

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИМЕНИ В.Ф. УТКИНА»

Кафедра «Вычислительная и прикладная математика»

«СОГЛАСОВАНО»

Заведующий кафедрой ВПМ  
/ Г.В. Овечкин

10.12 2023 г



«УТВЕРЖДАЮ»

Проректор по учебной работе  
/ А.В. Корячко

10.12 2023 г

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**АНАЛИЗ АЛГОРИТМОВ**

Направление подготовки  
09.03.04 Программная инженерия

Направленность (профиль) подготовки  
Программное обеспечение систем искусственного интеллекта

Квалификация выпускника — бакалавр

Форма обучения — очная

Рязань 2023 г

Программу составил(и):

д.техн.н., проф., Пруцков А. В.

Рабочая программа дисциплины

**Анализ алгоритмов**

разработана в соответствии с ФГОС ВО:

ФГОС ВО - бакалавриат по направлению подготовки 09.03.04 Программная инженерия (приказ Минобрнауки России от 19.09.2017 г. № 920)

составлена на основании учебного плана:

09.03.04 Программная инженерия

утвержденного учёным советом вуза от 27.01.2023 протокол № 6.

Рабочая программа одобрена на заседании кафедры

**Вычислительной и прикладной математики**

Протокол от 29.12.2022 г. № 4

Срок действия программы: 2023-2027 уч.г.

Зав. кафедрой Овечкин Геннадий Владимирович

Общая трудоемкость

**3 ЗЕТ**

**Распределение часов дисциплины по семестрам**

Семестр (<Курс>.<Семестр на курсе>)	6 (3.2)		Итого	
	16			
Неделя	16			
Вид занятий	УП	РП	УП	РП
Лекции	32	32	32	32
Лабораторные	16	16	16	16
Практические	16	16	16	16
Иная контактная работа	0,25	0,25	0,25	0,25
Итого ауд.	64,25	64,25	64,25	64,25
Контактная работа	64,25	64,25	64,25	64,25
Сам. работа	35	35	35	35
Часы на контроль	8,75	8,75	8,75	8,75
Итого	108	108	108	108

---

---

**Визирование РПД для исполнения в очередном учебном году**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2024-2025 учебном году на заседании кафедры  
**Вычислительной и прикладной математики**

Протокол от \_\_\_\_\_ 2024 г. № \_\_\_\_

Зав. кафедрой \_\_\_\_\_

---

---

**Визирование РПД для исполнения в очередном учебном году**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2025-2026 учебном году на заседании кафедры  
**Вычислительной и прикладной математики**

Протокол от \_\_\_\_\_ 2025 г. № \_\_\_\_

Зав. кафедрой \_\_\_\_\_

---

---

**Визирование РПД для исполнения в очередном учебном году**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2026-2027 учебном году на заседании кафедры  
**Вычислительной и прикладной математики**

Протокол от \_\_\_\_\_ 2026 г. № \_\_\_\_

Зав. кафедрой \_\_\_\_\_

---

---

**Визирование РПД для исполнения в очередном учебном году**

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2027-2028 учебном году на заседании кафедры

**Вычислительной и прикладной математики**

Протокол от \_\_\_\_\_ 2027 г. № \_\_\_\_

Зав. кафедрой \_\_\_\_\_

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)						
1.1	Целью освоения дисциплины является получение знаний и навыков в области анализа алгоритмов.					
2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ						
Цикл (раздел) ОП:		Б1.В				
2.1	<b>Требования к предварительной подготовке обучающегося:</b>					
2.1.1	Аналитическая геометрия					
2.1.2	Программирование					
2.1.3	Теоретическая информатика					
2.1.4	Основы программной инженерии					
2.1.5	Типы и структуры данных					
2.2	<b>Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b>					
2.2.1	Защита информации					
2.2.2	Функциональное и логическое программирование					
2.2.3	Моделирование					
3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)						
<b>ПК-3: Способен разрабатывать и тестировать программные компоненты решения задач в системах искусственного интеллекта</b>						
<b>ПК-3.2. Разрабатывает приложения систем искусственного интеллекта</b>						
<b>Знать</b> современные языки программирования, библиотеки и программные платформы для функционального, логического, объектно-ориентированного программирования приложений систем искусственного интеллекта (Python, R, C++, C#)						
<b>Уметь</b> разрабатывать программные приложения систем искусственного интеллекта, с использованием современных языков программирования, библиотек и программных платформ функционального, логического, объектно-ориентированного программирования (Python, R, C++, C#)						
<b>Владеть</b> основными принципами разработки приложений систем искусственного интеллекта						
<b>В результате освоения дисциплины (модуля) обучающийся должен</b>						
3.1	<b>Знать:</b>					
3.1.1	понятие алгоритма, классификацию алгоритмов, методы оптимизации					
3.2	<b>Уметь:</b>					
3.2.1	использовать в работе методы декомпозиции, параллельного программирования					
3.3	<b>Владеть:</b>					
3.3.1	навыками динамического программирования					
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)						
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Форма контроля
	<b>Раздел 1. Анализ алгоритмов</b>					
1.1	Основные понятия анализа алгоритмов. Асимптотические оценки сложности алгоритмов. Методы оценки сложности итеративных алгоритмов. Сложностные классы задач. Теоретические и экспериментальные оценки трудоёмкости и ёмкости по памяти. /Тема/	6	0			
1.2	Понятие алгоритма. Формализм Э.Л. Поста. История теории алгоритмов. Формализация понятия алгоритма. Пространство символов и примитивные операции в машине Поста, алгоритм как финитный 1-процесс, доставляющий решение общей проблемы, гипотеза Поста. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет

