

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор физтех-школы
прикладной математики и
информатики**

А.М. Райгородский

	Рабочая программа дисциплины (модуля)
по дисциплине:	Алгоритмы и структуры данных. Продвинутый поток
по направлению:	Прикладная математика и информатика
профиль подготовки:	Информатика
	Физтех-школа Прикладной Математики и Информатики
	кафедра алгоритмов и технологий программирования
курс:	1
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

1 (осенний) - Экзамен

2 (весенний) - Экзамен

3 (осенний) - Экзамен

Аудиторных часов: 180 всего, в том числе:

лекции: 90 час.

семинары: 90 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 135 час.

Подготовка к экзамену: 90 час.

Всего часов: 405, всего зач. ед.: 9

Количество контрольных работ, заданий: 6

Программу составил: И.Д. Степанов, ассистент

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 01.06.2023

Аннотация

В начале вводятся общие математические обозначения, позволяющие работать с асимптотиками и оценивать сложность работы алгоритмов. Семестр посвящён изучению структур данных, необходимых для разнообразных более сложных алгоритмов. Простейшие структуры стек, очередь, вектор анализируются на предмет эффективности и времени выполнения. Вводятся кучи (двоичная, биномиальная и фибоначиева), описываются границы их применимости. Изучаются деревья поиска (splay, AVL, декартово, В-дерево) вместе с подробными доказательствами корректности и асимптотики, а также с описанием прикладных преимуществ каждой структуры. Рассматриваются наиболее универсальные техники обработки запросов: хэш-таблицы, деревья отрезков, деревья Фенвика (в том числе многомерные), разреженные таблицы. В рамках рассматриваемых тем оттачиваются различные техники оценки временной сложности алгоритмов: метод потенциалов и метод бухгалтерского учёта. Курс в целом рассчитан на изучение базовых структур, реализация которых требуется во множестве более продвинутых алгоритмов.

Слушатели знакомятся с понятиями алгоритмов, асимптотик, графов, а также рассчитывая на умение студентов самостоятельно реализовывать программный код, обеспечивающий работу сформулированного алгоритма, курс раскрывает аспекты теории графов в области поиска максимального потока в транспортных сетях (в том числе минимальной стоимости), затем переходит к изучению базовых и продвинутых алгоритмов на строках (вкуче с оптимальными способами хранения строк и структур над ними в памяти компьютера), и, наконец, завершается блоком о вычислительной геометрии (освещаются технические детали возможного представления геометрических примитивов в цифровом виде, а также обсуждаются многие классические алгоритмы, актуальность которых обусловлена в первую очередь практическими приложениями).

Дисциплина включает подробное освещение теоретической стороны алгоритмов, разбор и тренировка решений практических задач, а также предполагает самостоятельное изучение студентами материала предмета через решение домашних теоретических и практических задач. Для освоения курса необходимы базовые понимания о понятии алгоритма и работе компьютера; также требуется достаточная подкованность в простейших определениях и терминах дискретной математики.

1. Цели и задачи

Цель дисциплины

Целями дисциплины «Алгоритмы и структуры данных» являются расширенное ознакомление студентов с основными принципами проектирования и анализа алгоритмов и структур данных, закрепление навыков обоснования корректности алгоритмов, их практической реализации, теоретической и экспериментальной оценки их временной сложности.

Задачи дисциплины

- научить формулировать задачи в терминах изученных теорий, выбирать подходящий алгоритм для поставленной задачи;
- научить разрабатывать комбинации алгоритмов для решения поставленных задач, оценивать сложности алгоритмов, выбирать подходящие структуры данных для поставленных задач, реализовывать алгоритмы на языке программирования C++.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук и использовать их в профессиональной деятельности	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-2 Способен использовать современные информационные технологии и	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области

программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.3 Знает основные требования информационной безопасности

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- определения асимптотик в O-нотации;
- определения простейших линейных структур данных (стек, очередь, вектор) и времена обработки запросов в них;
- алгоритм быстрой сортировки;
- определение и практическую необходимость деревьев поиска;
- определения потоков в сети, базовых функций над строками, базовых геометрических объектов;
- алгоритмы для нахождения максимального потока в сети (в т.ч. минимальной стоимости);
- алгоритмы поиска шаблона в тексте;
- способы представления геометрических объектов в памяти компьютера.

