

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Смирнов Сергей Николаевич
Должность: врио ректора
Дата подписания: 13.05.2024 10:49:30
Уникальный программный ключ:
69e375c64f7e975d4e8830e7b4fcc2ad1bf35f08

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Тверской государственный университет»

Утверждаю:
Руководитель ООП

/С.М.Дудаков/

«13» *мая* 2024 года

Рабочая программа дисциплины (с аннотацией)

АЛГОРИТМЫ И АНАЛИЗ СЛОЖНОСТИ

Направление подготовки
02.03.02 ФУНДАМЕНТАЛЬНАЯ ИНФОРМАТИКА
И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Направленность (профиль)
Программная инженерия в искусственном интеллекте

Для студентов 3 и 4-го курса

Очная форма

Составитель: Б.Н. Карлов

Тверь, 2024

I. Аннотация

1. Цель и задачи дисциплины:

Изложить классификацию алгоритмических задач и алгоритмов, основанную на их сложности. Ознакомить студентов с типичными методами разработки эффективных алгоритмов и с эффективными алгоритмами решения задач из важнейших разделов дискретной математики и программирования. В частности, рассмотреть алгоритмы сортировки и поиска информации, алгоритмы для работы с множествами, алгоритмы для задач теории графов, базовые алгоритмы вычислительной геометрии, алгоритмы умножения матриц, алгоритмы для поиска образцов в строках. Развить у студентов умение оценивать сложность готовых алгоритмов и задач и конструировать собственные эффективные алгоритмы. Дать представление о типичных NP-полных задачах, для которых неизвестны эффективные алгоритмы, и о подходах к их решению.

2. Место дисциплины в структуре ООП

Дисциплина входит в раздел «Информатика и коммуникационно-информационные технологии» обязательной части блока 1.

Предварительные знания и навыки. Знание курсов «Дискретная математика», «Теоретические основы информатики», «Методы программирования», «Математическая логика и теория алгоритмов».

Дальнейшее использование. Полученные знания используются в последующем при изучении предметов: «Базы данных», «Программная инженерия» и других, а также для продолжения обучения в магистратуре.

3. Объем дисциплины: 9 зачетных единиц, 324 академических часа, в том числе:

контактная аудиторная работа лекций 109 часов, практических занятий 46 часов,

контактная внеаудиторная работа контроль самостоятельной работы 10 часов, в том числе расчетно-графическая работа 10 часов;

самостоятельная работа 159 часов, в том числе контроль 68 часов.

4. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы:

Планируемые результаты освоения образовательной программы (формируемые компетенции)	Планируемые результаты обучения по дисциплине
ОПК-3, Способен к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	ОПК-3.1 Знает основные положения и концепции в области программирования ОПК-3.2 Знает архитектуру языков программирования ОПК-3.3 Составляет программы

5. Форма промежуточной аттестации и семестр прохождения:

экзамен в 6–7 семестрах

6. Язык преподавания:

русский

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

Для студентов очной формы обучения

Учебная программа – наименование разделов и тем	Всего (час.)	Контактная работа (час.)				Контроль сам. раб., в т.ч. РГР	Сам. раб., в т.ч. контроль (час.)
		Лекции		Практические занятия			
		Всего	В т.ч. практ. подготовка	Всего	В т.ч. практ. подготовка		
1	2	3	4	5	6	7	8
Модели вычислений	20	8		3		0	10
Базовые структуры	31	9		3		10	10

данных и основные методы разработки эффективных алгоритмов							
Сортировка и поиск k -го наименьшего элемента	46	14		6		0	26
Задачи поиска и работа с множествами	77	24		11		0	40
Алгоритмы на графах	61	18		8		0	34
Умножение матриц и связанные задачи	19	8		3		0	9
Алгоритмы на строках	35	14		6		0	15
NP-полные задачи	35	14		6		0	15
Итого	324	109	0	46	0	10	159

Учебная программа дисциплины

1. Модели вычислений

- Сравнение скорости роста функций.
- Машины с произвольным доступом к памяти. Меры сложности вычислений.
- РАМ (ПДП-машины) и машины Тьюринга.
- Линейные программы. Битовые линейные программы. Ветвящиеся программы (деревья сравнений).
- Модельный алгоритмический язык. Сложность реализации основных конструкций на ПДП-машине.

2. Базовые структуры данных и основные методы разработки эффективных алгоритмов

- Списки, стеки (магазины), очереди. Алгоритмы выполнения основных операций
- Графы, деревья, бинарные деревья. Способы представления. Алгоритмы обхода деревьев.
- Метод разработки алгоритмов «разделяй и властвуй». Алгоритм умножения двоичных чисел (алгоритм Карацубы). Техническая теорема об оценке роста функций, заданных рекуррентными соотношениями. Возведение в степень по простому модулю и его применения в шифровании.
- Динамическое программирование. Оптимальное умножение последовательности матриц. Алгоритм Кока-Янгера-Касами для распознавания КС-языков. Задача глобального выравнивания слов.

