

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Смирнов Сергей Николаевич  
Должность: врио ректора  
Дата подписания: 06.10.2023 11:35:14  
Уникальный программный ключ:  
69e375c64f7e975d4e8830e7b4fcc2ad1bf35f08

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



Утверждаю:  
Руководитель ООП  
*А.В. Язенин* / А.В. Язенин /  
«13» *февраля* 2020 года

**Рабочая программа дисциплины (с аннотацией)**

## **МЕТОДЫ ПРОГРАММИРОВАНИЯ**

Направление подготовки  
02.03.02 ФУНДАМЕНТАЛЬНАЯ ИНФОРМАТИКА  
И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Профиль подготовки  
Инженерия программного обеспечения

Для студентов 1-го курса  
Форма обучения – очная

Составитель:

к.ф.-м.н., доцент И.С.Солдатенко

Тверь, 2020

## **I. Аннотация**

### **1. Цель и задачи дисциплины**

Целью освоения дисциплины является:

Изучение общих принципов разработки программного обеспечения, основных эффективных алгоритмов обработки и хранения информации, а также эффективных структур данных.

Задачами освоения дисциплины являются:

- освоение основных способов представления, хранения и обработки информации в компьютерных программах;
- изучение и получение навыков применения основных эффективных алгоритмов и структур данных.

### **2. Место дисциплины в структуре ООП**

Дисциплина относится к части учебного плана, формируемой участниками образовательных отношений, раздел «Дисциплины профиля подготовки».

**Предварительные знания и навыки:**

Для успешного освоения дисциплины необходимы навыки программирования на структурированном языке программирования, а также основные сведения, полученные в ходе изучения дисциплины «Теоретические основы информатики».

**Дальнейшее использование:**

Полученные знания используются в последующем при изучении предметов: «Алгоритмы и анализ сложности», «Технологии программирования», а также во всех дисциплинах, где требуются знания и навыки разработки программного обеспечения с использованием эффективных алгоритмов и структур данных. Знания, умения и навыки, полученные при изучении дисциплины, закрепляются во время лабораторных занятий на дисциплине «Практикум на ЭВМ».

### **3. Объем дисциплины:**

4 зачетных единицы, 144 академических часа, в том числе:

- **контактная аудиторная работа:**  
лекции 48 часов;
- **контактная внеаудиторная работа:** контроль самостоятельной работы и курсовая работа не предусмотрены учебным планом;
- **самостоятельная работа:**  
96 часов, в том числе контроль 32 часа.

### **4. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы**

Планируемые результаты освоения образовательной программы (формируемые компетенции)	Планируемые результаты обучения по дисциплине
---	---

<i>Указывается код и наименование компетенции</i>	<i>Приводятся индикаторы достижения компетенции в соответствии с учебным планом</i>
ПК-2 Способен проводить под научным руководством локальные исследования на основе существующих методов в конкретной области профессиональной деятельности; представлять результаты собственных научных исследований	<p>ПК-2.1 Проводит анализ состояния разработок по теме исследуемой задачи, осуществляет формальную постановку исследуемой задачи</p> <p>ПК-2.2 Решает научные задачи в связи с поставленной целью и в соответствии с выбранной методикой</p> <p>ПК-2.3 Представляет результаты собственных исследований, ведет корректную дискуссию в профессиональной деятельности</p>
ПК-3 Способен собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности; разрабатывать новые алгоритмические, методические и технологические решения в конкретной сфере профессиональной деятельности	<p>ПК-3.1 Знает основы проектирования и элементы архитектурных решений информационных систем</p> <p>ПК-3.2 Применяет в практической деятельности профессиональные стандарты в области информационных технологий, осуществляет алгоритмизацию методов решения прикладных задач</p> <p>ПК-3.3 Имеет практический опыт составления технического задания на разработку информационной системы</p>

**5. Форма промежуточной аттестации и семестр прохождения:**  
экзамен и РГР во 2-м семестре.

**6. Язык преподавания русский.**

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

**1. Для студентов очной формы обучения**

Учебная программа – наименование разделов и тем	Всего (час.)	Контактная работа (час.)			Самостоятельная работа, в том числе контроль (час.)
		Лекции		Контроль самостоятельной работы (в том числе курсовая работа)	
		всего	в т.ч. практическая подготовка		
<p><b>1. Алгоритмы и языки программирования</b></p> <p>ЯП, программа, алгоритм, исполнитель, интерпретация и компиляция. Алгоритмы: определение, свойства, различные стратегии решения задач (сверху-вниз, снизу-вверх). ЯП: история возникновения и развития, уровни языков, формальные и естественные языки, интерпретация, компиляция, трансляция в промежуточный байт-код, виртуальные машины.</p>	6	2	0	0	4
<p><b>2. Эффективность алгоритмов</b></p> <p>Эффективность алгоритмов. Порядок роста. Асимптотическая оценка. «Плохие» алгоритмы. Экспоненциальные алгоритмы, примеры решения задачи Коммивояжера и дискретной упаковки рюкзака, подсчет сложности. Классы NP-трудных и NP-полных задач.</p>	30	10	0	0	20

<p><b>3. Задачи поиска и сортировки</b></p> <p>Задача поиска. Линейный поиск, максимальные и средние оценки времени для линейного поиска. Двоичный поиск и его разновидности. Оценки времени для двоичного поиска. Поиск методом Фибоначчи, золотого сечения. Интерполяционный поиск и его обобщения.</p> <p>Задача сортировки. Инверсии. Сортировка пузырьком. Временные оценки пузырьковой сортировки. Сортировка подсчетом, цифровая сортировка.</p> <p>Элементарные сортировки: сортировка выбором, сортировка вставками, сортировка Шелла. Задача выбора последовательности шагов. Оценки для различных последовательностей</p> <p>Слияние упорядоченных массивов. Сортировка слиянием. Оценка времени. Быстрая сортировка. Оценка времени.</p>	30	10	0	0	20
<p><b>4. Линейные структуры данных</b></p> <p>Структуры данных. Динамические и статические структуры. Линейные структуры с последовательным и произвольным доступом. Принципы работы со статическими одномерными и многомерными структурами данных.</p> <p>Линейные динамические списки. Односвязные и двусвязные списки. Основные операции: добавление, удаление, сцепление и расцепление списков.</p> <p>Кольцевые списки. Двухмерные списки. Представление матриц с помощью двумерных списков. Стеки, очереди, деки, матрицы.</p>	18	6	0	0	12