уметь:

- оценивать сложность алгоритмов;
- строго доказывать утверждения о корректности алгоритмов;
- применять необходимую технику для решения алгоритмических задач;

владеть:

- разнообразными методами пересечения базовых геометрических примитивов;
- методами доказательства корректности утверждений об алгоритмах;
- приёмами сведения общих задач к более конкретным и простым.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Асимптотики, мастер-теорема	7	7		12
2	Линейные структуры данных	8	8		12
3	Сортировки и порядковые статистики	7	7		11
4	Кучи	8	8		10
5	Деревья поиска	6	6		10
6	Дерево отрезков, дерево Фенвика	6	6		10
7	Хэш-таблицы, фильтры Блума	6	6		10
8	Паросочетания, алгоритм Куна	6	6		8
9	Максимальные потоки в сетях	6	6		7
10	Простейшие строковые алгоритмы	6	6		10
11	Строковые суффиксные структуры	6	6		10
12	Геометрические примитивы	6	6		10
13	Выпуклая оболочка	6	6		8
14	Рандомизированные алгоритмы и комбинаторные игры	6	6		7
Итого часов		90	90		135
Подготовка к экзамену		90 час.			

Общая трудоёмкость	405 час., 9 зач.ед.
--------------------	---------------------

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 1 (Осенний)

1. Асимптотики, мастер-теорема

Обозначения в О-нотации: о-малое и О-большое, омега-малое и Омега-большое, Тета-большое. Независимость определения О-большого и Омега-большого от начального сдвига. Мастер-теорема, пример применения для рекурренты $T(n) = 2T(n/2) + O(n)$.

2. Линейные структуры данных

Структуры данных стек, очередь, вектор, дек. Поиск ближайшего большего справа за $O(n)$ в массиве. Поиск минимума в стеке и очереди. Метод бухгалтерского учёта для доказательства асимптотики времени обработки запросов в векторе.

3. Сортировки и порядковые статистики

Задача сортировки. Определение стабильной сортировки. Сортировка слиянием, подсчёт числа инверсий в перестановке. Стабильная сортировка подсчётом, цифровая сортировка LSD. Быстрая сортировка со случайным выбором пивота, поиска k-й порядковой статистики. Дерандомизация: детерминированный алгоритм быстрой сортировки с выбором в качестве пивота медианы массива медиан пятёрок.

4. Кучи

Определение кучи и запросы, необходимые для обработки. Двоичная куча: операции siftUp и siftDown. Выражение остальных операций через данные. Асимптотика времени работы. Биномиальные деревья и биномиальная куча: скорость работы и преимущества по сравнению с двоичной кучей. Фибоначчиева куча: асимптотика с помощью метода бухгалтерского учёта.

Семестр: 2 (Весенний)

5. Деревья поиска

Определение дерева поиска, обрабатываемые запросы. Теоретическая реализация и анализ времени работы деревьев: splay-дерева, AVL-дерева, декартового дерева, B-дерева как частного случая (a, b)-дерева. Практические применения и преимущества каждого типа деревьев.

6. Дерево отрезков, дерево Фенвика

Обрабатываемые запросы в дереве отрезков. Отложенные операции. Дерево отрезков снизу. Двумерное дерево отрезков. Динамическое и персистентное дерево отрезков. Дерево Фенвика: булевы операции над битами. Многомерное дерево отрезков, запросы к подотрезкам и подпрямоугольникам.

7. Хэш-таблицы, фильтры Блума

Задача хэширования. Определения совершенного и универсального семейства хэш-функций. Вероятность коллизии. Хэш-таблицы с открытой адресацией, хэш-таблицы методом цепочек. Двойное хэширование. Фильтры Блума: применения и реализация.

8. Паросочетания, алгоритм Куна

Определение паросочетания, двудольного графа. Максимальное и наибольшее паросочетание. Теорема Берга, увеличивающие и чередующиеся цепи. Алгоритм Куна, корректность и асимптотика. Реализация. Теорема Дилворта и Мирского. Покрытие частично упорядоченного множества путями. Минимальное вершинное покрытие и максимальное независимое множество

9. Максимальные потоки в сетях

Определение транспортной сети, потока, разреза. Теорема Форда-Фалкерсона. Алгоритм Форда-Фалкерсона, Эдмондса-Карпа. Блокирующий поток, слоистая сеть. Алгоритм Диница. Взвешенная задача, поток минимальной стоимости. Алгоритм поиска максимального потока минимальной стоимости (с использованием алгоритмов Форда-Беллмана и Дейкстры с потенциалами).