3. Сортировка и поиск k -го наименьшего элемента

- Нижние оценки числа сравнений (в «худшем» и в «среднем»).
 - Алгоритмы сортировки обменами (быстрая сортировка) и слиянием.
 - Алгоритм пирамидальной сортировки (с помощью дерева).
 - Алгоритм лексикографической сортировки.
 - Линейные алгоритмы нахождения k -го наименьшего элемента (в «худшем» и в «среднем»).
 - Точная оценка числа сравнений для нахождения 2-го по величине элемента множества (теорема Кислицына).
4. Задачи поиска и работа с множествами
- Метод расстановки (хеширование).
 - Методы разрешения коллизий: внешние и внутренние цепочки, открытая адресация, повторное хеширование.
 - Анализ сложности вставки и поиска при хешировании. Практический выбор хеш-функции.
 - Деревья двоичного поиска. Алгоритм построения оптимального дерева двоичного поиска.
 - 2-3-деревья. Алгоритмы вставки и удаления элементов из 2-3-дерева.
 - B-деревья и алгоритмы работы с ними.
 - Алгоритмы для работы с системами непересекающихся множеств с использованием массивов и списков.
 - Алгоритмы для работы с системами непересекающихся множеств с использованием древовидных структур (сжатие путей).
 - Алгоритм проверки эквивалентности конечных автоматов.
 - Сливаемые деревья и сцепляемые очереди.
 - Биномиальные и фибоначчиевы кучи и алгоритмы работы с ними.
 - Структуры данных для представления пространственной информации: 2-d деревья, точечные квадродеревья, MX-квадродеревья.
5. Алгоритмы на графах
- Минимальное остовное дерево: алгоритмы Крускала, Прима и Борувки.
 - Поиск в глубину и поиск в ширину в неориентированных и ориентированных графах. Топологическая сортировка.
 - Алгоритмы определения мостов и двусвязных компонент графа.
 - Алгоритмы определения сильно связных компонент и баз графа.
 - Алгоритмы построения транзитивного замыкания графа и нахождения кратчайших путей (алгоритмы Уоршола и Уоршола-Флойда).
 - Задача о кратчайших путях из одного источника (алгоритмы Дейкстры и Беллмана-Форда). Эффективная реализация алгоритма Дейкстры.
 - Задача о максимальном потоке в сетях. Алгоритм Форда-Фалкерсона.
 - Алгоритм нахождения максимального потока за кубическое время.
 - Простые сети и задача о максимальном паросочетании для двудольных графов.
 - Задача о стабильном паросочетании.

- Планарные графы.
 - Раскраска графов. 5-раскраска плоских графов.
6. Умножение матриц и связанные задачи
- Связь задачи умножения булевых матриц с задачей нахождения транзитивного замыкания графа.
 - Алгоритм Штрассена для умножения матриц.
 - Алгоритм «четырех русских» для умножения булевых матриц.
 - НВП-разложение матриц.
 - Сложность обращения матриц, вычисления определителя и решения системы линейных уравнений.
7. Алгоритмы на строках
- Регулярные выражения, языки и недетерминированные конечные автоматы. Распознавание образцов, задаваемых регулярными выражениями.
 - Алгоритм Кнута-Морриса-Пратта для задачи вхождения подслов.
 - Z-алгоритм для задачи вхождения подслов.
 - Алгоритм Бойера-Мура для задачи вхождения подслов.
 - Алгоритм Рабина-Карпа для задачи вхождения подслова.
 - Суффиксные деревья и решаемые с их помощью задачи. Алгоритм Укконена для построения суффиксного дерева за линейное (от его размера) время.
 - Применения суффиксных деревьев. Поиск повторяющихся структур в строке. Сжатие данных методом Зива-Лемпеля.
 - Коды Хаффмана.
8. NP-полные задачи
- Классы P и NP. Сводимость за полиномиальное время. Теорема Кука-Левина о NP-полноте задачи выполнимости булевых формул.
 - Полиномиальный алгоритм для задачи выполнимости 2-КНФ.
 - Примеры NP-полных задач в логике, теории графов, алгебре, комбинаторике, математическом программировании: 3-КНФ, КЛИКА, ВЕРШИННОЕ ПОКРЫТИЕ, ГАМИЛЬТОНОВ ЦИКЛ, РАСКРАСКА, 3-СОЧЕТАНИЕ, РАЗБИЕНИЕ, РЮКЗАК, 0-1 ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ, КОММИВОЯЖЕР, МНОГОПРОЦЕССОРНОЕ РАСПИСАНИЕ, УРАВНЕНИЯ В СЛОВАХ и др.
 - Подходы к решению NP-полных задач с использованием эвристических алгоритмов. Псевдополиномиальные и приближенные алгоритмы. Метод ветвей и границ.
 - Класс PSPACE. Полнота задачи QBF в этом классе. PSPACE-полные игры (обобщенная география, гекс).
 - Соотношения между классами сложности. Теорема Сэвича. Теорема Иммермана. Иерархия по времени и емкости. Теорема о том, что $P \neq EXPTIME$. Теорема Ладнера.
 - Машины Тьюринга с оракулами. Пример оракулов A и B таких, что $P^A = NP^A$, $P^B \neq NP^B$.

- Абстрактная сложность вычислений. Аксиомы Блюма. Теорема Цейтина. Теорема Рабина. Теорема Блюма. Теорема о пробелах.

III. Образовательные технологии

Учебная программа – наименование разделов и тем	Вид занятия	Образовательные технологии
Модели вычислений	лекции, практические занятия	изложение теоретического материала, решение задач
Базовые структуры данных и основные методы разработки эффективных алгоритмов	лекции, практические занятия	изложение теоретического материала, решение задач, выполнение РГР
Сортировка и поиск k -го наименьшего элемента	лекции, практические занятия	изложение теоретического материала, решение задач
Задачи поиска и работа с множествами	лекции, практические занятия	изложение теоретического материала, решение задач
Алгоритмы на графах	лекции, практические занятия	изложение теоретического материала, решение задач
Умножение матриц и связанные задачи	лекции, практические занятия	изложение теоретического материала, решение задач
Алгоритмы на строках	лекции, практические занятия	изложение теоретического материала, решение задач
NP-полные задачи	лекции, практические занятия	изложение теоретического материала, решение задач

IV. Оценочные материалы для проведения текущей и промежуточной аттестации

Типовые контрольные задания и/или критерии для проверки индикатора ОПК-3.1

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
Знать типичные приемы и методы разработки эффективных алгоритмов.	Примеры вопросов к экзамену: • Метод «Разделяй и властвуй». Основная теорема о рекуррентных оценках. Умножение длинных чисел (алгоритм Карацубы). Возведение в степень по простому модулю и его применения в шифровании.	оценка 3 — знает основные методы разработки эффективных алгоритмов,