<p><b>5. Графы и алгоритмы на графах</b></p> <p>Графы. Основные понятия и определения. Способы представления графов. Обход в ширину, глубину.</p> <p>Алгоритмы на графах. Алгоритмы Дейкстры, Флойда-Уоршелла. Построение минимального остовного дерева: Крускал и Прим.</p> <p>Циклы. Нахождение компонент связности, топологическая сортировка. Поиск циклов.</p>	24	8	0	0	16
<p><b>6. Деревья и алгоритмы на деревьях</b></p> <p>Деревья. Способы отображения деревьев. Двоичные (бинарные) деревья. Структура бинарного дерева. Бинарные деревья, представляемые массивами.</p> <p>Система непересекающихся множеств. Представление, основные операции: создание, объединение, поиск. Алгоритм сокращения путей.</p> <p>Двоичные деревья поиска. Поиск по дереву, алгоритмы обхода дерева, копирование и удаление деревьев, удаление из дерева. Турнирная сортировка. Оптимальные деревья поиска.</p> <p>Сбалансированные деревья. Основные определения. Узлы AVL-дерева. Включение в сбалансированное дерево. Повороты. Удаление из сбалансированного дерева.</p>	30	10	0	0	20
<p><b>7. Пирамиды</b></p> <p>Пирамиды (кучи, heap). Преобразование массива в пирамиду. Включение элемента в пирамиду. Удаление из пирамиды. Пирамидальная сортировка.</p>	6	2	0	0	4
<p><b>ИТОГО</b></p>	144	48	0	0	96

### III. Образовательные технологии

Учебная программа – наименование разделов и тем	Вид занятия	Образовательные технологии
1. Алгоритмы и языки программирования	• лекция	<ul style="list-style-type: none"> <li>• традиционные (фронтальная лекция, решение упражнений),</li> <li>• цифровые (показ презентаций, выполнение компьютерных лабораторных работ, расчетно-графической работы),</li> <li>• технология проблемного обучения,</li> <li>• групповая работа</li> </ul>
2. Эффективность алгоритмов		
3. Задачи поиска и сортировки		
4. Линейные структуры данных		
5. Графы и алгоритмы на графах		
6. Деревья и алгоритмы на деревьях		
7. Пирамиды		

### IV. Оценочные материалы для проведения текущей и промежуточной аттестации

№	Результат (индикатор)	Примерная формулировка заданий	Вид/способ	Критерии оценивания
<b>МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕЙ АТТЕСТАЦИИ</b>				
1	ПК-2.1 ПК-2.2 ПК-2.3 ПК-3.2	<b>Расчетно-графическая работа «Анализ эффективности сортировок»</b> Работа делается в юпитеровском ноутбуке с уже готовым сценарием выполнения. Цель работы – провести анализ эффективности различных сортировок, построить соответствующие графики.	<b>вид:</b> самостоятельная лабораторная работа <b>способ:</b> на компьютере <b>результаты:</b> рабочая книга Jupyter Notebook.	<b>Максимум – 10 б.</b>  Работа выполняется пошагово. Каждый шаг оценивается в 0.2-0.5 баллов. Один шаг может заключаться в написании недостающей части алгоритма, подборе параметров, ответе на устный вопрос.
2	ПК-2.3 ПК-3.2	<b>Домашнее задание №1 «Построение и анализ сложности алгоритма»</b> Задание является творческим и посвящено разработке собственного алгоритма для решения поставленной задачи, причем может быть заранее указана максимальная временная сложность алгоритма или необходимость использования метода "Разделяй и властвуй".	<b>вид:</b> самостоятельная творческая исследовательская работа <b>способ:</b> на компьютере <b>результаты:</b>	<b>Максимум – 8 б.</b>  <b>Критерии оценки:</b> • разработан алгоритм, доказана его временная сложность, выполнены условия задачи – макс. балл,

		<p><b>Примеры задач:</b></p> <ol style="list-style-type: none"> <li>Вам дан неупорядоченный массив размера <math>n</math>, все элементы которого различны, причем <math>n</math> является степенью 2. Придумайте алгоритм, который находит второй максимум (второй по порядку элемент, если массив отсортировать по убыванию), и который использует не более <math>n + \log n - 2</math> сравнений.</li> <li>Вам дан унимодальный массив из <math>n</math> различных элементов. Унимодальность означает, что значения массива сначала идут в возрастающем порядке вплоть до максимального элемента, а затем идут в убывающем порядке. Придумайте алгоритм, который находит максимум в этом массиве за время <math>O(\log n)</math>.</li> </ol>	рабочая книга Jupyter Notebook.	<ul style="list-style-type: none"> <li>разработан алгоритм, не отвечающий условиям задачи – 2 балла,</li> <li>разработан алгоритм, отвечающий условиям задачи, но нет формального доказательства оценки сложности – 4 балла.</li> </ul> <p>Если задание выполнено, но не оформлено в среде Jupyter Notebook, то снимается 2 балла.</p>
3	<p>ПК-2.1 ПК-2.2 ПК-2.3 ПК-3.1 ПК-3.2 ПК-3.3</p>	<p><b>Домашнее задание №2 «Проектное задание»</b></p> <p>Задание является групповой творческой исследовательской работой. Для выполнения дается описание проблемы, файлы с тестовыми данными, если того требует задачи. Необходимо разработать программу и с ее помощью провести исследование проблемы, сделать выводы и оформить полученные результаты в виде отчета.</p> <p><b>Пример задачи:</b></p> <p>Необходимо провести анализ устойчивости компьютерной сети, в то время как она подвергается кибератакам. В частности, надо промоделировать ситуацию, когда все возрастающее количество серверов сети выводится из строя. С математической точки зрения компьютерная сеть моделируется с помощью неориентированного графа и во время очередного эксперимента из него поочередно удаляются узлы. В ходе экспериментов следует определять устойчивость сети как функцию, отражающую зависимость размера наибольшей компоненты связности от количества удаленных узлов.</p>	<p><b>вид:</b> самостоятельная групповая исследовательская работа</p> <p><b>способ:</b> на компьютере</p> <p><b>результаты:</b> рабочая книга Jupyter Notebook.</p>	<p><b>Максимум – 12 б.</b></p> <p><b>Критерии оценки:</b></p> <ul style="list-style-type: none"> <li>разработан алгоритм, с его помощью произведено исследование проблемы, сделаны выводы, оформлен отчет – макс. балл,</li> <li>разработан алгоритм, однако исследование произведено некорректно, выводы неправильные – 2 балла,</li> <li>разработан алгоритм, произведено исследование проблемы, сделаны корректные выводы, однако отчет не оформлен – 8 баллов.</li> </ul> <p>Если задание выполнено не в группе, а индивидуально, то полученная сумма баллов делится на два.</p>
4	<p>ПК-3.1 ПК-3.2</p>	<p><b>Домашнее задание №3 «Эффективные алгоритмы»</b></p> <p>Запрограммировать следующие эффективные алгоритмы:</p>	<p><b>вид:</b> самостоятельная лабораторная работа</p>	<p><b>Максимум – 6 б.</b></p> <p>Каждый алгоритм оценивается в 1.5 балла.</p>