1.3	Основная терминология и обозначения в анализе алгоритмов. Вход как эквивалент конкретной проблемы по Посту. Длина входа как функция от мощности множества входных данных. Оценки качества алгоритма. Понятие трудоёмкости. Трудоёмкость в лучшем, худшем и среднем случаях. Ёмкостная эффективность. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.4	Классификация алгоритмов по трудоёмкости. Влияние длины входа и значений на трудоёмкость. Классификация алгоритмов по характеристическим особенностям множества исходных данных, определяющим вид функции трудоёмкости. Классы N, PR, NPR. Примеры алгоритмов в рамках классификации. Дополнительная детализация классов. Особенности анализа алгоритмов по классам. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.5	Анализ NPR алгоритмов методом классов входных данных. Метод классов входных данных. Особенности его применения. Задача сортировки трёх чисел по месту. Классы входных данных, порождаемые этой задачей. Алгоритмы и их анализ методом классов входных данных. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.6	Метод вероятностного анализа для получения трудоёмкости в среднем. Алгоритм сортировки вставками и анализ его трудоёмкости в среднем. Замечания об экспериментальном анализе алгоритмов по трудоёмкости. Выборочное среднее и математическое ожидание. Определение необходимого числа экспериментов. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.7	Сравнительный анализ алгоритмов по трудоёмкости. Методика сравнительного анализа алгоритмов и определение границ применимости по характеристикам исходных данных. Простейшие примеры сравнительного анализа для задачи сортировки — алгоритм сортировки вставками и сортировки методом индексов. Рациональный выбор в параметрическом пространстве. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.8	Сложность алгоритмов. Теоретическая нижняя граница сложности задачи. Асимптотический анализ функций и основные обозначения. Сложность как асимптотическая оценка трудоёмкости в худшем случае. Теоретическая нижняя граница сложности задачи. Доказательство теоретической нижней границы для задачи сортировки сравнениями /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.9	Введение в теорию сложности вычислений. Исторический очерк теории сложности вычислений. Сложностные классы задач. Определение и взаимосвязь классов P и NP. Определение и примеры NP-полных задач. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет

1.10	Алгоритмы нахождения расстояния Левенштейна и расстояния Дамерау-Левенштейна между строками. Метод динамического программирования. Алгоритмы нахождения расстояний Левенштейна и Дамерау-Левенштейна между строками; матричная реализация алгоритмов; рекурсивная реализация одного из алгоритмов. Сравнительный анализ рекурсивной и не рекурсивной реализаций алгоритма нахождения расстояния Левенштейна по временной и емкостной эффективности /Пр/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.11	Оценка трудоемкости алгоритмов. Алгоритмы умножения матриц. Модели вычислений (базовые операции и их стоимость; оценка циклов; оценка условных переходов). Теоретическая оценка трудоемкости на примере стандартного алгоритма умножения матриц. Трудоемкость в лучшем и худшем случае. Эффективный алгоритм Винограда умножения матриц. Подходы к оптимизации алгоритмов. Получение экспериментальной оценки трудоёмкости /Пр/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.12	Оценка трудоемкости не рекурсивных алгоритмов сортировки. Поиск подстроки в строке: алгоритмы различной трудоемкости. Регулярные выражения. Оценка трудоемкости алгоритмов в зависимости от классов входных данных на примере алгоритма сортировки пузырьком. Трудоемкость в лучшем, худшем, произвольном случае. Поиск подстроки в строке: алгоритмы стандартный, Кнута-Морриса-Пратта, Бойера-Мура, регулярные выражения /Пр/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.13	Математические основы параллельных вычислений. Поточковые алгоритмы. Графовые модели алгоритмов. Описание алгоритма четырьмя графовыми моделями: графом управления, информационным графом, операционной историей и информационной историей. Каноническая ярусно-параллельная форма графа алгоритма. Закон Амдала. Поточковые алгоритмы. Поточковые алгоритмы нахождения максимального значения, среднего значения /Пр/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.14	Метод динамического программирования. Линейная и рекурсивная реализации алгоритмов определения расстояния Левенштейна и расстояния Дамерау-Левенштейна между строками и их анализ. /Лаб/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.15	Реализация и анализ традиционного алгоритма умножения матриц, эффективного алгоритма Винограда, оптимизированного алгоритма Винограда. Теоретические и экспериментальные оценки трудоёмкости и ёмкости по памяти. /Лаб/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.16	Реализация и анализ не рекурсивных алгоритмов сортировки. Теоретические и экспериментальные оценки трудоёмкости. /Лаб/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.17	Реализация алгоритмов поиска подстроки в строке. /Лаб/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет

1.18	Основные понятия анализа алгоритмов. Асимптотические оценки сложности алгоритмов. Методы оценки сложности итеративных алгоритмов. Сложностные классы задач. Теоретические и экспериментальные оценки трудоёмкости и ёмкости по памяти. /Ср/	6	19	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.19	Разработка вычислительно эффективных алгоритмов. Метод декомпозиции задачи. Математические основы параллельных вычислений. Основы разработки параллельных алгоритмов, приемы оценки их сложности. Влияние архитектуры вычислительной системы на разработку алгоритмов. Методы оценки сложности рекурсивных алгоритмов. /Тема/	6	0			
1.20	Временная эффективность и особенности перехода к временным оценкам. Переход к пооперационному анализу трудоёмкости алгоритмов. Построение временных оценок. Методы перехода к временным оценкам на основе функции трудоёмкости. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.21	Рекурсивные алгоритмы — Особенности реализации и анализа. Рекурсивные функции и рекурсивные алгоритмы. Рекурсивная реализация алгоритмов. Анализ механизма рекурсивного вызова. Дерево рекурсивных вызовов. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.22	Метод декомпозиции и особенности его применения. Рекурсивные алгоритмы как реализация метода декомпозиции. Методы анализа рекурсивных алгоритмов. Теорема Бенгли-Хакен-Сакса и её применение для получения сложностных оценок алгоритмов, разработанных методом декомпозиции. Анализ рекурсивных алгоритмов методом прямого подсчета вершин рекурсивного дерева. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.23	Анализ рекурсивных алгоритмов методом прямого подсчета вершин рекурсивного дерева. Пример анализа — алгоритм сортировки слиянием. Отличие детального анализа от асимптотического — граница применимости алгоритма сортировки вставками /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.24	Рекурсивный алгоритм возведения в степень. Метод повторного возведения в квадрат. Функция $\square(n)$ . Рекурсивный и итерационный алгоритм быстрого возведения в степень и их анализ с использованием функции $\square_1(n)$ /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.25	Задачи умножения длинных целых чисел и умножения матриц. Задача умножения длинных целых чисел. Гипотеза Колмогорова. Алгоритм умножения в столбик. Алгоритм Карацубы. /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.26	Особенности параллельных вычислений. Переход к параллельным вычислениям. Операционная, информационная зависимость фрагментов алгоритма. Особенности параллельной реализации алгоритмов: синхронизация, разделяемый доступ к ресурсам, обеспечение монопольного доступа к разделяемым ресурсам на запись, соблюдение актуальности данных при параллельном чтении разделяемых ресурсов. /Пр/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет

1.27	Балансировка нагрузки на параллельные вычислительные системы. Особенности обмена данными при различных архитектурных решениях параллельных вычислительных систем. Архитектуры параллельных вычислительных систем: с диспетчером, без диспетчера. Алгоритмы планирования задач, статической и динамической балансировки нагрузки на узлы вычислительной системы при наличии априорной оценки трудоемкости задач и в отсутствие таковой. /Пр/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.28	Муравьиный алгоритм. Эволюционные алгоритмы. Муравьиный алгоритм. Применение муравьиного алгоритма к решению задачи коммивояжера, сравнение с решением задачи полным перебором. Параметризация муравьиного алгоритма, изучение влияния настроечных параметров алгоритма на результирующие значения (в задаче коммивояжера – на длину пути). /Пр/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.29	Графовые модели алгоритмов. Поточковые алгоритмы. /Лаб/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.30	Параллельная реализация алгоритмов умножения матриц (алгоритмы традиционный, Винограда), её анализ, экспериментальные оценки времени выполнения. /Лаб/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.31	Моделирование конвейера в параллельном режиме: одна лента конвейера симулируется одним потоком /Лаб/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.32	Разработка вычислительно эффективных алгоритмов. Метод декомпозиции задачи. Математические основы параллельных вычислений. Основы разработки параллельных алгоритмов, приемы оценки их сложности. Влияние архитектуры вычислительной системы на разработку алгоритмов. Методы оценки сложности рекурсивных алгоритмов. /Ср/	6	11	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.33	Пошаговая оптимизация. Метод динамического программирования. Муравьиный алгоритм. Некоторые эффективные алгоритмы. /Тема/	6	0			
1.34	Динамическое программирование. Задача оптимальной упаковки (задача о рюкзаке). Основы метода динамического программирования. Пошаговая оптимизация. Классическая задача упаковки и основное уравнение Беллмана для точного решения задачи упаковки — табличная реализация функционального уравнения /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.35	Рекурсивный алгоритм метода динамического программирования для задачи упаковки. Рекурсивная реализация алгоритма Беллмана для задачи упаковки. Параметризация задачи упаковки по среднему объёму грузов. Структура рекурсивного дерева и подсчет количества вершин порождённого дерева рекурсии. Оценка сложности алгоритма. Сравнительный анализ табличного и рекурсивного алгоритмов. Алгоритмы теории чисел. Алгоритмическое обеспечение криптосистемы RSA. Теорема Ферма-Эйлера. Расширенный алгоритм Евклида /Лек/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет



1.36	Метод декомпозиции задачи. Некоторые эффективные алгоритмы. Метод декомпозиции задачи. Модификации алгоритмов поиска максимума, умножения комплексных чисел, умножения матриц /Пр/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.37	Реализация и параметризация муравьиного алгоритма. /Лаб/	6	2	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
1.38	Пошаговая оптимизация. Метод динамического программирования. Муравьиный алгоритм. Некоторые эффективные алгоритмы. /Ср/	6	5	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	Зачет
<b>Раздел 2. Промежуточная аттестация</b>						
2.1	Промежуточная аттестация /Тема/	6	0			
2.2	Прием зачета /ИКР/	6	0,25	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	
2.3	Подготовка к зачету /Зачёт/	6	8,75	ПК-3.2-3 ПК-3.2-У ПК-3.2-В	Л1.1 Л1.2Л2.1Л3.1 Э1	