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
	<ul style="list-style-type: none"> • Динамическое программирование. Оптимальное умножение последовательности матриц. Алгоритм Кока-Янгера-Касами для распознавания КС-языков. Глобальное выравнивание слов. 	<p>оценка 4 — кроме того знает стандартные алгоритмы динамического программирования и алгоритмы типа «Разделяй и властвуй», оценка 5 — кроме того знает доказательства корректности этих алгоритмов</p>
<p>Знать основные структуры данных, алгоритмы сортировки и поиска информации.</p>	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Списки, стеки, очереди, графы, деревья. Представления деревьев. Обходы деревьев. • Алгоритмы сортировки. Быстрая сортировка, пирамидальная сортировка, лексикографическая сортировка. Нижние оценки числа сравнений в алгоритмах сортировки. • Линейные алгоритмы нахождения k-го наименьшего элемента. Точная оценка числа сравнений для нахождения второго по величине элемента множества. • Хеширование. Методы разрешения коллизий (внешние цепочки, внутренние цепочки, открытая адресация, повторное хеширование). Среднее время выполнения n операций на хеш-таблице. • Деревья двоичного поиска. Оптимальные деревья двоичного поиска. • 2-3-деревья, В-деревья. Алгоритмы 	<p>оценка 3 — знает основные структуры данных и простейшие алгоритмы для работы с ними, оценка 4 — кроме того знает более сложные структуры данных и доказательства правильности алгоритмов, оценка 5 — кроме того знает теоремы о нижних границах вре-</p>

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
	<p>вставки и удаления элемента.</p> <ul style="list-style-type: none"> • Алгоритмы для работы с системами непесекающихся множеств с использованием массивов и списков и с использованием деревьев. Проверка эквивалентности конечных автоматов. • Сливаемые деревья и сцепляемые очереди и алгоритмы работы с ними. • Биномиальные и фибоначчиевы кучи и алгоритмы работы с ними. • Структуры данных для представления пространственной информации: 2-d деревья, точечные квадродревья, МХ-квадродревья. 	мени сортировки и поиска и их доказательства
Знать основные алгоритмы для задач теории графов.	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Минимальное остовное дерево: алгоритмы Крускала, Прима и Борувки. • Поиск в глубину и поиск в ширину. Топологическая сортировка. • Алгоритмы определения мостов и двусвязных компонент графа. • Алгоритмы нахождения сильно связанных компонент и баз графа. • Алгоритмы построения транзитивного замыкания графа и нахождения кратчайших путей (алгоритмы Уоршолла и Уоршолла-Флойда). • Задача о кратчайших путях из одного источника (алгоритмы Дейкстры и Беллмана-Форда). • Задача о максимальном потоке в сетях. Алгоритм Форда-Фалкерсона. Алгоритм нахождения максимального потока за кубическое время. Простые сети и задача о максимальном паросочетании для двудольных графов. • Планарные графы. Примеры непланарных графов. Теорема Куратовского (формулировка). 	оценка 3 — знает основные алгоритмы для задач на графах, оценка 4 — кроме того знает утверждения о работе этих алгоритмов, оценка 5 — кроме того знает доказательства этих утверждений

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
	<ul style="list-style-type: none"> • Задача раскраски графов. 5-раскраска плоских графов. 	
Знать основные алгоритмы для работы с матрицами.	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Связь задачи умножения булевых матриц с задачей нахождения транзитивного замыкания графа. • Алгоритм Штрассена. • Алгоритм «четырёх русских». • НВП-разложение матрицы. Обращение матриц, вычисление определителя, решение систем линейных уравнений. 	оценка 3 — знает простейшие алгоритмы для работы с матрицами, оценка 4 — кроме того знает более эффективные алгоритмы для работы с матрицами, оценка 5 — кроме того знает доказательства перечисленных выше утверждений и корректности алгоритмов
Знать основные NP-полные задачи и методы их решения.	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • NP-полнота задач 3-КНФ, КЛИКА, ВЕРШИННОЕ ПОКРЫТИЕ, ГАМИЛЬТОНОВ ЦИКЛ, РАСКРАСКА, 3-СОЧЕТАНИЕ, РАЗБИЕНИЕ, РЮКЗАК, 0-1 ЦЕЛОЧИСЛЕННОЕ ПРОГРАММИРОВАНИЕ, КОММИВОЯЖЕР, МНОГОПРОЦЕССОРНОЕ РАСПИСАНИЕ. • Методы решения NP-полных задач. Приближённые алгоритмы. Псевдополиномиальные алгоритмы. Метод ветвей и границ. 	оценка 3 — знает примеры NP-полных задач, оценка 4 — кроме того знает методы решения NP-полных задач, оценка 5 — кроме того знает доказательства NP-полноты стандартных задач
Знать основ-	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Полиномиальная сводимость. Трудные и 	оценка 3 —

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
ные теоретические результаты теории сложности	<p>полные проблемы. Теорема Кука-Левина об NP-полноте SAT.</p> <ul style="list-style-type: none"> • Теорема о PSPACE-полноте QBF. Примеры PSPACE-полных игр (обобщенная география, гекс). • Соотношения между классами сложности. Теорема Сэвича ($PSPACE = NPSPACE$). Теорема Иммермана ($NLOGSPACE = coNLOGSPACE$). Иерархия по времени и емкости. Теорема о том, что $P \neq EXPTIME$. Теорема Ладнера. • Машины Тьюринга с оракулами. Пример оракулов A и B таких, что $P^A = NP^A$, $P^B \neq NP^B$. • Абстрактная сложность вычислений. Аксиомы Блюма. Теорема Цейтина. Теорема Рабина. Теорема Блюма. Теорема о пробелах. 	<p>знает основные классы сложности, некоторые свойства этих классов, оценка 4 — знает более глубокие результаты теории сложности, оценка 5 — кроме того знает доказательства перечисленных выше утверждений</p>

Типовые контрольные задания и/или критерии для проверки индикатора ОПК-3.2

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
Знать основные алгоритмы для работы со строками.	<p>Примеры вопросов к экзамену:</p> <ul style="list-style-type: none"> • Моделирование недетерминированных конечных автоматов. Распознавание образцов, задаваемых регулярными выражениями. • Задача вхождения подстроки. Алгоритм Кнута-Морриса-Пратта. Z-алгоритм. Алгоритм Бойера-Мура. Алгоритм Рабина-Карпа. • Суффиксные деревья. Алгоритм Укконена. Применения суффиксных деревьев: поиск повторяющихся структур в строке, 	<p>оценка 3 — знает основные алгоритмы для задач на строках, оценка 4 — кроме того знает утверждения о работе этих алгоритмов, оценка 5 —</p>