Семестр: 3 (Осенний)

10. Простейшие строковые алгоритмы

Зет- и префикс-функция строки. Алгоритмы их нахождения за линейное время. Алгоритм Манакера. Структура данных бор. Алгоритм Ахо-Корасик. Поиск вхождения шаблона в строку и шаблонов в текст. Подсчёт числа вхождений.

11. Строковые суффиксные структуры

Суффиксное дерево и суффиксный автомат. Нахождение числа всех подстрок строки.

12. Геометрические примитивы

Точка, прямая, окружность, отрезок, луч. Представление в памяти компьютера. Скалярное и векторное произведение, две формы выражения. Поиск пересечения двух прямых, прямой и окружности, двух окружностей. Нахождение замечательных точек треугольника: точки пересечения биссектрис, медиан, высот.

13. Выпуклая оболочка

Определение выпуклой оболочки. Алгоритмы Джарвиса, Грэхема, Эндрю. Алгоритм Чена. Поиск объемлющей фигуры минимального периметра или площади.

14. Рандомизированные алгоритмы и комбинаторные игры

Минимальная покрывающая окружность. Непустота пересечения полуплоскостей. Определение игры ним, эквивалентности игр. Теория Шпрага-Гранди об эквивалентности любой справедливой ациклической игры игре ним. Ретроанализ.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, с доской.

6. Перечень рекомендуемой литературы

Основная литература

1. Введение в теорию алгоритмов и структур данных, Электронная версия печатной публикации / М. А. Бабенко, М. В. Левин. — Москва, МЦНМО, 2016

Дополнительная литература

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Не используются

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

Для успешного освоения дисциплины, студент использует следующие программные средства:

- компилятор языка C++.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

В преподавании дисциплины сочетаются практические занятия в компьютерном классе и домашние задания, состоящие из теоретических задач и задач на программирование с автоматической проверкой корректности.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладная математика и информатика
профиль подготовки:	Информатика Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	1
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Экзамен
- 2 (весенний) - Экзамен
- 3 (осенний) - Экзамен

Разработчик: И.Д. Степанов, ассистент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук и использовать их в профессиональной деятельности	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.3 Знает основные требования информационной безопасности

2. Показатели оценивания компетенций

В результате изучения дисциплины «Алгоритмы и структуры данных. Продвинутый поток (ФПМИ)» обучающийся должен:

знать:

- определения асимптотик в O-нотации;
- определения простейших линейных структур данных (стек, очередь, вектор) и времена обработки запросов в них;
- алгоритм быстрой сортировки;
- определение и практическую необходимость деревьев поиска;
- определения потоков в сети, базовых функций над строками, базовых геометрических объектов;
- алгоритмы для нахождения максимального потока в сети (в т.ч. минимальной стоимости);
- алгоритмы поиска шаблона в тексте;
- способы представления геометрических объектов в памяти компьютера.

уметь:

- оценивать сложность алгоритмов;
- строго доказывать утверждения о корректности алгоритмов;
- применять необходимую технику для решения алгоритмических задач;

владеть:

- разнообразными методами пересечения базовых геометрических примитивов;
- методами доказательства корректности утверждений об алгоритмах;
- приёмами сведения общих задач к более конкретным и простым.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1 семестр:

1. По данному числу n найдите все пары целых положительных чисел (a, b) , такие что $a \leq b \leq n$, и $a \mid b$. Оцените асимптотическое поведение числа таких пар.
2. Число 0 записано в n -разрядной двоичной системе. К нему $2^n - 1$ раз прибавляется единица. Будем считать, что время, необходимое на прибавление единицы, равно количеству единиц в двоичной записи числа, которые становятся нулями. Оцените среднюю сложность всех таких операций. Какие операции являются самыми дешёвыми, а какие -- самыми дорогими?
3. Предложите метод хранения минимального значения в очереди с помощью структуры `deque`. Если к очереди поступило n запросов, время работы программы должно составлять $O(n)$.