### 5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

«Оценочные материалы приведены в приложении к рабочей программе дисциплины (см. документ «Оценочные материалы по дисциплине «Анализ алгоритмов»»)

### 6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

#### 6.1. Рекомендуемая литература

##### 6.1.1. Основная литература

№	Авторы, составители	Заглавие	Издательство, год	Количество/название ЭБС
Л1.1	Алябьева В. Г., Пастухова Г. В.	Теория алгоритмов : учебное пособие для специальности 050201.65 – «математика с дополнительной специальностью “информатика”», направление подготовки 050100 – «педагогическое образование»	Пермь: Пермский государственный гуманитарно-педагогический университет, 2013, 125 с.	978-5-85218-624-9, <a href="http://www.iprbookshop.ru/32100.html">http://www.iprbookshop.ru/32100.html</a>
Л1.2	Брыкалова А. А.	Теория алгоритмов : учебное пособие	Ставрополь: Северо-Кавказский федеральный университет, 2016, 129 с.	2227-8397, <a href="http://www.iprbookshop.ru/69440.html">http://www.iprbookshop.ru/69440.html</a>

##### 6.1.2. Дополнительная литература

№	Авторы, составители	Заглавие	Издательство, год	Количество/название ЭБС
Л2.1	Поляков В. И., Скорубский В. И.	Основы теории алгоритмов	Санкт-Петербург: НИУ ИТМО, 2012, 51 с.	, <a href="http://e.lanbook.com/books/element.php?pl1_id=43564">http://e.lanbook.com/books/element.php?pl1_id=43564</a>

##### 6.1.3. Методические разработки

№	Авторы, составители	Заглавие	Издательство, год	Количество/название ЭБС
ЛЗ.1	Опабекова А. М., Умбетбаев К. У., Беделов К. А.	Теория алгоритмов : учебно-методический комплекс	Алматы: Нур-Принт, 2012, 79 с.	9965-756-08-2, <a href="http://www.iprbookshop.ru/67154.html">http://www.iprbookshop.ru/67154.html</a>
<b>6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"</b>				
Э1	Электронная библиотека РГРТУ <a href="https://elib.rsreu.ru/ebs">https://elib.rsreu.ru/ebs</a>			
<b>6.3 Перечень программного обеспечения и информационных справочных систем</b>				
<b>6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства</b>				
Наименование		Описание		
Adobe Acrobat Reader		Свободное ПО		
Visual studio community		Свободное ПО		
Chrome		Свободное ПО		
OpenOffice		Свободное ПО		
LibreOffice		Свободное ПО		
Notepad++		Свободное ПО		
PyCharm Community		Свободное ПО		
Qt		Лицензия Opensource		
<b>6.3.2 Перечень информационных справочных систем</b>				
6.3.2.1	Справочная правовая система «КонсультантПлюс» (договор об информационной поддержке №1342/455-100 от 28.10.2011 г.)			
6.3.2.2	Система КонсультантПлюс <a href="http://www.consultant.ru">http://www.consultant.ru</a>			
6.3.2.3	Информационно-правовой портал ГАРАНТ.РУ <a href="http://www.garant.ru">http://www.garant.ru</a>			

<b>7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)</b>	
1	110 учебно-административный корпус. Аудитория для самостоятельной работы 20 мест Проектор: НITACHI CP-X400 3LCD 21 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel Core i5-4570 ОЗУ: 8 Гб ПЗУ: 1 Тб (1 шт.)
2	110 учебно-административный корпус. Учебная аудитория для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации 20 мест Проектор: НITACHI CP-X400 3LCD 21 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel Core i5-4570 ОЗУ: 8 Гб ПЗУ: 1 Тб (1 шт.)
3	106а учебно-административный корпус. Учебная аудитория для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации 42 мест проектор BENQ 15 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: 2x Intel Pentium II/III class 2126, ОЗУ: 2 Гб, ПЗУ: 74 Гб (1 шт) ЦП: Intel Pentium II/III class 3192, ОЗУ: 4 Гб, ПЗУ: 200 Гб (13 шт.) ЦП: Intel Pentium II/III class 2128, ОЗУ: 2 Гб ПЗУ: 74 Гб (1 шт.)

4	206-3 учебно-административный корпус. Учебная аудитория для проведения практических занятий, лабораторных работ Проектор: InFocus LP640 18 ПК с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду: ЦП: Intel Core 2 ОЗУ: 4 Гб ПЗУ: 70 Гб (19 шт.)
---	--

#### **8. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Лекционные занятия посвящены рассмотрению ключевых, базовых положений курса и разъяснению учебных заданий, выносимых на самостоятельную проработку.

Семинары проводятся для закрепления усвоенной информации, приобретения навыков ее применения для решения практических задач в предметной области дисциплины.

Лабораторные работы предназначены для приобретения опыта практической реализации основной профессиональной образовательной программы. Методические документы к лабораторным работам прорабатываются студентами во время самостоятельной подготовки. Необходимый уровень подготовки контролируется перед проведением лабораторных работ.