		<ul style="list-style-type: none"> <li>• Пирамидальная сортировка</li> <li>• Алгоритм Кана</li> <li>• Алгоритм Тарьяна</li> <li>• Алгоритм Кюрасао</li> </ul> <p>В код каждого алгоритма включить функцию протоколирования (вывода на экран) всех промежуточных состояний.</p>	<p><b>способ:</b> на компью- тере</p> <p><b>результаты:</b> выполненные уроки на платформе Stepik.</p>	<p>Проверка осуществляется с помощью тестов. Если код правильно отработал на всех тестах, то ставится 1.5 балла. Если ошибка хотя бы на одном тесте – 0 баллов.</p>
5	ПК-3.1	<p><b>Модульная контрольная работа 1</b></p> <p>Пример задания приведен в разделе VI.</p>	<p><b>вид:</b> кон- трольная ра- бота</p> <p><b>способ:</b> письменно</p> <p><b>результаты:</b> выполненные задания</p>	<p><b>Максимум – 12 б.</b></p> <p>Контрольная состоит из 18 заданий. Каждое задание оценивается в 0.4-0.6 баллов.</p> <p><b>Критерии оценки:</b> Корректно выполненное задание – балл выставляется полностью.</p>
6	ПК-3.1	<p><b>Модульная контрольная работа 2</b></p> <p>Пример задания приведен в разделе VI.</p>		<p><b>Максимум – 12 б.</b></p> <p>Контрольная состоит из 17 заданий. Каждое задание оценивается в 0.4-0.6 баллов.</p> <p><b>Критерии оценки:</b> Корректно выполненное задание – балл выставляется полностью.</p>
<b>МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ</b>				
7	ПК-3.1 ПК-3.2	<p>Программа экзамена приведена в разделе VI.</p> <p><b>Пример вопроса:</b> Доказать по индукции корректность сортировки выбором (по желанию – устойчивой или неустойчивой). Привести и обосновать оценку сложности относительно операций сравнения и записи (отдельно) для устойчивой и неустойчивой версий алгоритма.</p>	<p><b>вид:</b> традиционный экзамен</p> <p><b>способ:</b> устно</p> <p><b>результаты:</b> устные ответы</p>	<p><b>Максимум – 40 б.</b></p> <p>4 вопроса по 10 баллов каждый.</p> <p><b>Критерии оценки:</b> - ответ дан полностью и корректно – максимальный балл; - ответ дан полностью, допущены 1-2 ошибки – 6 баллов; - ответ дан частично, но корректно – 4 балла.</p>

**Шкала оценивания выполнения индикаторов:**

Индикатор считается выполненным, если либо во время текущей, либо промежуточной аттестации студент набрал как минимум пороговое количество баллов за те виды активности, которые отвечают за данный индикатор. Типовые оценочные материалы с привязкой к отдельным индикаторам приведены в таблице выше.

№	Индикатор	Текущая аттестация		Промежуточная аттестация (экзамен)	
		Порог	Максимум	Порог	Максимум
1	ПК-2.1	9	22	–	–
2	ПК-2.2	9	22	–	–
3	ПК-2.3	12	30	–	–
4	ПК-3.1	17	42	40	100
5	ПК-3.2	14	36	40	100
6	ПК-3.3	5	12	–	–

### **Шкала и критерии выставления оценок за дисциплину:**

Шкала и критерии выставления оценок «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно» описаны в локальной нормативной документации Тверского государственного университета (Положение о рейтинговой системе обучения студентов ТвГУ). Положительная оценка может быть выставлена только в том случае, если выполнены все индикаторы.

### **V. Учебно-методическое и информационное обеспечение дисциплины**

#### 1) Рекомендуемая литература

##### а) Основная литература:

1. Самуйлов С.В. Алгоритмы и структуры обработки данных [Электронный ресурс]: учебное пособие/ Самуйлов С.В.— Электрон. текстовые данные. — Саратов: Вузовское образование, 2016. — 132 с.— Режим доступа: <http://www.iprbookshop.ru/47275.html>
2. Костюкова Н.И. Комбинаторные алгоритмы для программистов [Электронный ресурс]/ Костюкова Н.И.— Электрон. текстовые данные. — М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 216 с. — Режим доступа: <http://biblioclub.ru/index.php?page=book&id=429067>
3. Методы программирования: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, Ю.В. Кулаков и др.; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов: Издательство ФГБОУ ВПО «ТГТУ», 2012. - 144 с.: табл., схем. - Библиогр. в кн. -

ISBN 978-5-8265-1076-6; То же [Электронный ресурс]. — Режим доступа: <http://biblioclub.ru/index.php?page=book&id=437089>

б) Дополнительная литература:

1. Дехтярь М.И. Алгоритмические задачи на графах. Тверь: Тверской государственный университет, 2014. Режим доступа: <http://texts.lib.tversu.ru/texts/09311ucheb.pdf>
2. Комлева, Н.В. Структуры и алгоритмы компьютерной обработки данных: руководство по дисциплине, практикум, тесты, учебная программа: учебное пособие / Н.В. Комлева; Международный консорциум «Электронный университет», Московский государственный университет экономики, статистики и информатики, Евразийский открытый институт. - М.: Московский государственный университет экономики, статистики и информатики, 2004. - 140 с.: ил., табл., схем. - ISBN 5-7764-0400-2 ; То же [Электронный ресурс]. — Режим доступа: <http://biblioclub.ru/index.php?page=book&id=93226>
3. Ковалевская Е.В. Методы программирования [Электронный ресурс]: учебное пособие/ Ковалевская Е.В., Комлева Н.В.– Электрон. текстовые данные. – М.: Евразийский открытый институт, 2011. – 320 с.– Режим доступа: <http://www.iprbookshop.ru/10784>