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
	сжатие данных по Зиву-Лемпелю. • Коды Хаффмана.	кроме того знает доказательства этих утверждений
Уметь использовать стандартные алгоритмы на строках и разрабатывать собственные алгоритмы.	Примеры задач для контрольных работ: • Вычислите используемые в алгоритме Бойера-Мура функции R ($\Sigma = \{0, 1, 2\}$), L' и l' для распознавания вхождения подстроки $y = 101001201$ в текст методом Бойера-Мура. Используя их, найдите вхождение y в слово $x = 0112101001201012010$. • Обобщите алгоритм Рабина-Карпа на случай, когда надо искать квадрат A размером $t \times t$ в матрице S размером $n \times n$ (элементами квадрата и матрицы являются символы алфавита $\Sigma = \{a_1, \dots, a_r\}$). • Как расширить алгоритм моделирования НКА M , чтобы он находил в слове $x = a_1 \dots a_n$ наименьшее k такое, что для некоторого j подслово $a_j \dots a_k$ принадлежит $L(M)$, и наибольшее из таких j ? • Подстрока α называется префиксным повтором строки S , если α является префиксом S и имеет вид $\alpha = \beta\beta$ для некоторой строки β . Предложите линейный по времени алгоритм для отыскания наибольшего префиксного повтора входной строки S .	оценка 3 — умеет использовать стандартные алгоритмы на строках, оценка 4 — кроме того умеет модифицировать стандартные алгоритмы, оценка 5 — кроме того умеет разрабатывать алгоритмы для решения новых задач

Типовые контрольные задания и/или критерии для проверки индикатора ОПК-3.3

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
<p>Уметь применять типичные приемы и методы для решения алгоритмических задач.</p>	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Используя алгоритм Кока-Янгера-Касами, проверьте, входит ли слово $caabb$ в КС-язык, порождённый следующей КС-грамматикой ($N = \{A, B, C, S\}$, $\Sigma = \{a, b, c\}$). $S \rightarrow AA \mid AB$ $A \rightarrow a \mid AB \mid CA$ $B \rightarrow b \mid BB \mid BC$ $C \rightarrow c \mid CC \mid CA$ • Дополните алгоритм вычисления минимального числа операций при умножении матриц, чтобы он выдавал и оптимальный порядок умножений (соответствующую расстановку скобок). • Назовем элементарным преобразованием слова удаление буквы, вставку буквы или изменение одной буквы слова. Написать алгоритм, который по любой паре слов u, v подсчитывает минимальное число элементарных преобразований, необходимых для преобразования слова u в слово v и находит минимальную цепочку преобразований. Оценить его сложность. <p>Примеры тем для расчетно-графических работ:</p> <ul style="list-style-type: none"> • Напишите программу для умножения длинных чисел методом Карацубы и в столбик. Определите, при каком количестве цифр метод Карацубы оказывается эффективнее. 	<p>оценка 3 — умеет использовать стандартные алгоритмы, оценка 4 — кроме того умеет модифицировать стандартные алгоритмы, оценка 5 — кроме того умеет разрабатывать алгоритмы для новых задач</p>
<p>Уметь использовать стандартные структуры данных и разрабатывать</p>	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Выполните операцию ВСТАВИТЬ в МХ-квадродерево порядка 3 (область разбита на 8×8 ячеек) для следующей последовательности точек (X_i, Y_i): (6, 2), 	<p>оценка 3 — умеет пользоваться стандартными структурами</p>

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
<p>собственные структуры данных и алгоритмы для работы с ними.</p>	<p>(1, 1), (6, 5), (7, 6), (2, 5), (5, 7), (4, 5), (5, 1). Затем УДАЛИТЬ вершины с координатами (6, 2) и (5, 1). Изобразите деревья после всех вставок, а также после каждого из удалений.</p> <ul style="list-style-type: none"> • Напишите алгоритм корректировки меток $L:M$ во внутренних вершинах 2-3-дерева после выполнения операции ВСТАВИТЬ. Оцените его сложность (она должна быть $O(\log_2 n)$). • Назовем 2-3-4-деревом такое дерево, у каждой вершины которого, кроме корня и листьев, имеется 2, 3 или 4 сына и все листья располагаются на одинаковой глубине. 2-3-4-куча — это 2-3-4-дерево, в котором листья содержат ключи (не упорядоченные), а в каждой внутренней вершине хранятся значения наименьших ключей у каждого из её сыновей. В корне также хранится высота дерева. Предложите алгоритмы для реализации следующих операций над 2-3-4-кучами. Сложность должна быть $O(\log_2 n)$. <ul style="list-style-type: none"> 1) $\text{MIN}(T)$ — возвращает ссылку на лист с минимальным ключом. 2) $\text{DELETE}(T, x)$ — удаляет заданный лист x. • Элемент x массива $T[1..n]$ называется большинством, если $\{i : T[i] = x\} > n / 2$. Предложите алгоритм, который определяет, существует ли большинство, и находит его за время $O(n)$. Разрешается использовать только операции $=$ и \neq. <p>Примеры тем для расчетно-графических работ:</p> <ul style="list-style-type: none"> • Напишите программу для работы с B-деревьями. Реализуйте следующие операции: поиск, вставка, удаление, поиск минимума, поиск максимума. 	<p>данных, оценка 4 — кроме того умеет разрабатывать алгоритмы для выполнения новых операций на стандартных структурах данных, оценка 5 — кроме того умеет разрабатывать алгоритмы для новых структур данных</p>