Практическая подготовка при реализации учебной дисциплины организуется путем проведения семинаров, практических занятий, практикумов, лабораторных работ и индивидуальных и(или) групповых консультаций, предусматривающих участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

Практическая подготовка может включать в себя отдельные занятия лекционного типа, которые предусматривают передачу учебной информации обучающимся, необходимой для последующего выполнения работ, связанных с будущей профессиональной деятельностью.

Самостоятельная работа студентов включает следующие виды: проработка учебного материала лекций, подготовка к семинарам, подготовка к лабораторным работам.

Для завершения работы в семестре студент должен выполнить все контрольные мероприятия.

Промежуточная аттестация по дисциплине проходит в форме зачета.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»**

Кафедра «Вычислительная и прикладная математика»

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ  
«Анализ алгоритмов»**

Направление подготовки  
09.03.04 «Программная инженерия»

Направленность (профиль) подготовки  
«Программное обеспечение систем искусственного интеллекта»

Уровень подготовки – бакалавриат

Квалификация выпускника – бакалавр

Форма обучения – очная

Срок обучения – 4 года

Рязань 2023 г.

## 1. ОБЩИЕ ПОЛОЖЕНИЯ

*Оценочные материалы* – это совокупность учебно-методических материалов и процедур, предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы.

*Цель* – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

*Основная задача* – обеспечить оценку уровня сформированности компетенций и индикаторов их достижения, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний обучающихся проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости и промежуточная аттестация проводятся с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся, организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков обучающихся на практических занятиях по результатам выполнения и защиты обучающимися индивидуальных заданий, по результатам выполнения контрольных работ и тестов, по результатам проверки качества конспектов лекций и иных материалов.

В качестве оценочных средств на протяжении семестра используется устные и письменные ответы студентов на индивидуальные вопросы, письменное тестирование по теоретическим разделам курса, реферат. Дополнительным средством оценки знаний и умений студентов является отчет о выполнении практических заданий и его защита.

По итогам курса обучающиеся сдают зачет.

### 1. Перечень компетенций с указанием этапов их формирования

При освоении дисциплины формируются следующие компетенции: ПК-3 (индикатор ПК-3.2).

Указанные компетенции формируются в соответствии со следующими этапами:

- формирование и развитие теоретических знаний, предусмотренных указанными компетенциями (лекционные занятия, самостоятельная работа студентов);
- приобретение и развитие практических умений предусмотренных компетенциями (практические занятия, самостоятельная работа студентов);
- закрепление теоретических знаний, умений и практических навыков, предусмотренных компетенциями, в ходе решения конкретных задач на занятиях, выполнения индивидуальных заданий на практических занятиях и их защиты, а так же в процессе сдачи зачета.

### 2. Показатели и критерии оценивания компетенций (результатов) на различных этапах их формирования, описание шкал оценивания

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

При достаточном качестве освоения более 80% приведенных знаний, умений и

навыков преподаватель оценивает освоение данной компетенции в рамках настоящей дисциплины на эталонном уровне, при освоении более 60% приведенных знаний, умений и навыков – на продвинутом, при освоении более 40% приведенных знаний, умений и навыков – на пороговом уровне. При освоении менее 40% приведенных знаний, умений и навыков компетенция в рамках настоящей дисциплины считается неосвоенной.

**Уровень сформированности** каждой компетенции на различных этапах ее формирования в процессе освоения данной дисциплины оценивается в ходе текущего контроля успеваемости и представлено различными видами оценочных средств.

**Оценке сформированности в рамках данной дисциплины подлежат компетенции/индикаторы:**

Показатели достижения планируемых результатов обучения и критерии их оценивания на разных уровнях формирования компетенций приведены в таблице 1.

**Таблица 1. Показатели достижения индикаторов компетенции**

1	2	3	4
<b>Компетенция: код по ФГОС 3++, формулировка</b>	<b>Индикаторы</b>	<b>Этап</b>	<b>Наименование оценочного средства</b>
ПК-3 (09.03.04/02 Программное обеспечение систем искусственного интеллекта) Способен разрабатывать и тестировать программные компоненты решения задач в системах ИИ	<b>ПК-3.2 Разрабатывает приложения систем искусственного интеллекта</b> <b>ЗНАТЬ</b> - знает современные языки программирования, библиотеки и программные платформы для функционального, логического, объектно-ориентированного программирования приложений систем искусственного интеллекта (Python, R, C++, C#) <b>УМЕТЬ</b> - разрабатывать программные приложения систем искусственного интеллекта, с использованием современных языков программирования, библиотек и программных платформ функционального, логического, объектно-ориентированного программирования (Python, R, C++, C#) <b>ВЛАДЕТЬ</b> - основными принципами разработки приложений систем искусственного интеллекта	1	Зачет

Преподавателем оценивается содержательная сторона и качество материалов, приведенных в отчетах студента по практическим занятиям. Кроме того, преподавателем

учитываются ответы студента на вопросы по соответствующим видам занятий при текущем контроле:

- контрольные опросы;
- задания для практических занятий.