2) Программное обеспечение

а) Лицензионное программное обеспечение:

- Microsoft Office профессиональный плюс 2013
- Microsoft Windows 10 Enterprise

б) Свободно распространяемое программное обеспечение

- Google Chrome
- Python 3
- Jupyter Notebook

3) Современные профессиональные базы данных и информационные справочные системы

- ЭБС «ZNANIUM.COM» [www.znanium.com](http://www.znanium.com);
- ЭБС «Университетская библиотека онлайн» <https://biblioclub.ru/>;
- ЭБС «Лань» <http://e.lanbook.com>

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

- Электронная образовательная среда ТвГУ <http://lms.tversu.ru>
- Научная библиотека ТвГУ <http://library.tversu.ru>

## VI. Методические материалы для обучающихся по освоению дисциплины

### 1. Структура рейтинговых баллов

Название работы	Кол-во баллов
ТЕКУЩАЯ АТТЕСТАЦИЯ	
<b>Первый модуль</b>	
Расчетно-графическая работа «Анализ эффективности сортировок»	10
Домашнее задание №1 «Построение и анализ сложности алгоритма»	8
Модульная контрольная 1	12
<b>ИТОГО за первый модуль</b>	<b>30</b>
<b>Второй модуль</b>	
Домашнее задание №2 «Проектное задание»	12
Домашнее задание №3 «Эффективные алгоритмы»	6
Модульная контрольная 2	12
<b>ИТОГО за второй модуль</b>	<b>30</b>
ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ	
Экзамен	40

## 2. Примеры заданий для модульных контрольных

### 2.1 Пример первой модульной контрольной

- Расположите следующие функции в порядке увеличения скорости роста (каждая функция есть  $O$ (следующая)):  
1)  $\log \log n$  2) 1 3)  $2^{2^n}$  4)  $n^2 \log n$  5)  $\log(n)$
- Укажите наименьшую верхнюю асимптотическую границу для функции  $f(n) = \frac{n}{1+n}$
- Пусть  $T(n) = \frac{1}{2}n + 3n$ . Какие утверждения верны?  
1)  $T(n) = O(n)$  2)  $T(n) = \Omega(n)$  3)  $T(n) = \Theta(n^2)$  4)  $T(n) = O(n^3)$
- Какова сложность данного алгоритма в худшем случае? (Нотация  $O$  для краткости опущена)

```
sum = 0
для i = 1 до N*N делай
```

```
для j = 1 до i делай
sum = sum + 1
```

- Подсчитайте количество инверсий в массиве  $A = [1\ 4\ 3\ 0\ 5\ 7\ 2\ 6\ 8\ 9]$ .
- Выведите состояние массива  $A = [9\ 3\ 4\ 5\ 6\ 8\ 0\ 7\ 2\ 1]$  после первых трех проходов сортировки пузырьком.
- Пусть дан массив чисел  $A = [0\ 4\ 8\ 7\ 2\ 3\ 6\ 5\ 9\ 1]$ . Выведите состояние массива после пятого шага 3-сортировки. Под шагом понимается одна итерация внешнего цикла в соответствии с псевдокодом из лекции.
- Пусть дан массив чисел  $A = [3\ 9\ 0\ 6\ 2\ 7\ 4\ 14\ 15\ 10\ 1\ 13\ 5\ 8\ 11\ 12]$ . В качестве алгоритма сортировки используется простая (неустойчивая) реализация сортировки выбором. Напечатайте состояние, в котором массив  $A$  будет непосредственно после четырех операций «поменять местами». Замечание: предполагается, что операция «поменять местами» для индексов  $i$  и  $j$  выполняется только в том случае, если  $i$  не равно  $j$ . Обратите внимание: количество итераций может отличаться от 4.
- Выведите значения счетчиков  $i$  и  $j$  процедуры слияния (merge) сразу после пятого шага ее работы на двух массивах:  $A = [0\ 3\ 4\ 6]$  и  $B = [1\ 2\ 5\ 7]$ . Замечание: индексация массивов идет с нуля.
- Пусть дан массив чисел  $A = [55\ 47\ 43\ 30\ 55\ 10\ 80\ 32\ 16\ 8\ 97\ 61\ 5]$ . Укажите его состояние непосредственно после седьмого вызова процедуры слияния (merge) в рекурсивной неоптимизированной (без свичинга) версии сортировки слиянием. Можно считать, что при делении массива нечетной длины средний элемент отходит в левую часть.
- Пусть дан массив чисел  $A = [5\ 8\ 6\ 10\ 4\ 13\ 0\ 9\ 7\ 12\ 2\ 1\ 11\ 3]$ . Укажите его состояние после первого прохода естественной сортировки слиянием.
- Пусть дан массив чисел  $A = [4\ 5\ 1\ 2\ 3\ 0\ 7\ 8\ 6]$ . Укажите его состояние после первого прохода нерекурсивной трехпутевой сортировки слиянием.
- Определим глубину рекурсии в QuickSort как максимальное количество последовательных рекурсивных вызовов, происходящих до того момента, когда рекурсия достигнет базового случая. Оно эквивалентно номеру последнего уровня соответствующего дерева рекурсивных вызовов. Обратите внимание, что глубина рекурсии зависит от того, какие опорные элементы выбирались "по пути". Какова минимально и максимально возможная глубина рекурсии QuickSort?
  - минимум -  $O(\log n)$ , максимум -  $O(n)$
  - минимум -  $O(\log n)$ , максимум -  $O(n \log n)$
  - минимум -  $O(1)$ , максимум -  $O(n)$
  - минимум -  $O(\sqrt{n})$ , максимум -  $O(n)$
- Предположим, что сложность алгоритма задается рекуррентным соотношением  $T(n) = 9 \cdot T(n/3) + n^2$ . Используя основную теорему о рекуррентных оценках, укажите асимптотическую оценку времени выполнения данного алгоритма:
  - $O(n \log n)$
  - $O(n^2 \log n)$
  - $O(n^{2.81})$
  - $O(n^2)$
- Для пустой очереди было вперемешку произведено 10 операций enqueue и 10 операций dequeue. Операции enqueue клали в очередь числа от 0 до 9 по порядку, операции dequeue печатали вынимаемое из очереди число. Какие из указанных последовательностей могли быть в результате напечатаны? а) 0 1 2 3 4 5 8 6 9 7 б) 0 1 2 3 4 5 6 7 8 9 в) 0 1 2 3 4 5 6 7 9 8 г) 4 0 9 1 8 6 3 5 7 2
- Пусть дан массив чисел  $A = [0\ 1\ 2\ 4\ 5\ 6\ 7\ 11\ 12\ 19\ 22\ 26\ 27\ 29]$ . Напечатайте элементы, с которыми будет сравнивать поиск Фибоначчи число 3 после вызова функции fibSearch(A, 3).
- Операция dequeue() печатает вынимаемое из очереди число. Укажите через пробел последовательность чисел, которая будет напечатана в результате выполнения следующей программы:

```
enqueue(1) enqueue(2) enqueue(3) enqueue(4) enqueue(5) enqueue(6) dequeue() dequeue()
dequeue() dequeue() enqueue(7) enqueue(8) dequeue() dequeue() enqueue(9)
```

