Обучающийся демонстрирует неполное соответствие следующих знаний, Возможности существующей программно-технической архитектуры. возможности современных и перспективных средств разработки программных продуктов, технических средств. Методологии разработки программного обеспечения и технологии программирования. Методологии и технологии	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины возможности существующей программно-технической архитектуры. Возможности современных и перспективных средств разработки программных продуктов, технических средств. Методологии разработки программного обеспечения и	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины возможности существующей программно-технической архитектуры. Возможности современных и перспективных средств разработки программных продуктов, технических средств. Методологии разработки программного обеспечения и

функциональных спецификаций. Методы и приемы формализации задач. Методы и средства проектирования программного обеспечения. Методы и средства проектирования программных интерфейсов. Методы и средства проектирования баз данных. Принципы построения и виды архитектуры компьютерного программного обеспечения; Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения.	программирования. Методологии и технологии проектирования и использования баз данных. Языки формализации функциональных спецификаций. Методы и приемы формализации задач. Методы и средства проектирования программного обеспечения. Методы и средства проектирования программных интерфейсов. Методы и средства проектирования баз данных. Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения. Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения..	проектирования и использования баз данных. Языки формализации функциональных спецификаций. Методы и приемы формализации задач. Методы и средства проектирования программного обеспечения. Методы и средства проектирования программных интерфейсов. Методы и средства проектирования баз данных. Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения. Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей,	технологии программирования. Методологии и технологии проектирования и использования баз данных. Языки формализации функциональных спецификаций. Методы и приемы формализации задач. Методы и средства проектирования программного обеспечения. Методы и средства проектирования программных интерфейсов. Методы и средства проектирования баз данных. Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения. Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения. Но допускаются незначительные ошибки,	технологии программирования. Методологии и технологии проектирования и использования баз данных. Языки формализации функциональных спецификаций. Методы и приемы формализации задач. Методы и средства проектирования программного обеспечения. Методы и средства проектирования программных интерфейсов. Методы и средства проектирования баз данных. Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения. Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения. Свободно оперирует приобретенными

		обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	неточности, затруднения при аналитических операциях.	знаниями.
ПК-1.2. уметь: Проводить сбор и систематизацию требований к компьютерному программному обеспечению проводить оценку и обоснование рекомендуемых решений; использовать существующие типовые решения и шаблоны проектирования компьютерного программного обеспечения; применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов;	Обучающийся не умеет Проводить сбор и систематизацию требований к компьютерному программному обеспечению проводить оценку и обоснование рекомендуемых решений; использовать существующие типовые решения и шаблоны проектирования компьютерного программного обеспечения; применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов;	Обучающийся частично умеет Проводить сбор и систематизацию требований к компьютерному программному обеспечению проводить оценку и обоснование рекомендуемых решений; использовать существующие типовые решения и шаблоны проектирования компьютерного программного обеспечения; применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов;	Обучающийся в не полном объеме умеет Проводить сбор и систематизацию требований к компьютерному программному обеспечению проводить оценку и обоснование рекомендуемых решений; использовать существующие типовые решения и шаблоны проектирования компьютерного программного обеспечения; применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов;	Обучающийся в полном объеме умеет Проводить сбор и систематизацию требований к компьютерному программному обеспечению проводить оценку и обоснование рекомендуемых решений; использовать существующие типовые решения и шаблоны проектирования компьютерного программного обеспечения; применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов;
ПК-1.3. Владеть: Инструментами и технологиями разработки требований и проектирования программного обеспечения.	Обучающийся не владеет инструментами и технологиями разработки требований и проектирования программного	Обучающийся частично владеет инструментами и технологиями разработки требований и проектирования программного	Обучающийся владеет в не полном объеме инструментами и технологиями разработки требований и проектирования	Обучающийся в полном объеме владеет инструментами и технологиями разработки требований и проектирования

Инструментами и технологиями разработки программного кода.	обеспечения. Инструментами и технологиями разработки программного кода.	обеспечения. Инструментами и технологиями разработки программного кода.	программного обеспечения. Инструментами и технологиями разработки программного кода.	программного обеспечения. Инструментами и технологиями разработки программного кода.

Шкалы оценивания результатов промежуточной аттестации и их описание:

Форма промежуточной аттестации: экзамен

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине, при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо»,
«удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Технологии обработки больших данных» (выполнили практические и лабораторные работы).