4. Напомним, что процедура $\text{Partition}(A, x)$ переупорядочивает элементы массива A так, что сначала идут все элементы, не превосходящие x , в некотором порядке, а затем -- все элементы, большие x . Покажите, как реализовать $\text{Partition}(A, x)$ с привлечением $O(1)$ дополнительной памяти.
5. Нижняя оценка на число сравнений в сортировке, основанной на сравнениях.
6. Лемма о корректности siftDown и siftUp .
7. Удаление в куче (по указателю и по значению).
8. Двумерное дерево отрезков снизу: прибавление в точке и сумма в прямоугольнике.
9. Фибоначчиева куча: операций decreaseKey . Асимптотика и корректность.
10. Фильтры Блума: асимптотика и корректность.

2 семестр:

1. Сведите задачу поиска максимального паросочетания в двудольном графе к поиску максимального потока в некоторой сети. Сравните время работы алгоритмов Форда--Фалкерсона и Эдмондса--Карпа. Можно ли проделать то же в произвольном (необязательно двудольном) графе?
2. На фабрику поступило n заказов, реализация i -го из которых принесёт ей прибыль в $a_{\{i\}}$ рублей. Каждый заказ для производства требует некоторого набора инструментов. Пусть всего зависимостей “заказ--инструмент” ровно k . Инструменты можно переиспользовать, то есть задействовать в нескольких заказах. Однако в данный момент на фабрике вообще нет инструментов, так что покупка j -го из них обойдётся в $b_{\{j\}}$ рублей. Конечно, от приёма каких-то заказов или покупки каких-то инструментов фабрика может отказаться. Какие заказы следует реализовать для максимизации прибыли (с учётом затрат на инструменты)?
3. Задан полный взвешенный двудольный граф, в обеих долях которого находится по n вершин. Предложите алгоритм поиска самого дешёвого совершенного паросочетания.
4. С помощью суффиксного автомата найдите количество различных подстрок строки s за время, линейное от её длины.
5. Пусть из n объектов только один обладает хорошим свойством. Смоделируем выбор с возвращением: на каждом шаге выбирается случайный равновероятный объект. Если он не обладает хорошим свойством, предмет кладётся обратно в кучу. В противном случае игра останавливается. Определите математическое ожидание числа шагов до конца игры.
6. Поиск минимального вершинного покрытия с доказательством корректности.
7. Теоремы Карзанова. Леммы для первой теоремы (б/д), вывод из них самой теоремы. Вторая теорема -- б/д. Единичные сети и алгоритм Хопкрофта--Карпа.
8. Поиск окружности минимального радиуса, покрывающей данное множество точек.
9. Зет-функция: определение, линейный алгоритм построения и доказательство корректности. Поиск всех вхождений шаблона s в текст t .
10. Алгоритм Ахо--Корасик: поиск всех вхождений шаблона s (в котором допускается не более k вопросов) в текст t .

3 семестр:

1. Пусть $c > 0$ -- константа. Докажите, что решение рекуррентности $T(n) = T(n/3) + T(2n/3) + cn$ ведёт себя как $\Omega(n \log n)$.
2. Разработайте стек, который умеет прибавлять ко всем хранящимся значениям произвольную поправку x за $O(1)$. Иными словами, нужно реализовать операцию увеличения всех чисел в стеке на x .
3. В $2n - 1$ ящиках лежат яблоки и апельсины. Требуется выбрать n ящиков так, что в них окажется не менее половины всех яблок и не менее половины всех апельсинов. Докажите, что такой выбор всегда существует.
4. Двоичная куча с минимумом в корне на n элементах расположена в памяти в виде массива. По числу k найдите минимальные k элементов в куче за $O(k \log k)$.
5. Пусть G -- ациклический ориентированный граф. Предложите алгоритм поиска наименьшего количества вершинно непересекающихся путей, которые покрывают все вершины G . Асимптотика: $O(nm)$, где n -- число вершин в G , а m -- число его рёбер.
6. Модифицируйте алгоритм Куна так, чтобы его время работы составляло $O(am)$, где a -- размер максимального паросочетания, а m -- число рёбер графа.
7. Докажите теорему о декомпозиции потока: если в сети G течёт некий поток f , то его можно разбить на несколько путей из s в t , а также на несколько (замкнутых) циклов.

8. Как реализовать алгоритм Ахо--Корасик на динамически расширяющемся множестве строк? Используйте идею разложения n по степеням двойки.
9. Предложите алгоритм нахождения пересечения двух выпуклых многоугольников за $O(nm)$, если n -- это количество сторон в первом из них, а m -- во втором.
10. Поток минимальной стоимости. Корректность алгоритма, асимптотика. Применение алгоритмов Форда-Беллмана и Дейкстры (с потенциалами).