Принимается во внимание **знания** обучающимися:

- основных программных платформ и компонентов систем искусственного интеллекта: механизмы логического вывода (рассуждений) объяснений, приобретения знаний, интеллектуальных интерфейсов, принципы Data Ops и Dev Ops;

- современных языков программирования, библиотек и программных платформ для функционального, логического, объектно-ориентированного программирования приложений систем искусственного интеллекта (Python, R, C++, C#);

- основных критериев качества систем искусственного интеллекта, методов и инструментальных средств тестирования работоспособности и качества функционирования систем искусственного интеллекта;

**наличие умений:**

- настраивать основные программные платформы и компоненты систем искусственного интеллекта: механизмов логического вывода (рассуждений), объяснений, приобретений знаний, интеллектуальных интерфейсов на особенности проблемной области, участвует в их разработке;

- разрабатывать программные приложения систем искусственного интеллекта, с использованием современных языков программирования, библиотек и программных платформ функционального, логического, объектно-ориентированного программирования (Python, R, C++, C#);

- проводить тестирование работоспособности и качества функционирования систем искусственного интеллекта и проверять выполнение требований к системам искусственного интеллекта со стороны пользователя;

**обладание навыками:**

- настройки программного обеспечения и компонентов систем искусственного интеллекта;

- основными принципами разработки приложений систем искусственного интеллекта;

- методологией тестирования систем искусственного интеллекта.

Критерии оценивания уровня сформированности компетенции в процессе выполнения практических работ:

41%-60% правильных ответов соответствует пороговому уровню сформированности компетенции на данном этапе ее формирования;

61%-80% правильных ответов соответствует продвинутому уровню сформированности компетенции на данном этапе ее формирования;

81%-100% правильных ответов соответствует эталонному уровню сформированности компетенции на данном этапе ее формирования.

Сформированность уровня компетенций не ниже порогового является основанием для допуска обучающегося к промежуточной аттестации по данной дисциплине.

Формой промежуточной аттестации по данной дисциплине является зачет, оцениваемый по принятой в ФГБОУ ВО «РГРТУ» системе: «зачтено» и «не зачтено».

Критерии оценивания промежуточной аттестации представлены в таблице.

<b>Шкала оценивания</b>	<b>Критерии оценивания</b>
<b>«зачтено»</b>	<b>оценки «зачтено»</b> заслуживает обучающийся, продемонстрировавший полное знание материала изученной дисциплины, усвоивший основную литературу, рекомендованную рабочей программой дисциплины; выполнивший все практические задания; показавший систематический характер знаний по дисциплине, ответивший на все вопросы билета или допустивший погрешность в ответе вопросы, но обладающий необходимыми знаниями для их устранения под руководством преподавателя;
<b>«не зачтено»</b>	<b>оценки «не зачтено»</b> заслуживает обучающийся, не выполнивший практические задания, продемонстрировавший серьезные пробелы в знаниях основного материала изученной дисциплины, не ответивший на все вопросы билета и дополнительные вопросы. Оценка «не зачтено» ставится обучающимся, которые не могут продолжить обучение по образовательной программе без дополнительных занятий по соответствующей дисциплине (формирования и развития компетенций, закрепленных за данной дисциплиной).

### **3. Типовые контрольные задания или иные материалы**

ФОС по дисциплине содержит следующие оценочные средства, позволяющие оценить знания, умения и уровень приобретенных компетенций при текущем контроле и промежуточной аттестации, разбитые по модулям дисциплины:

- перечни вопросов на зачет;
- макеты билетов к зачету.

Средства для оценки различных уровней формирования компетенций по категориям «знать», «уметь», «владеть» обеспечивают реализацию основных принципов контроля, таких, как объективность и независимость, практико-ориентированность, междисциплинарность.

С учетом этого, контрольные вопросы (задания, задачи,) входящие в ФОС, для различных категорий и уровней освоения компетенций имеют следующий вид:

#### **Уровень ЗНАТЬ**

<b>Дескрипторы</b>	<b>Пример задания из оценочного средства</b>
современные языки программирования, библиотеки и программные платформы для функционального, логического, объектно-ориентированного программирования приложений систем ИИ (Python, R, C++, C#)	<ol style="list-style-type: none"> <li>1. Описать различия между рекурсивной и не рекурсивной реализациями алгоритма поиска расстояния Левенштейна. Указать на особенности емкостной и временной сложности.</li> <li>2. Дать классификацию алгоритмов по трудоемкости. Указать, что требуется для задания модели вычислений.</li> <li>3. Дать классификацию видов параллелизма. Сформулировать закон Амдала. Перечислить факторы, ограничивающие выигрыш от распараллеливания.</li> <li>4. Дать определение графовых моделей алгоритма: информационного графа, графа управления, информационной и операционной истории. Указать, для которой модели невозможно ветвление.</li> </ol>



## Уровень УМЕТЬ

Дескрипторы	Пример задания из оценочного средства
разрабатывать программные приложения СИИ с использованием современных языков программирования, библиотек и программных платформ для функционального, логического, объектно-ориентированного программирования (Python, R, C++, C#)	<ol style="list-style-type: none"><li>1. Провести сравнение по трудоёмкости алгоритмов сортировки пузырьком и на основании бинарного дерева.</li><li>2. Описать стандартный алгоритм умножения матриц четырьмя графовыми моделями.</li><li>3. Разработать схему параллельного алгоритма нахождения расстояния Левенштейна. Указать ограничения, проистекающие из зависимости фрагментов алгоритма по данным.</li><li>4. Разработать алгоритм нахождения слова в словаре, предполагающий сегментированное хранение словаря и учитывающий частоту употребления первой буквы слова.</li><li>5. Написать регулярное выражение, позволяющее выделять из текста электронные адреса, и составить схему состояний конечного автомата, допускающего тот же поиск.</li><li>6. Оценить согласно закону Амдала максимально достижимое ускорение на 4 процессорах при 20 % кода, не допускающего параллельного выполнения. Приняв за константы затраты за диспетчеризацию, время выполнения линейного участка кода и время последовательного выполнения распараллеливаемого участка кода, дать оценку времени выполнения кода в параллельном режиме на 4 процессорах.</li></ol>