Шкала оценивания	**Описание**
Отлично	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент в основном демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены некоторые ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Удовлетворительно	Студент демонстрирует удовлетворительное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются умеренные ошибки, проявляется неполное наличие знаний, умений, навыков по ряду показателей, студент испытывает затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Неудовлетворительно	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.3 Оценочные средства

Примерный перечень вопросов к экзамену по дисциплине «Большие данные»:

1. Определение асимптотического обозначения $f(n)=Theta(g(n)$, его основные свойства (наличие или отсутствие транзитивности, рефлексивности, симметричности). Примеры. Аналогия с отношением равенства на числовых множествах.
2. Формулировка основной теоремы о рекуррентных оценках и иллюстрация применения её для первого из рассматриваемых в ней случаев не менее, чем на трёх различных примерах.
3. Формулировка основной теоремы о рекуррентных оценках и иллюстрация применения её для второго из рассматриваемых в ней случаев не менее, чем на трёх различных примерах.
4. Формулировка основной теоремы о рекуррентных оценках и иллюстрация применения её для третьего из рассматриваемых в ней случаев не менее, чем на трёх различных примерах.
5. Формулировка основной теоремы о рекуррентных оценках. Примеры (не менее трёх) ситуаций, в которых эта теорема не применима.
6. Определение выпуклого множества и выпуклой оболочки. Примеры. Точная постановка задачи, рассматриваемой в эталонной версии проекта "Выпуклая оболочка", и описание идеи её решения.
7. Основные классы, задающие выпуклую оболочку, используемые в проекте "Выпуклая оболочка".
8. Класс R2Point. Применение векторной алгебры и аналитической геометрии при реализации некоторых методов этого класса. Иллюстрация на примерах возможности использования методов экземпляра вместо методов класса и наоборот.
9. Использование графического интерфейса в проекте "Выпуклая оболочка". Преобразование координат, рисование точек, отрезков и многоугольников.
10. Понятие ребра, освещённого из заданной точки. Реализация метода light?. Использование понятия освещённости при реализации метода initialize класса Polygon.
11. Использование идей и фрагментов кода проекта "Выпуклая оболочка" при написании максимально эффективной программы, выясняющей, лежит ли точка плоскости M(x,y) строго внутри некоторого треугольника, заданного координатами его вершин.
12. Определение полиэдра и способ его задания. Точная постановка задачи, рассматриваемой в эталонной версии проекта "Изображение проекции полиэдра", и описание её решения без удаления невидимых линий.
13. Основные классы, задающие полиэдр, используемые в проекте "Изображение проекции полиэдра". Класс R3 и применение векторной алгебры и аналитической геометрии при реализации некоторых методов этого класса.
14. Идея решения задачи удаления невидимых линий в проекте "Изображение проекции полиэдра". Тени, просветы и одномерные координаты. Реализация методов класса Segment.
15. Учёт тени на ребре от одной грани. Призма "Тень". Описание методов класса Edge.
16. Геометрические основы решения задачи нахождения пересечения отрезка с полупространством. Реализация метода cross класса Edge.
17. Различные способы оптимизации проекта "Изображение проекции полиэдра". Сравнение их относительной эффективности.
18. На занятиях была рассмотрена непрерывная реализация стека на базе ограниченного

вектора. Какие методы этой реализации данного контейнера и как именно нужно модифицировать, чтобы при работе с целыми числами контейнер "терял нули": добавить сколь угодно много нулей можно было бы всегда, а взять ни одного нуля не получалось бы никогда?

19. Опишите несколькими предложениями идею непрерывной реализации контейнера "три стека, ограниченные в совокупности" и реализуйте методы инициализации (конструктор) и извлечения элемента из контейнера.
20. Опишите несколькими предложениями идею непрерывной реализации контейнера "ограниченное множество (линейный поиск)" и реализуйте методы инициализации (конструктор) и добавления элемента в контейнер.
21. На занятиях была рассмотрена непрерывная реализация очереди на базе ограниченного вектора. Какие методы этой реализации данного контейнера и как именно нужно модифицировать, чтобы при работе с целыми числами контейнер "работал сумматором": при взятии элемента возвращалась бы сумма всех элементов, содержавшихся в контейнере до извлечения этого элемента?
22. Опишите несколькими предложениями идею непрерывной реализации контейнера "три стека, ограниченные в совокупности" и реализуйте методы инициализации (конструктор) и добавления элемента в контейнер.
23. Опишите несколькими предложениями идею непрерывной реализации контейнера "ограниченное множество (двоичный поиск)" и реализуйте методы инициализации (конструктор) и удаления элемента из контейнера.
24. На занятиях была рассмотрена непрерывная реализация стека на базе ограниченного вектора. Какие методы этой реализации данного контейнера и как именно нужно модифицировать, чтобы при работе с целыми числами контейнер "удваивал нули": положив один нуль, возьмёшь два; положив два нуля, извлечёшь четыре ...
25. Опишите несколькими предложениями идею непрерывной реализации контейнера "два стека, ограниченные в совокупности" и реализуйте методы инициализации (конструктор) и извлечения элемента из контейнера.
26. Опишите несколькими предложениями идею непрерывной реализации контейнера "ограниченное множество (двоичный поиск)" и реализуйте методы инициализации (конструктор) и добавления элемента в контейнер.
27. На занятиях была рассмотрена непрерывная реализация очереди на базе ограниченного вектора. Какие методы этой реализации данного контейнера и как именно нужно модифицировать, чтобы при работе с целыми числами контейнер "ополовинивал двойки": положив одну двойку, взять её никогда не сможешь; положив две двойки подряд, извлечёшь потом только одну двойку; положив четыре двойки подряд, извлечёшь две двойки...
28. Опишите несколькими предложениями идею непрерывной реализации контейнера "два стека, ограниченные в совокупности" и реализуйте методы инициализации (конструктор) и добавления элемента в контейнер.
29. Опишите несколькими предложениями идею непрерывной реализации контейнера "ограниченное множество (двоичный поиск)" и реализуйте методы инициализации (конструктор) и удаления элемента из контейнера.
30. На занятиях была рассмотрена непрерывная реализация стека на базе ограниченного вектора. Какие методы этой реализации данного контейнера и как именно нужно модифицировать, чтобы при работе с целыми числами контейнер "удваивал единицы и ополовинивал двойки": положив единицу, извлечёшь двойку; положив двойку, извлечёшь единицу; положив тройку, извлечёшь тройку; положив четвёрку, извлечёшь четвёрку...
31. Внутренние сортировки. Их основные характеристики. Основные рассмотренные в курсе алгоритмы внутренних сортировок.
32. Сортировка простыми включениями (реализация). Анализ сложности (нижние и верхние оценки для числа сравнений и пересылок).
33. Сортировка простым выбором (реализация). Анализ сложности (нижние и верхние