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
<p>Уметь использовать стандартные алгоритмы на графах и разрабатывать собственные алгоритмы.</p>	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Используя алгоритм Форда-Фалкерсона, постройте максимальный поток для сети $N: \{(s, v_1, 6), (s, v_2, 2), (s, v_3, 5), (v_1, v_3, 3), (v_1, t, 2), (v_3, v_2, 1), (v_2, t, 4), (v_3, t, 4)\}$. • Модифицируйте алгоритм Уоршолла-Флойда так, чтобы после его завершения можно было за время $O(V)$ найти кратчайший путь между любой парой вершин. (Сложность «по порядку» не должна увеличиться.) • Назовём диаметром неориентированного дерева $T = (V, E)$ величину $\max \{ \delta(u, v) : u, v \in V \}$, где $\delta(u, v)$ — это число рёбер на простом пути между u и v. Предложите алгоритм для вычисления диаметра дерева за время $O(V + E)$, докажете его корректность и оцените сложность. • Предположим, что известен максимальный поток f_m в сети $N = (V, E, s, t, c)$ и пропускные способности всех рёбер — целые числа. Предположим, что пропускная способность некоторого ребра (u, v) уменьшилась на 1: $c'(u, v) = c(u, v) - 1$. Предложите алгоритм, находящий максимальный поток для изменённой сети за время $O(V + E)$. 	<p>оценка 3 — умеет использовать стандартные алгоритмы на графах, оценка 4 — кроме того умеет модифицировать стандартные алгоритмы, оценка 5 — кроме того умеет разрабатывать алгоритмы для решения новых задач</p>
<p>Уметь применять стандартные приемы для решения NP-полных задач, доказывать NP-полноту новых задач.</p>	<p>Примеры задач для контрольных работ:</p> <ul style="list-style-type: none"> • Для нагруженного графа G, представленного матрицей C, найдите гамильтонов путь, используя алгоритм 2MOD (с двойным обходом минимального остовного дерева). Сравните полученный результат с длиной 12 минимального пути. 	<p>оценка 3 — умеет использовать приближённые и псевдополиномиальные алгоритмы для решения NP-полных задач, оценка 4 — кроме того</p>

Требования к обучающемуся	Типовые контрольные задания для оценки знаний, умений, навыков	Показатели и критерии оценивания, шкала оценивания
	$C = \begin{pmatrix} 0 & 3 & 10 & 10 & 10 & 2 & 1 \\ 3 & 0 & 2 & 12 & 12 & 2 & 10 \\ 10 & 2 & 0 & 1 & 10 & 12 & 6 \\ 10 & 12 & 1 & 0 & 2 & 12 & 10 \\ 10 & 12 & 10 & 2 & 0 & 5 & 3 \\ 2 & 1 & 12 & 12 & 5 & 0 & 2 \\ 1 & 10 & 60 & 10 & 3 & 2 & 0 \end{pmatrix}$ <ul style="list-style-type: none"> • Пусть требуется уложить в контейнеры единичного объема $30m$ предметов с весами: $s(A_i) = 0.5 + \varepsilon \quad (1 \leq i \leq 6m)$ $s(A_i) = 0.25 + 2\varepsilon \quad (6m < i \leq 12m)$ $s(A_i) = 0.25 + \varepsilon \quad (12m < i \leq 18m)$ $s(A_i) = 0.25 - 2\varepsilon \quad (18m < i \leq 30m) \quad (\varepsilon > 0).$ Сколько контейнеров потребуется алгоритму, основанному на принципе «в первый подходящий в порядке убывания»? Покажите, что при оптимальной укладке достаточно $9m$ контейнеров и оцените ошибку приближённого алгоритма. • Докажите NP-полноту задачи ГАМИЛЬТОНОВО ПОПОЛНЕНИЕ. Вход: неориентированный граф $G = (V, E)$, число $k > 0$. Вопрос: можно ли к E добавить k рёбер, так чтобы в графе появился гамильтонов цикл? (Указание: к этой задаче сводится задача ГАМИЛЬТОНОВ_ЦИКЛ.) 	умеет использовать метод ветвей и границ, оценка 5 — кроме того умеет доказывать NP-полноту новых задач

V. Учебно-методическое и информационное обеспечение дисциплины

1. Рекомендованная литература

а) Основная литература

[1] Белов В. В. Алгоритмы и структуры данных: учебник / В. В. Белов, В. И. Чистякова; Рязанский государственный радиотехнический университет имени В.Ф. Уткина. - 1. - Москва: ООО "КУРС", 2020. - 240 с. - (Бакалавриат). - ВО - Бакалавриат. — Режим доступа:

<https://znanium.com/catalog/document?id=347241>

[2] Белик А. Г. Алгоритмы и структуры данных [Электронный ресурс]: учебное пособие / А. Г. Белик, В. Н. Цыганенко. - Омск: ОмГТУ, 2022. - 104 с.

– Режим доступа : <https://e.lanbook.com/book/343688>

[3] Рысин М. Л. Введение в структуры и алгоритмы обработки данных. Ч. 1: Сложность алгоритмов. Сортировки. Линейные структуры данных. Поиск в таблице / М. Л. Рысин, М. В. Сартаков, М. Б. Туманова. - Москва: РТУ МИРЭА, 2022. - 110 с. – Режим доступа: <https://e.lanbook.com/book/256592>

[4] Рысин М. Л. Введение в структуры и алгоритмы обработки данных. Ч. 2: Поиск в тексте. Нелинейные структуры данных. Кодирование информации. Алгоритмические стратегии / М. Л. Рысин, М. В. Сартаков, М. Б. Туманова. - Москва: РТУ МИРЭА, 2022. - 111 с. – Режим доступа: <https://e.lanbook.com/book/310826>

б) Дополнительная литература

[5] Бабенко, М.А. Введение в теорию алгоритмов и структур данных [Электронный ресурс] / М.А. Бабенко, М.В. Левин. — Электрон. дан. — Москва: МЦНМО, 2016. — 144 с. — Режим доступа: <https://e.lanbook.com/book/80136>. — Загл. с экрана.

[6] Дехтярь М.И. Алгоритмические задачи на графах. [Электронный ресурс] Тверь: ТвГУ, 2011. — Режим доступа: <http://texts.lib.tversu.ru/texts/09311ucheb.pdf>

2. Программное обеспечение

Наименование помещений	Программное обеспечение
Ауд. 201а (компьютерная лаборатория ПМик) (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Перечень программного обеспечения (со свободными лицензиями): Linux, Kubuntu, KDE, TeXLive, TeXStudio, LibreOffice, GIMP, Gwenview, ImageMagick, Okular, Skanlite, Google Chrome, KDE Connect, Konversation, KRDC, KTorrent, Thunderbird, Elisa, VLC media player, PulseAudio, KAppTemplate, KDevelop, pgAdmin4, PostgreSQL, Qt, QtCreator, R, RStudio, Visual Studio Code, Perl, Python, Ruby, clang, clang++, gcc, g++, nasm, flex, bison, Maxima, Octave, Dolphin, HTop, Konsole, KSystemLog, Xterm, Ark, Kate, Kcalc, Krusader, Spectacle, Vim.