18. Пусть дан массив чисел  $A = [1\ 2\ 4\ 5\ 8\ 12\ 14\ 16\ 19\ 21\ 24\ 26]$ . Напечатайте элементы, с которыми будет сравнивать двоичный поиск число 2 после вызова функции `binSearch(A, 2)`. Замечание: будем считать, что для массивов четной длины поиск берет последний элемент первой половины.

## 2.2 Пример второй модульной контрольной

1. Приведите состояния массива `id` после выполнения следующей последовательности операций объединить: 9-1 1-2 1-5 5-0 3-6 3-7 0-2 6-7 5-9 1-3 в системе непересекающихся множеств, организованной по принципу:
1. Quick-find (с помощью обычного массива),
  2. Quick-union (с помощью леса, хранящегося в массиве) без использования эвристик.

Запись  $p$ - $q$  означает вызов `объединить(p,q)`. Также договоримся, что во время выполнения `объединить(p,q)` в случае (1) в качестве идентификатора объединения выступает `id[p]`, а в случае (2) корнем дерева с результатом объединения будет корень дерева  $p$ .

2. Приведите состояния массива `id` и дополнительного массива (в случае (1) - весов, в случае (2) - рангов) после выполнения следующей последовательности операций объединить: 8-5 8-5 2-0 0-9 5-6 9-1 7-3 2-3 5-8 7-2 в системе непересекающихся множеств, организованной по принципу Quick-union (с помощью леса, хранящегося в массиве) с помощью:
1. весовой эвристики,
  2. эвристика ранга.

Запись  $p$ - $q$  означает вызов `объединить(p,q)`. Также договоримся, что во время выполнения `объединить(p,q)`, если веса (ранги) обоих деревьев совпадают, корнем дерева с результатом объединения будет корень дерева  $p$ . Обратите внимание: имеется в виду, что надо привести отдельно результат работы СНМ с весовой эвристикой и отдельно результат работы СНМ с эвристикой ранга.

3. На изначально пустой мин-куче выполняется последовательность операций: 7 1 6 0 2 - 9 8 - 3 5 - 4 Приведите состояния массива `heap` и нарисуйте соответствующие им деревья после выполнения каждой операции. Запись  $x$  означает вызов операции `добавить(x)`, а символ  $-$  означает вызов операции `взять-минимум`.
4. Для следующей двоичной мин-кучи: 1 38 26 74 60 40 27 90 83 67 65 49 45 37 99 для элемента с ключом 37 выполнили операцию `Изменить-приоритет(37, 2)`. Приведите состояние массива после выполнения указанной операции.
5. Для графа №1 найдите минимальный остов с помощью алгоритма Крускала. В ответе перечислите все ребра графа в том порядке, в котором их рассматривал алгоритм, и для каждого ребра укажите вердикт - входит оно в остов или нет.

**Дополнительное задание.** Предположите, что алгоритм Крускала работает с системой непересекающихся множеств и распишите содержимое массива СНМ после каждого добавления ребра в остов (СНМ представлена в самом тривиальном виде - в виде массива меток; использование другого -

даже более сложного представления СНМ - засчитываться не будет; в качестве метки используется минимальная из двух меток).

6. Для графа №1 постройте минимальный остов прямо на картинке и укажите его размер.

**Дополнительное задание.** Напишите ответ в виде массива предков.

7. Для графа №1 найдите минимальный остов с помощью алгоритма Прима. Выведите ребра в том порядке, в котором они будут добавляться в остов. Алгоритм Прима начинает работу с вершины с минимальным номером.

**Дополнительное задание.** Предположите, что алгоритм Прима работает с кучей и нарисуйте состояние кучи (в виде графа) после каждого шага алгоритма (по две картинки: первая - после взятия минимума, вторая - после релаксации всех ребер). *Замечание: обратите внимание, что в отличие от предыдущих упражнений на кучу, здесь вам в каждой вершине кучи необходимо хранить не одно, а два значения: метку вершины и ключ (приоритет).*

8. Для графа №1 укажите минимальное ребро разреза  $X = \{7, 5, 3, 8, 4\}$ .

9. Пусть дан неупорядоченный массив целых чисел: 90 64 73 95 93 66 6 28 75 40 78 22 50 23 35 Мы делаем **первый** проход сортировки кучей - строим по этому массиву мин-кучу (прямо в этом же массиве, используя **линейный** алгоритм). Укажите его состояние после 13-го шага данной процедуры, а также нарисуйте соответствующее массиву двоичное дерево и выделите в нем уже сформированные корректные кучи.