### Перечни вопросов к зачету

1. Написать регулярное выражение, позволяющее выделять из текста подстроку согласно варианту. Составить схему состояний конечного автомата, допускающего тот же поиск. Разработать программу, составляющую конечный автомат для поиска введённой подстроки в строке. Разработать программу, осуществляющую поиск подстроки в строке на основании сформированного конечного автомата.

*Примеры вариантов:*

*Вариант №1.* Поиск электронных адресов.

*Вариант №2.* Поиск телефонных номеров.

2. Реализовать рекурсивно и не рекурсивно алгоритм, согласно варианту. Провести оценку реализаций алгоритма по трудоёмкости и затратам по памяти, учитывая затраты на рекурсию. Разработать программное обеспечение, реализующее обе версии алгоритма и позволяющее провести замеры времени выполнения каждой реализации и их затраты по памяти. Провести эксперименты, на основании замеров и сравнительного анализа двух реализаций сделать вывод об их эффективности.

При решении задачи использовать элементы программного кода для замеров времени работы реализаций алгоритмов, созданные в лабораторной работе 2.

*Примеры вариантов:*

*Вариант №1.* Умножение матрицы на основании метода декомпозиции задачи, с делением матрицы на части, шаг останова рекурсии – тривиальные размеры текущих участков матриц.

*Вариант №2.* Нахождение факториала от целого числа N. При переполнении разрядной сетки выдавать ответ 0.

## **Перечень лабораторных работ**

### **ЛР1.1 Метод динамического программирования. Линейная и рекурсивная реализации алгоритмов определения расстояния Левенштейна и расстояния Дамерау-Левенштейна между строками и их анализ.**

Цель работы: изучение и применение метода динамического программирования на материале алгоритмов Левенштейна и Дамерау-Левенштейна; получение практических навыков реализации указанных алгоритмов.

Задачи работы: изучение алгоритмов Левенштейна и Дамерау-Левенштейна нахождения расстояния между строками; реализация двух алгоритмов в матричной версии и одного из алгоритмов в рекурсивной версии; сравнение рекурсивной и не рекурсивных реализаций по затрачиваемым ресурсам (времени и памяти); экспериментальное подтверждение различий в эффективности рекурсивной и не рекурсивных реализаций при помощи разработанного программного обеспечения на материале замеров времени выполнения каждого алгоритма на варьирующихся длинах строк; описание и обоснование полученных результатов в отчете о выполненной лабораторной работе.

### **ЛР 1.2 Реализация и анализ традиционного алгоритма умножения матриц, эффективного алгоритма Винограда, оптимизированного алгоритма Винограда. Теоретические и экспериментальные оценки трудоёмкости и ёмкости по памяти.**

Цель работы: получение практических навыков оценки трудоёмкости алгоритмов умножения матриц, получение практических навыков реализации алгоритмов умножения матриц с замером времени выполнения.

Задачи работы: изучение трудоёмкостей алгоритмов умножения матриц и получение их теоретических оценок на основании анализа количества выполняемых операций в выбранной модели оценки трудоёмкости; получение навыка оптимизации алгоритма с целью снижения трудоёмкости его выполнения на примере решения задачи умножения матриц; программная реализация стандартного алгоритма умножения, алгоритма Винограда и оптимизированного по трудоёмкости студентом алгоритма Винограда (должны быть применены минимум 3 типа улучшений; эффект от оптимизации должен быть описан в терминах трудоёмкости) с функциональностью замера времени выполнения реализации каждого алгоритма; экспериментальное подтверждение при помощи разработанного программного обеспечения оценок трудоёмкости на материале замеров времени выполнения реализаций трех алгоритмов на варьирующихся размерах матриц (в т.ч. для лучшего и худшего случаев для алгоритма Винограда); описание и обоснование полученных результатов в отчете о выполненной лабораторной работе.

### **ЛР 1.3 Реализация и анализ нерекурсивных алгоритмов сортировки. Теоретические и экспериментальные оценки трудоёмкости.**

Цель работы: получение практических навыков оценки трудоёмкости нерекурсивных алгоритмов на материале алгоритмов сортировки; получение практических навыков реализации не рекурсивных алгоритмов сортировки с подсчётом количества выполненных операций и с замером времени выполнения.

Задачи работы: изучение трудоёмкостей не рекурсивных алгоритмов сортировки массива и получение теоретических оценок трех алгоритмов на основании анализа количества выполняемых операций в выбранной модели оценки трудоёмкости; разработка

программного обеспечения, реализующего не рекурсивные алгоритмы сортировки с замером времени выполнения каждого алгоритма; экспериментальное подтверждение при помощи разработанного программного обеспечения оценок трудоёмкости на материале замеров времени выполнения трех алгоритмов на варьирующихся размерах массивов для лучшего, худшего и произвольного случаев (которые могут различаться для различных алгоритмов); описание и обоснование полученных результатов в отчете о выполненной лабораторной работе.