3. Современные профессиональные базы данных и информационные справочные системы

[1] ЭБС «ZNANIUM.COM» <http://www.znanium.com>

[2] ЭБС «Университетская библиотека онлайн» <https://biblioclub.ru>

[3] ЭБС IPRbooks <http://www.iprbookshop.ru>

[4] ЭБС «Лань» <http://e.lanbook.com>

[5] ЭБС «Юрайт» <https://urait.ru>

[6] ЭБС ТвГУ <http://megapro.tversu.ru/megapro/Web>

- [7] Научная электронная библиотека eLIBRARY.RU (подписка на журналы) https://elibrary.ru/projects/subscription/rus_titles_open.asp
- [8] Репозиторий ТвГУ <http://eprints.tversu.ru>

4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

- [1] Data Structure and Algorithms Tutorial, https://www.tutorialspoint.com/data_structures_algorithms/index.htm
- [2] Graph Theory Tutorial, https://www.tutorialspoint.com/graph_theory/index.htm
- [3] Design and Analysis of Algorithms, <http://www.learnerstv.com/Free-Computer-Science-Video-lectures-ltv147-Page1.htm>

VI. Методические материалы для обучающихся по освоению дисциплины

Примеры задач для подготовки к контрольным работам

1. Расположите следующие функции в порядке возрастания (при $n \rightarrow \infty$):
a) $n \log n \log \log n$; b) $2^n / n^5$; c) $(n!)^{\log n!}$; d) $2^{n \log n}$; e) $n^{51/10} / \log^9 n$;
f) $n^5 / \log^{10} n$; g) $n^{0.5} \log^{10} n$; h) $n^5 \log^6 n$; i) 2^{2n} ; j) $n^{5.01} \log^{115} n$.
2. Назовем общей подпоследовательностью двух слов $V = v(1)v(2)\dots v(n)$ и $W = w(1)w(2)\dots w(m)$ последовательность индексов $1 \leq i_1 < \dots < i_k \leq n$ и последовательность индексов $1 \leq j_1 < \dots < j_k \leq m$ таких, что $v(i_1) = w(j_1), \dots, v(i_k) = w(j_k)$. Используя динамическое программирование, предложите алгоритм нахождения самой длинной общей подпоследовательности и оцените его сложность.
3. Пусть $A[1\dots n]$ и $B[1\dots n]$ — два возрастающих массива. Разработайте алгоритм, который за время $O(\log n)$ находит медиану множества, полученного объединением элементов этих массивов.
4. Пусть A — упорядоченный массив положительных и отрицательных целых чисел: $A[1] < A[2] < \dots < A[n]$. Написать алгоритм нахождения такого i , что $A[i] = i$ (если такое i существует). Оценить его сложность.
5. Написать процедуру HASH-DELETE для удаления элемента из таблицы с открытой адресацией (если потребуется, измените для этого процедуру вставки).
6. Постройте алгоритм для реализации в биномиальных кучах операции BIN-HEAP-MERGE(H_1, H_2), которая сливает корневые списки куч H_1 и H_2 в единый список, в котором вершины расположены в порядке неубывания степеней. Оцените сложность построенного алгоритма.

7. Перечислите все ситуации, в которых помеченная вершина попадает в корневой список фибоначчиевой кучи.
8. Предложите алгоритм ответа на запрос об области вида $\langle (x, y), R \rangle$ для МХ-квадродерева T , который должен вернуть множество вершин T , представляющих точки, которые находятся на расстоянии не больше R от точки (x, y) . Оцените его сложность.
9. Найти максимальное паросочетание в двудольном графе, используя сведения к задаче о максимальном потоке для простой сети.
 $G = (V = \{v_1, v_2, v_3, v_4\} \cup \{u_1, u_2, u_3, u_4, u_5\}, E = \{(v_1, u_3), (v_1, u_4), (v_1, u_5), (v_2, u_2), (v_3, u_1), (v_3, u_2), (v_4, u_2)\})$.
10. Предложите алгоритм, который по сети N и максимальному потоку в ней f_m находит такое ребро, увеличение пропускной способности которого приводит к увеличению максимального потока. Оцените его сложность.
11. Пусть $G = (V, E)$ — это ориентированный взвешенный граф с весовой функцией $c : E \rightarrow \{0, 1, \dots, M - 1\}$, где $M > 0$ — целое число. Модифицировать алгоритм Дейкстры, чтобы он находил в таком случае кратчайшие пути из одной вершины за время $O(M|V| + |E|)$ (еще лучше получить сложность $O(|V| + |E| \log M)$).
12. Требуется по двум словам S_1 и S_2 найти самое длинное слово P , которое входит как подслово и в S_1 , и в S_2 .
 - а) Докажите, что с использованием суффиксных деревьев эту задачу можно решить за время, линейное от $(|S_1| + |S_2|)$.
 - б) Используя предложенный алгоритм, найдите самое длинное общее подслово для слов $S_1 = abbba$ и $S_2 = baabba$.
13. Докажите NP-полноту задачи ГАМИЛЬТОНОВ ПУТЬ МЕЖДУ ДВУМЯ ВЕРШИНАМИ.
 Вход: неориентированный граф $G = (V, E)$ и две вершины $a, b \in V$.
 Вопрос: существует ли в G простой путь из a в b , проходящий по одному разу через все вершины графа?