10. Пусть нам дано двоичное дерево поиска: [32 9 71 1 20 52 82 - - - - 33 62 88]. Знак - в массиве означает, что соответствующей вершины в дереве нет. Нарисуйте его, аугментируйте информацией о количестве элементов в каждом поддереве и нарисуйте дерево рекурсивных вызовов функции  $kth(0, 9)$ . *Напоминаем:  $kth(t, n)$  возвращает  $n$ -ю порядковую статистику дерева с корнем в  $t$  (т.е.  $n$ -й по счету элемент в отсортированной последовательности).*

11. Для изначально пустого AVL-дерева выполняется следующая последовательность операций: 11 1 27 34 85 -1 87 73 -73, где положительное число означает добавление данного числа в дерево, а отрицательное - удаление соответствующего числа методом Хиббарда (в данном методе при наличии обоих сыновей у вершины всегда выбирается следующая, а не предыдущая вершина). Нарисуйте состояние дерева после каждой операции (уже после выполнения балансировки).

12. В пустое (обычное) двоичное дерево поиска добавляются элементы в указанном порядке: 95 0 69 32 91 70 17 68 39 59 86 54 92 20 3. Нарисуйте получившееся дерево.

**Дополнительное задание.** Укажите коэффициенты сбалансированности для каждого узла в результирующем дереве.

13. Дана матрица последователей  $next$ , построенная алгоритмом Флойда-Уоршелла:

```
0 7 5 5 7 5 5 7 5 5
4 1 2 4 4 2 2 2 2 9
6 1 2 6 6 6 6 6 6 9
4 1 9 3 4 4 9 9 9 9
0 9 5 3 4 5 5 9 9 9
2 2 2 3 2 5 2 2 9 9
4 7 5 4 4 5 6 7 8 5
4 1 1 4 4 4 1 7 8 4
7 7 7 7 7 6 6 7 8 9
4 1 2 4 4 5 8 8 8 9
```

Напечатайте кратчайший путь из вершины 2 в вершину 0.

14. Для графа №1 продемонстрировать работу алгоритма Дейкстры для вершины 1: напечатайте состояние массива жадных оценок  $d$  после каждого шага алгоритма (помечайте номера тех вершин, которые выбирались на соответствующем шаге).

**Дополнительное задание №1.** Параллельно с массивом  $d$  вычисляйте и печатайте также массив предков (массив дерева кратчайших путей)  $pi$ .

**Дополнительное задание №2.** Предположите, что алгоритм Дейкстры работает с кучей и нарисуйте состояние кучи (в виде графа) после каждого шага алгоритма (по две картинки: первая - после взятия минимума, вторая - после релаксации всех ребер).

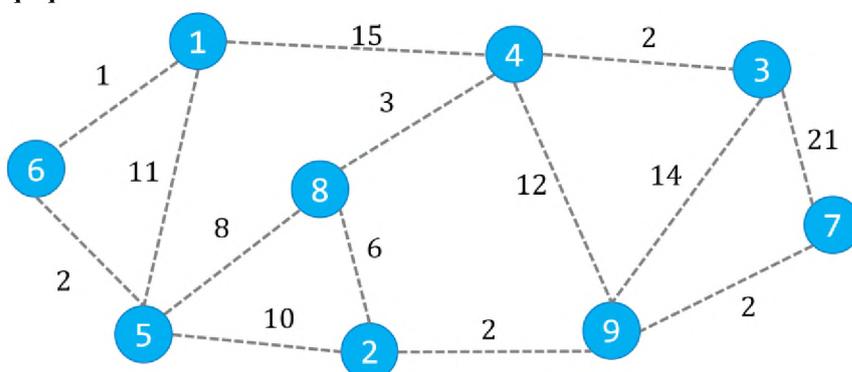
15. Для графа №3 определите количество компонент слабой и сильной связности.

16. Для графа №2 выполните топологическую сортировку алгоритмом Тарьяна и напечатайте получившийся массив времен выхода из вершин  $tout$ . Напечатайте также топологически упорядоченную последовательность вершин, соответствующую данному массиву  $tout$ .

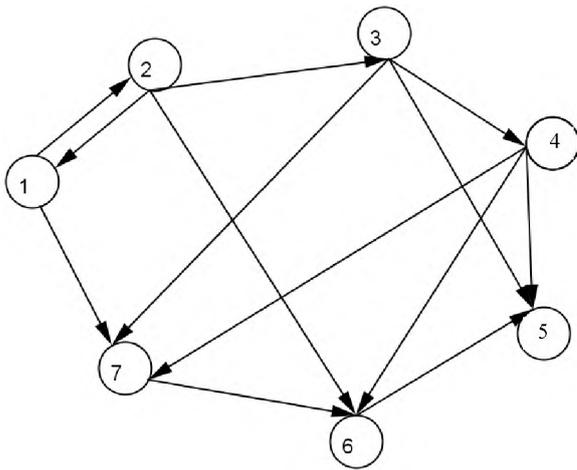
17. Для графа №3 найдите компоненты сильной связности методом Косараю. Для этого:

- Укажите вершину, с которой вы запустите первый обход графа в глубину, а также подпишите рядом с каждой вершиной подсчитанное время выхода из этой вершины.
- Постройте транспонированный граф и перепишите время выхода для каждой вершины.
- В транспонированном графе пометьте те вершины, для которых второй запуск DFS-loop будет инициировать обход DFS (а также отдельно выпишите их номера в порядке запуска). Обведите те компоненты сильной связности, которые обнаружат соответствующие запуски обхода.