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

1 семестр:

1. Динамический массив.
2. Амортизационный анализ.
3. Учетная оценка времени добавления элемента в динамический массив (с удвоением заполненного буфера).
4. Связные списки.
5. Стек, очередь, дек и их реализации.
6. Быстрая сортировка (QuickSort).
7. Поиск порядковой статистики методом "Разделяй и властвуй" (QuickSelect).
8. Сортировка слиянием (MergeSort).
9. Поразрядные сортировки.
10. Двоичная куча и сортировка кучей (HeapSort).
11. Слияние k отсортированных массивов с помощью кучи.
12. Хеш-таблица, полиномиальная хеш-функция.
13. Динамическое программирование: общая идея, линейная динамика, матричная, динамика на отрезках, на масках.
14. RMQ. Sparse table. Дерево отрезков.
15. LCA: сведение к RMQ и метод двоичного подъёма.
16. Двоичное дерево поиска.

Примеры билетов 1 семестр:

Билет №1:

1. Сортировка слиянием (MergeSort).
2. Хеш-таблица, полиномиальная хеш-функция.

Билет №2:

1. Двоичная куча и сортировка кучей (HeapSort).
2. RMQ. Sparse table. Дерево отрезков.

2 семестр:

1. Обходы в глубину и в ширину.
2. Поиск ключа, наивные вставка и удаление ключа.
3. AVL-дерево.
4. Красно-чёрное дерево.
5. Определение, описание алгоритма insert.
6. Декартово дерево.
7. Декартово дерево по неявному ключу.
8. Минимальное остовное дерево: алгоритмы Прима и Крускала.
9. Максимальные потоки в сети.
10. Методы: Форда-Фалкерсона; Эдмондса-Карпа (б/д).
11. Обход графа в глубину, ширину.
12. Поиск кратчайших путей в графе: алгоритмы Дейкстры, Форда-Беллмана, Флойда-Уоршелла.
13. Поиск сильно-связных компонент в графе. Задача 2SAT.
14. Мосты и точки сочленения в графе.
15. Нахождение подстроки в строке: префикс-функция, алгоритм Кнута-Морриса-Пратта.

Примеры билетов 2 семестр:

Билет №1:

1. Максимальные потоки в сети.
2. Поиск сильно-связных компонент в графе. Задача 2SAT.

Билет №2:

1. Обход графа в глубину, ширину.
2. Поиск кратчайших путей в графе: алгоритмы Дейкстры, Форда-Беллмана, Флойда-Уоршелла.

3 семестр:

1. Стандартные контейнеры: vector, deque, list, set, map, unordered_map, итераторы, компараторы.
2. Бор. Алгоритм Ахо-Корасик.
3. Суффиксное дерево.
4. Вычисление выпуклой оболочки множества точек в 2D.
5. Триангуляция Делоне.
6. Диаграмма Вороного.
7. Определение, алгоритмы построения (б/д). Примеры применения.
8. Длинная арифметика.
9. Сложение, вычитание, хранение знака, умножение, деление.
10. Алгоритм Карацубы.
11. Быстрое преобразование Фурье.
12. Планиметрия: примитивы точки, прямой, окружности.
13. Построение прямой. Пересечения (прямых, окружностей).
14. Проверка принадлежности точки многоугольнику.
15. Паросочетания в двудольных графах, алгоритм Куна.
16. Максимальное независимое множество, минимальное вершинное покрытие.

Примеры билетов 3 семестр:

Билет №1:

1. Вычисление выпуклой оболочки множества точек в 2D.
2. Планиметрия: примитивы точки, прямой, окружности.

Билет №2:

1. Сложение, вычитание, хранение знака, умножение, деление.
2. Паросочетания в двудольных графах, алгоритм Куна.

Критерии оценивания

отлично

- 10 Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы, код оформлен в едином удобочитаемом стиле
- 9 Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы
- 8 Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач

хорошо

- 7 Полностью решены все задачи. Допущены несущественные ошибки.
- 6 Полностью решено большинство задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.
- 5 Полностью решено две трети задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

удовлетворительно

4 Полностью решено более половины задач. В остальных задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

3 Полностью решено более половины задач.

неудовлетворительно

2 Решено менее половины задач.

1 Не решено ни одной задачи.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Во время проведения экзамена обучающиеся могут пользоваться программой дисциплины.