Требования к рейтинг контролю (6 семестр)

Контрольная работа 1. Темы: сравнение скорости роста функций, динамическое программирование, сортировка, поиск. Пример задания:

1. Расположите следующие функции в порядке возрастания (при $n \rightarrow \infty$):
 - а) $n \log n$; б) $2^n n$; в) $n / \log^{10} n$; д) $n^{19/10} \log^4 n$; е) $2^{n^{1/n}}$;
 - ф) $n^2(\log n)^n$; г) $n^{2.05} / \log^{40}(n + 10)$; х) $n \log^3(n!)$; и) $25n^2$; j) $n^{1.91} \log^7 n$.
2. Даны две строки x и y . Строка x состоит из нулей и единиц, строка y — из символов А и В. Можно ли строку x преобразовать в строку y по следующему правилу: цифра 0 преобразуется в непустую последовательность букв

А, а цифра 1 — либо в непустую последовательность букв А, либо в непустую последовательность букв В? Постройте алгоритм, который отвечает на этот вопрос и находит преобразование, и оцените его сложность.

Указание. Пусть $C[i, j] = 1 \Leftrightarrow$ первые i символов последовательности x можно преобразовать в первые j символов последовательности y . Предложите алгоритм для вычисления массива C размера $|x| \times |y|$.

3. Пусть в каждой вершине дерева хранится список указателей на сыновей этой вершины. Напишите рекурсивный алгоритм подсчёта числа потомков каждой вершины дерева. Оцените его сложность.
4. Модифицируйте алгоритм лексикографической сортировки, так чтобы он работал для слов разной длины.
5. Найдите оптимальное глобальное выравнивание для слов «МОРОЗ» и «ТОРМОЗ». Пусть функция веса сравнения двух символов $t(x, y)$ задана следующими равенствами: $t(x, x) = 1$, $t(-, x) = t(x, -) = -1$, $t(x, y) = -2$ ($x \neq y$), $x, y \in \{З, М, О, Р, Т\}$.
6. Упорядочить последовательность 13, 61, 2, 14, 41, 3, 8, 32, 11, применяя алгоритм сортировки слиянием. Сколько сравнений производится в процессе сортировки?

За решение каждой из задач 1, 2 выставляется максимум 4 балла. За решение каждой из задач 3–6 выставляется максимум 3 балла.

Расчетно-графическая работа 1. Темы: динамическое программирование, сортировки, поиск. Пример задания:

Напишите программу, которая находит оптимальное глобальное выравнивание

двух слов x и y за время $O(|x| \cdot |y|)$.

За решение задачи выставляется максимум 10 баллов.

Контрольная работа 2. Темы: структуры данных для работы с множествами.

Пример задания:

1. Пусть для представления сцепляемых очередей использованы 2-3 дерева (операции: ВСТАВИТЬ(S, a), УДАЛИТЬ(S, a), МИНИМУМ(S), СЦЕПИТЬ(T_1, T_2), РАСЦЕПИТЬ(a, S)). Предложите алгоритм, который реализует операцию вставки одной очереди внутрь другой: ВСТАВИТЬ(S_1, S_2, a)= S , где $S_1 = \{ a_1 < a_2 < \dots < a_i < a_{i+1} < \dots < a_n \}$, $S_2 = \{ b_1 < \dots < b_m \}$, $a = a_i$, $a_i < b_1$, $b_m < a_{i+1}$. Время работы алгоритма должно быть $O(\max\{\log_2 |S_1|, \log_2 |S_2|\})$.
2. Бинарное дерево называется Н-деревом, если все его листья находятся на одинаковой глубине и каждая вершина с одним сыном имеет правого брата с двумя сыновьями. Докажите, что Н-дерево высоты h имеет не менее $(3/2)^h$ листьев.
3. Элемент x массива $T[1 \dots n]$ называется большинством, если $|\{ i : T[i] = x \}| > n / 2$. Предложите алгоритм, который определяет, существует ли большинство, и находит его за время $O(n)$. Разрешается использовать только операции $=$ и \neq .

4. Найдите оптимальное дерево двоичного поиска для элементов a, b, c, d с вероятностями $p_i = 0,1; 0,3; 0,1; 0,2$ и $q_i = 0,1; 0; 0,1; 0; 0,1$.
 5. Выполните операцию ВСТАВИТЬ в точечное квадродерево для следующей последовательности точек (X_i, Y_i) : $(5, 3), (2, 0), (10, 5), (13, 10), (8, 4), (1, 12), (3, 6), (2, 4), (7, 7), (6, 11)$. Затем УДАЛИТЬ вершины с координатами $(5, 3)$ и $(8, 4)$. Нарисуйте точки на плоскости и разбиение плоскости на четверти после всех вставок и после каждого удаления.
- За решение каждой задачи выставляется максимум 4 балла.

Общая сумма В сумме за все задачи выставляет не более 60 баллов.
За ответ на экзамене выставляется максимум 40 баллов.

Требования к рейтинг контролю (7 семестр)

Контрольная работа 1. Темы: алгоритмы на графах. Пример задания:

1. Предложите эффективный алгоритм для построения «минимального» остовного дерева для неориентированного взвешенного графа $G = (V, E)$ с весовой функцией $c: E \rightarrow R$, если «весом» дерева $S = (V, T)$ считать вес самого тяжёлого ребра S , $c(S) = \max\{c(e): e \in T\}$.
2. Пусть $G = (V, E)$ — это ориентированный взвешенный граф с неотрицательной весовой функцией c . Пусть v и w — две вершины из V , а $k < |V|$ — целое число. Предложите алгоритм, который находит кратчайший путь из v в w , содержащий ровно k рёбер. (Замечание: этот путь не обязательно простой.)
3. Пусть обобщённая сеть — это сеть, для каждой дуги (u, v) которой указана как верхняя $c(u, v)$, так и нижняя $b(u, v)$ граница величины потока. Показать, что задачу о максимальном потоке в обобщённой сети можно свести к задаче о максимальном потоке в «обычной» сети. (Рассмотрите случай, когда для каждой v сумма $b(u, v)$ по входящим в v дугам равна сумме $b(v, w)$ по выходящим из v дугам.)
4. Докажите корректность алгоритма ССК выделения сильно связанных компонент ориентированного графа $G = (V, E)$ и оцените его сложность. (Указание: докажите, что если две вершины находятся в одной сильно связанной компоненте, то они окажутся в одном глубинном остовном дереве в графе G' с «перевёрнутыми» рёбрами).
5. Постройте вспомогательную сеть и тупиковый поток в ней для следующей сети (формат: $(v, u, c(v, u), f(v, u))$): $(s, v1, 7, 6), (s, v4, 3, 2), (s, v6, 3, 2), (v1, v2, 7, 6), (v2, v3, 8, 4), (v2, v4, 2, 1), (v2, v6, 3, 1), (v2, v5, 4, 3), (v3, v5, 3, 1), (v3, t, 4, 3), (v5, t, 6, 5), (v5, v7, 1, 1), (v7, t, 2, 1)$.
6. Найдите все двусвязные компоненты заданного неориентированного графа $G = (V, E)$: $V = \{v1, v2, v3, v4, v6, v7, v8, v9, v10, v11\}$, $E = \{(v1, v2), (v1, v4), (v1, v8), (v7, v8), (v2, v9), (v9, v11), (v4, v2), (v3, v8), (v6, v3), (v3, v7), (v6, v7), (v10, v9), (v10, v11)\}$.