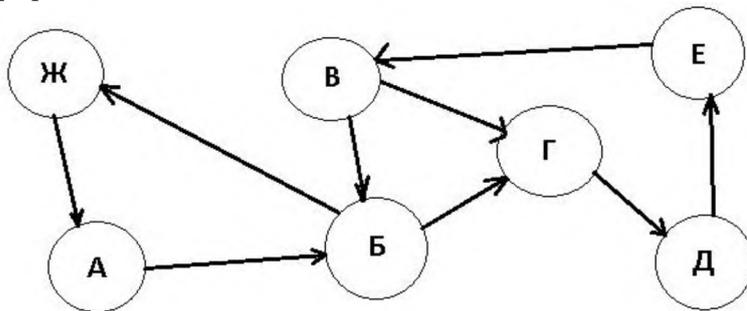
Граф №1



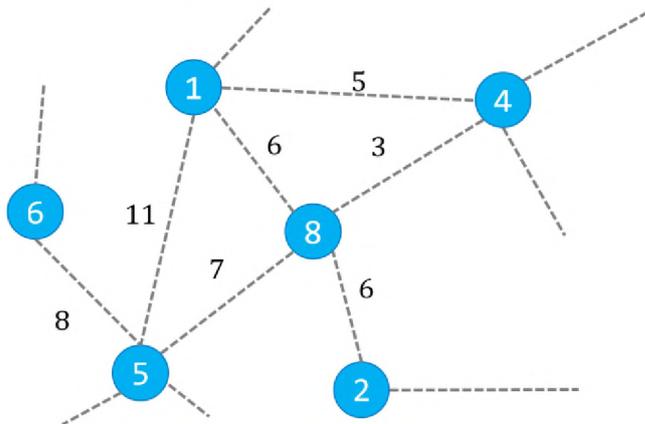
Граф №2



Граф №3



Фрагмента графа №1



### 3. Программа экзамена

#### 3.1 Определения

- Асимптотические нотации:  $O(g(n))$ ,  $\Omega(g(n))$ ,  $\Theta(g(n))$ .
- Инверсия
- Устойчивость сортировки
- Упорядоченность последовательности элементов
- Отношение, отношение порядка: полное, частичное
- Инвариант цикла
- Массив, список. Односвязные, двусвязные, кольцевые списки
- Стек, очередь, дека
- Граф: вершина, ребро, вес ребра, ориентированность, нагруженность, связность, достижимость вершины, расстояние между парой вершин

- Матрица смежности, списки смежных вершин
- Путь/цикл в графе, простой и элементарный путь/цикл, гамильтонов и эйлеров путь/цикл
- Дерево: предок, потомок, корень, лист, высота
- Двоичное дерево
- Обходы деревьев: левый, правый, постфиксный, инфиксный (для бинарного), префиксный
- Подграф, поддерев
- Разрез графа
- Компонента связности, компонента слабой/сильной связности
- Топологическая сортировка
- Система непересекающихся множеств
- Остовное дерево (остов)
- Полное и законченное бинарное дерево
- Пирамида (куча)
- Двоичное дерево поиска
- Конденсация графа, транспонирование графа
- Сбалансированное дерево поиска, AVL-дерево
- Путь в графе, вес пути, вес кратчайшего пути, кратчайший путь. Цикл отрицательно веса

## 3.2 Вопросы

### Часть I

1. Доказать по индукции корректность сортировки выбором (по желанию – устойчивой или неустойчивой). Привести и обосновать оценку сложности относительно операций сравнения и записи (отдельно) для устойчивой и неустойчивой версий алгоритма.
2. Доказать по индукции корректность пузырьковой сортировки (наивной). Привести и обосновать оценку сложности. Доказать методом от обратного, что эффективная версия сортировки пузырьком (с флагом) корректна.
3. Доказать по индукции корректность сортировки вставками. Привести и обосновать оценку времени алгоритма. Определить частичную упорядоченность массива. Доказать, что для частично упорядоченных массивов сортировка вставками работает линейное время.
4. Доказать теорему, что после  $h$ -сортировки вставками  $k$ -упорядоченный массив остается  $k$ -упорядоченным. Используя данную теорему, доказать корректность алгоритма сортировки Шелла. (Кнут, том 3, стр. 102)
5. Доказать по индукции корректность устойчивой сортировки подсчетом (что она сортирует и является устойчивой). Дать оценку времени работы алгоритма.
6. Доказать по индукции корректность цифровой сортировки. Дать оценку времени работы алгоритма.
7. Доказать по индукции корректность процедуры слияния и корректность сортировки слиянием. Привести и обосновать с помощью 1) дерева рекурсивных вызовов, 2) метода раскрытия рекуррентности и 3) по индукции оценку времени работы алгоритма.
8. Доказать, что любая сортировка, основанная на сравнениях, в худшем случае использует  $O(n \log n)$  сравнений. Доказать по индукции корректность 3-х путевой быстрой сортировки.

9. Доказать по индукции корректность алгоритма быстрой сортировки. Доказать оценку сложности алгоритма.
10. Доказать основную теорему о рекуррентных оценках.
11. Доказать по индукции корректность алгоритмов линейного и дихотомического поиска. Дать и обосновать оценки сложности алгоритмов (можно использовать основную теорему о рекуррентных оценках).
12. Доказать по индукции корректность интерполяционного поиска. Используя утверждение, что на каждом шаге интерполяционный поиск уменьшает размер подзадачи с  $n$  до  $\sqrt{n}$ , вывести оценку времени его работы.
13. Доказать по индукции корректность поиска Фибоначчи. Привести и неформально обосновать оценку сложности алгоритма.
14. Динамический массив. Анализ времени работы в неэффективном случае (вставка и удаление элемента после каждой операции) и амортизационный анализ (методом бухгалтерского учета [Кормен, Лейзерсон, Ривест, стр. 492, 502]) в эффективном случае.

## Часть II

15. Описать СНМ и все различные эвристики. Доказать, что в структуре Quick-Union с весовой эвристикой глубина любого узла  $x$  не превосходит  $\log(N)$ , где  $N$  – количество элементов.
16. Описать СНМ и все различные эвристики. Доказать, что в структуре Quick-Union с эвристикой ранга в дереве с корнем в  $p$  количество вершин не меньше  $2^{\text{rank}[p]}$ .
17. Описать бинарную кучу и основные операции. Докажите корректность процедур Всплытие и Взять-минимум в бинарной куче.
18. Квадратичный и эффективный алгоритм Краскала. Приведите и обоснуйте оценку сложности. Квадратичный и эффективный алгоритм Прима. Приведите и обоснуйте оценку сложности.
19. Докажите леммы о пустом разрезе, о двойном пересечении разреза, об одиоком ребре разреза. Докажите лемму о минимальном ребре разреза. Докажите корректность алгоритма Прима (что он строит остов, и что остов является минимальным).
20. Опишите бинарное дерево поиска и его основные операции. Докажите по индукции корректность алгоритмов поиска, удаления Хиббарда, операции floor, prev и взятия  $k$ -той порядковой статистики в бинарном дереве поиска.
21. Опишите AVL-дерево и его основные операции. Докажите корректность процедуры балансировки AVL-дерева.
22. Докажите лемму о кратчайших путях. Докажите корректность алгоритма Дейкстры. Приведите и обоснуйте оценку его сложности.
23. Алгоритм Флойда-Уоршелла. Докажите корректность его работы (приведите и объясните рекуррентные отношения), обоснуйте оценку его сложности.
24. Докажите корректность алгоритма топологической сортировки Кана. Приведите и обоснуйте оценку его сложности.
25. Докажите корректность алгоритма топологической сортировки Тарьяна. Приведите и обоснуйте оценку его сложности.
26. Докажите корректность алгоритма Косараю. Приведите и обоснуйте оценку его сложности.