оценки для числа сравнений и пересылок).

34. Сортировка простым обменом (реализация). Анализ сложности (нижние и верхние оценки для числа сравнений и пересылок).
35. Сортировка слиянием. Её основные характеристики (включая временную и ёмкостную сложность в лучшем и худшем случаях).
36. Сортировка подсчётом. Её основные характеристики (включая временную и ёмкостную сложность в лучшем и худшем случаях).
37. Идея быстрой сортировки. Разделение массива и рекурсивная версия быстрой сортировки (фрагменты реализации). Оценки эффективности быстрой сортировки в лучшем и худшем случаях (с помощью рекуррентного соотношения и теоремы о рекуррентных оценках).
38. Идея пирамидальной сортировки. Определение пирамиды (сортирующего дерева) на массиве. Оценка эффективности пирамидальной сортировки в худшем случае.
39. Внешние сортировки. Основные рассмотренные в курсе алгоритмы внешних сортировок (с изложением идей реализации).
40. Оптимизация алгоритмов внешних сортировок за счёт использования оперативной памяти и оценка получаемого выигрыша в скорости работы.
41. Методы sort и sort_by класса Enumerable стандартной библиотеки языка Ruby. Примеры их использования и рекомендации по их применению.
42. Расположите следующие функции в порядке увеличения скорости их роста, отметив среди них Θ-эквивалентные: $f(n) = 1$, $g(n) = n^{1/\log n}$, $h(n) = \log n$.
43. Расположите следующие функции в порядке увеличения скорости их роста, отметив среди них Θ-эквивалентные: $f(n) = \log n$, $g(n)
= \log(2n)$, $h(n) = 2\log(2n)$.
44. Воспользовавшись теоремой о рекуррентных соотношениях, найдите асимптотику функции $T(n)$, для которой $T(n) = 4T(n/2) + (2n^2+1)$.
45. Воспользовавшись теоремой о рекуррентных соотношениях, найдите асимптотику функции $T(n)$, для которой $T(n) = 9T(n/3) + 3n^2$.
46. Воспользовавшись теоремой о рекуррентных соотношениях, найдите асимптотику функции $T(n)$, для которой $T(n) = 2T(n/2) + n^3$.
47. Найдите точные значения максимального количества сравнений $C(n)$ и пересылок $M(n)$ элементов сортируемого массива длины n для следующей Ruby-программы, реализующей один из простейших алгоритмов сортировок. Поясните, как был найден ответ. Программа прилагается (6 вариантов).
48. Реализуйте метод, имеющий два аргумента (указатель на начало двусвязного списка и значение ключа), который ищет в заданном списке элемент с указанным ключом, возвращая его (или nil при отсутствии такого ключа в списке). Дайте оценку эффективности реализованного метода.
49. Реализуйте метод, имеющий два аргумента (указатель на начало односвязного списка и значение ключа), который удаляет первый из элементов заданного списка, имеющий указанный ключ (список не меняется, если элементов с такими значениями ключа в нём нет). Дайте оценку эффективности реализованного метода.
50. Как реализуется представление двоичного дерева? Дайте определение двоичного дерева поиска. Реализуйте три алгоритма обхода двоичного дерева.
51. Перечислите основные операции с двоичными деревьями поиска. Опишите идеи реализации и дайте оценки эффективности алгоритмов поиска, нахождения минимума и максимума.
52. Перечислите основные операции с двоичными деревьями поиска. Опишите идею реализации и дайте оценку эффективности алгоритма удаления элемента из дерева.
53. С помощью теоремы о рекуррентных оценках оцените максимальную сложность следующих алгоритмов: двоичный поиск элемента в упорядоченном массиве, сортировка слиянием, быстрая сортировка.

54. Теорема о нижней оценке эффективности алгоритмов сортировок, основанных на сравнении элементов. Её доказательство.
55. Теорема об оценке высоты случайного дерева поиска. Её доказательство и следствие из неё о сфере применимости деревьев поиска без управления ростом.