За решение каждой из задач 1–4 выставляется максимум 6 баллов. За решение каждой из задач 5, 6 выставляется максимум 3 балла.

Контрольная работа 2. Темы: алгоритмы на матрицах, алгоритмы на строках, NP-полные задачи. Пример задания:

1. В доказательстве теоремы о связи задач умножения матриц и транзитивного замыкания графов матрица инцидентности графа с $n = 2k$ вершинами X разбивалась на 4 подматрицы:

$$X = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

Пусть матрица транзитивного замыкания

$$T(X) = \begin{pmatrix} E & F \\ G & H \end{pmatrix}$$

Покажите, что $H = T(D) + T(D) \times C \times E \times B \times T(D)$, где $T(D)$ — транзитивное замыкание матрицы D .

2. Обобщите алгоритм Рабина-Карпа на случай, когда надо искать квадрат A размером $m \times m$ в матрице S размером $n \times n$ (элементами квадрата и матрицы являются символы алфавита $\Sigma = \{a_1, \dots, a_r\}$).
3. Пусть задано циклическое слово $S[1..n]$, в котором за каждым символом $S(i)$ следует символ $S(i + 1 \bmod n)$. Задача состоит в том, чтобы выбрать место разрезания этого слова i так, чтобы получившееся линейное слово $S^i = S(i) \dots S(n)S(1) \dots S(i - 1)$ было лексикографически минимальным среди всех таких линейных слов. Предложите алгоритм линейной сложности для линейризации циклических слов.
4. Докажите NP-полноту задачи ГАМИЛЬТОНОВО ПОПОЛНЕНИЕ.
Вход: неориентированный граф $G = (V, E)$, число $k > 0$.
Вопрос: можно ли к E добавить k рёбер, так чтобы в графе появился гамильтонов цикл?
(Указание: к этой задаче сводится задача ГАМИЛЬТОНОВ_ЦИКЛ.)
5. Постройте с помощью алгоритма Укконена суффиксное дерево для слова $x = ababaab\$$.
6. Пусть требуется уложить в контейнеры единичного объёма $30m$ предметов s
7. весами:

$$s(A_i) = 0.5 + \varepsilon \quad (1 \leq i \leq 6m)$$

$$s(A_i) = 0.25 + 2\varepsilon \quad (6m < i \leq 12m)$$

$$s(A_i) = 0.25 + \varepsilon \quad (12m < i \leq 18m)$$

$$s(A_i) = 0.25 - 2\varepsilon \quad (18m < i \leq 30m) \quad (\varepsilon > 0).$$

Сколько контейнеров потребуется алгоритму, основанному на принципе «в первый подходящий в порядке убывания»? Покажите, что при оптимальной укладке достаточно $9m$ контейнеров и оцените ошибку приближённого алгоритма.

За решение каждой из задач 1–4 выставляется максимум 6 баллов. За решение каждой из задач 5, 6 выставляется максимум 3 балла.

Важной составляющей данного раздела РПД являются требования к рейтинг-контролю с указанием баллов, распределенных между модулями и видами работы обучающихся.

Максимальная сумма баллов по учебной дисциплине, заканчивающейся экзаменом, по итогам семестра составляет 60 баллов (30 баллов - 1-й модуль и 30 баллов - 2-й модуль).

Обучающемуся, набравшему 40–54 балла, при подведении итогов семестра (на последнем занятии по дисциплине) в рейтинговой ведомости учета успеваемости и зачетной книжке может быть выставлена оценка «удовлетворительно».

Обучающемуся, набравшему 55–57 баллов, при подведении итогов семестра (на последнем занятии по дисциплине) в графе рейтинговой ведомости учета успеваемости «Премиальные баллы» может быть добавлено 15 баллов и выставлена экзаменационная оценка «хорошо».

Обучающемуся, набравшему 58–60 баллов, при подведении итогов семестра (на последнем занятии по дисциплине) в графе рейтинговой ведомости учета успеваемости «Премиальные баллы» может быть добавлено 27 баллов и выставлена экзаменационная оценка «отлично». В каких-либо иных случаях добавление премиальных баллов не допускается.

Обучающийся, набравший до 39 баллов включительно, сдает экзамен.

Распределение баллов по модулям устанавливается преподавателем и может корректироваться.

VII. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для аудиторной работы

Наименование помещений	Материально-техническое оснащение помещений
Ауд. 310 (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Набор учебной мебели.
Ауд. 3л (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Набор учебной мебели, экран, компьютер, проектор, МФУ.

Для самостоятельной работы

Наименование помещений	Материально-техническое оснащение помещений
Ауд. 201а (компьютерная лаборатория ПМиК) (170002, Тверская обл., г. Тверь, пер. Садовый, д. 35)	Набор учебной мебели, доска маркерная, компьютер, сервер (системный блок), концентратор сетевой.

VIII. Сведения об обновлении рабочей программы дисциплины

№ п/п	Обновленный раздел рабочей программы дисциплины	Описание внесённых изменений	Дата и протокол заседания кафедры, утвердившего изменения
1			
2			
3			
4			