### 3.3 Чем можно пользоваться

На экзамен можно принести с собой один лист формата А4, на котором можно заранее написать **все, что угодно**. Писать можно на обеих сторонах листа. Главное условие - текст должен быть написан его владельцем **самостоятельно, синей ручкой и только от руки**. Ксерокопии, распечатки, копии, сделанные с помощью копировальной бумаги, и тому подобное использовать нельзя. Преподаватель оставляет за собой право изъять данный лист при малейших подозрениях на чужое авторство без объяснения причин. Данную шпаргалку можно использовать только при подготовке к ответу.

Ничем другим пользоваться нельзя.

### **3.4 Процедура сдачи экзамена**

1. Желающие сдать экзамен заходят в аудиторию, складывают все личные вещи на отдельный стол и рассаживаются отдельно по партам.
2. Все сотовые телефоны и любые другие электронные устройства должны находиться в сумке или на отдельном столе. Любое появление за рабочим столом любого электронного устройства повлечет за собой смену билета или удаление с экзамена.
3. Каждый по отдельности подходит и отвечает на три кратких устных вопроса на знание определений. Если неправильных ответов больше одного, то человек садится на место и повторяет определения.
4. Если ошибок в ответах по базовым определениям не больше одной, то человек тянет четыре вопроса (по 10 баллов каждый).
5. Спустя полтора часа (по желанию - раньше) все отвечают в том же порядке, в котором были взяты билеты.
6. Во время подготовки можно использовать заготовленный ранее лист-шпаргалку (исключительно свою). Во время выступления пользоваться ей нельзя.
7. Для ответа на вопросы можно подготовить на отдельном листе тезисный<sup>1</sup> план ответа.
8. Учить алгоритмы наизусть не нужно. Во время выступления псевдокод вашего алгоритма будет транслироваться на доске (если это предусмотрено вопросом).
9. Все вопросы, связанные с теми или иными алгоритмами/структурами данных, предполагают обязательное описание логики (интуиции) их работы.
10. Выступление с ответом происходит на маркерной доске, на которой вы будете писать все доказательства и обоснования оценок сложности.

## **4. Указания для обучающихся**

Организуя свою учебную работу, студенты должны, во-первых, выявить рекомендуемый режим и характер учебной работы по изучению теоретического курса, практическому применению изученного материала, по выполнению заданий для самостоятельной работы, по использованию информационных технологий и т.д. Во-вторых, ознакомиться с указанным в методическом материале по дисциплине перечнем учебно-методических изданий, рекомендуемых студентам для подготовки к занятиям и выполнения самостоятельной работы, а также с методическими материалами на бумажных и/или электронных носителях, выпущенных кафедрой своими силами и предоставляемые студентам во время занятий.

---

<sup>1</sup> «Тезисный» - означает, что на нем не будет переписано содержимое вашей шпаргалки, а будет «список словосочетаний».

Самостоятельная работа студентов, предусмотренная учебным планом, должна соответствовать более глубокому усвоению изучаемого курса, формировать навыки исследовательской работы и ориентировать студентов на умение применять теоретические знания на практике.

#### *1. Работа с учебными пособиями.*

Для полноценного усвоения курса студент должен, прежде всего, овладеть основными понятиями этой дисциплины. Необходимо усвоить определения и понятия, уметь приводить их точные формулировки, приводить примеры объектов, удовлетворяющих этому определению. Кроме того, необходимо знать круг фактов, связанных с данным понятием. Требуется также знать связи между понятиями, уметь устанавливать соотношения между классами объектов, описываемых различными понятиями.

#### *2. Самостоятельное изучение тем.*

Самостоятельная работа студента является важным видом деятельности, позволяющим хорошо усвоить изучаемый предмет и одним из условий достижения необходимого качества подготовки и профессиональной переподготовки специалистов. Она предполагает самостоятельное изучение студентом рекомендованной учебно-методической литературы, различных справочных материалов, написание рефератов, выступление с докладом, подготовку к лекционным и практическим занятиям, подготовку к зачёту и экзамену.

#### *3. Подготовка к практическим занятиям.*

При подготовке к практическим занятиям студентам рекомендуется следовать методическим рекомендациям по работе с учебными пособиями, приведенным выше.

#### *4. Составление конспектов.*

В конспекте отражены основные понятия темы. Для наглядности и удобства запоминания используются схемы и таблицы.

### **VII. Материально-техническое обеспечение**

Для аудиторной работы

Учебная аудитория № 212 (170002, Тверская обл., г. Тверь, Садовый переулок, д.35)	Набор учебной мебели, меловая доска, мультимедийный комплекс "I - Lerner .ru" в составе: проектор Epson EB -575 Wi, маркерная доска, панель управления Epson ELPCB02, запасная лампа, запасной фильтр для проектора.
---	--

Для самостоятельной работы

<p>Помещение для самостоятельной работы обучающихся:</p> <p>Компьютерный класс факультета прикладной математики и кибернетики № 46 (170002, Тверская обл., г.Тверь, Садовый переулок, д.35)</p>	<p>Персональные ЭВМ (компьютер RAMEC STORM C2D 4600/160Gb/DVD-RW+Монитор LG TFT 17" L1753S-SF silver – 24 шт.), мультимедийный проектор BenQ MP 724 с потолочным креплением и экран 1105, кондиционер General Climate – 2 шт., коммутатор D-Link 10/100/1000mbps 16-portr DGS-1016D, коммутатор D-Link 10/100/1000mbps 16-portr DGS-1016D- 2 шт.</p>
---	--

### VIII. Сведения об обновлении рабочей программы дисциплины

№ п.п.	Обновленный раздел рабочей программы дисциплины	Описание внесенных изменений	Реквизиты документа, утвердившего изменения
1.	3. Объем дисциплины	Выделение часов на практическую подготовку	От 29.10.2020 года, протокол № 3 ученого совета факультета
2.	II. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий	Выделение часов на практическую подготовку по темам	От 29.10.2020 года, протокол № 3 ученого совета факультета