#### **ЛР 1.4 Реализация алгоритмов поиска подстроки в строке.**

Цель работы: получение практических навыков реализации алгоритмов поиска подстроки в строке.

Задачи работы: изучение алгоритмов поиска подстроки в строке (стандартного, Кнута-Морриса-Пратта, Бойера-Мура) и их трудоёмкостей в терминах затраченных сравнений в лучшем и худшем случаях; разработка программного обеспечения, реализующего указанные алгоритмы; описание алгоритмов поиска подстроки в строке, включая процедуры заполнения вспомогательных массивов, с примерами и тестами, а также описание реализованного программного обеспечения и полученных результатов в отчете о выполненной лабораторной работе.

#### **ЛР 2.1 Графовые модели алгоритмов. Поточковые алгоритмы.**

Цель работы: получение практических навыков описания алгоритма графовыми моделями и анализа алгоритма на материале графовых моделей на предмет выделения зависимости по данным и выделения ресурса параллелизма, получение практических навыков реализации потоковых алгоритмов.

Задачи работы: описание алгоритма четырьмя графовыми моделями: графом управления, информационным графом, операционной историей и информационной историей; анализ алгоритма на предмет выделения зависимостей по данным, информационно независимых участков, ресурса параллелизма (конечного, массового); разработка и реализация потокового алгоритма; описание и обоснование полученных результатов в отчете о выполненной лабораторной работе.

#### **ЛР 2.2 Параллельная реализация алгоритмов умножения матриц (алгоритмы традиционный, Винограда), её анализ, экспериментальные оценки времени выполнения.**

Цель работы: получение практических навыков оценки ресурса параллелизма алгоритмов умножения матриц; получение практических навыков оценки трудоёмкости параллельных алгоритмов умножения матриц; получение практических навыков реализации параллельных алгоритмов умножения матриц с замером времени выполнения для параллельных и последовательных реализаций алгоритмов.

Задачи работы: изучение подходов к выделению ресурса параллелизма и к распараллеливанию алгоритмов на материале алгоритмов умножения матриц (стандартного и Винограда); изучение трудоёмкости для фрагментов алгоритмов умножения матриц (подвергающихся и не подвергающихся распараллеливанию) и получение её оценок для алгоритма в целом; получение навыка анализа информационной зависимости фрагментов алгоритма и корректной адаптации разработанного параллельного алгоритма к этим зависимостям; получение навыка параллельной

реализации алгоритма с целью снижения трудоёмкости выполнения в сравнении с последовательным алгоритмом на примере решения задачи умножения матриц; экспериментальное подтверждение при помощи разработанного программного обеспечения оценок трудоёмкости на материале замеров времени выполнения алгоритмов на варьирующихся размерах матриц (для лучшего и худшего случаев) при выполнении параллельной реализации с различным количеством потоков и при выполнении последовательной версии алгоритмов; описание и обоснование полученных результатов в отчете о выполненной лабораторной работе.

### **ЛР 2.3 Моделирование конвейера в параллельном режиме: одна лента конвейера симулируется одним потоком**

Цель работы: получение практических навыков оценки ресурса параллелизма алгоритмов при переносе на конвейерную схему вычислений; получение практических навыков оценки трудоёмкости потоковых параллельных алгоритмов; получение практических навыков реализации параллельных алгоритмов.

Задачи работы: изучение подходов к выделению ресурса параллелизма и к распараллеливанию алгоритмов на материале выбранной задачи (по вариантам; например, блочное шифрование); получение навыка анализа информационной зависимости фрагментов алгоритма и корректной адаптации разработанного параллельного алгоритма к этим зависимостям; разработка алгоритма, моделирующего работу конвейера, с учетом особенностей синхронизации между потоками; получение навыка параллельной реализации поточного алгоритма как конвейера с целью снижения трудоёмкости выполнения в сравнении с последовательным алгоритмом; экспериментальное подтверждение при помощи разработанного программного обеспечения эффективности параллельной реализации по сравнению с однопоточной; описание и обоснование полученных результатов в отчете о выполненной лабораторной работе.

### **ЛР 3.1 Реализация и параметризация муравьиного алгоритма.**

Цель работы: получение практических навыков реализации эвристических алгоритмов, позволяющих решить задачу оптимизации с приемлемым качеством и за меньшее время; получение практических навыков параметризации алгоритма на примере муравьиного алгоритма и изучения влияния настроечных параметров алгоритма на результирующие значения (в задаче коммивояжера – на длину пути).

Задачи работы: изучение эволюционных алгоритмов, в частности, муравьиного алгоритма; применение муравьиного алгоритма к решению задачи коммивояжера; разработка программного обеспечения, реализующего метод перебора и муравьиный алгоритм применительно к решению задачи коммивояжера; экспериментальное подтверждение различия в трудоёмкостях реализованных алгоритмов при помощи разработанного программного обеспечения на материале замеров времени выполнения алгоритмов; параметризация муравьиного алгоритма и изучение влияния настроечных параметров алгоритма на результирующие значения (в задаче коммивояжера – на длину пути) при помощи разработанного программного обеспечения; описание и обоснование полученных результатов в отчете о выполненной лабораторной работе.