

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных**

Документ подписан усиленной неквалифицированной электронной подписью
Организация: Финансовый университет при Правительстве РФ
Утверждено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: h/IeAi56hs3aHjVwA+4u5M+AhO6bKFP+
Дата: 29.12.2025 г.

А.А. Антонов

Теория алгоритмов

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

09.03.03 - Прикладная информатика,

Образовательная программа «Прикладные информационные системы в экономике
и финансах»

Профиль:

«Инженерия данных»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 03 от 16.12.2025 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 12 от 03.12.2025 г.)*

© Москва 2025

СОДЕРЖАНИЕ

1.	Наименование дисциплины	3
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3.	Место дисциплины в структуре образовательной программы	4
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	4
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	5
5.1.	Содержание дисциплины	5
5.2.	Учебно-тематический план	7
5.3.	Содержание семинаров	8
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	10
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	10
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	12
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	16
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	25
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	25
10.	Методические указания для обучающихся по освоению дисциплины	27
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	27
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	28

1. Наименование дисциплины

«Теория алгоритмов».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКи-2	Способность разрабатывать алгоритмы и программы с использованием современных технологий программирования	Владеет объектно-ориентированным языком программирования на уровне знания синтаксиса и семантики, основ стандартной библиотеки	знать: синтаксис и семантику базовых конструкций объектно-ориентированного языка программирования уметь: строить обоснованные выводы, понимать концепции доказуемости
		Использует инструментальные средства программирования (IDE, SDK, API, популярные фреймворки и библиотеки)	знать: основные понятия теории алгоритмов и основные алгоритмы на графах уметь: применять методы анализа сложности алгоритмов для решения прикладных задач профессиональной деятельности
		Организовывает кодовую базу, ориентируется в существующем коде, демонстрирует знание общепринятых соглашений и политик в области оформления кода	знать: принципы организации кодовой базы, общепринятые соглашения по оформлению кода и основы работы с системами контроля версий уметь: применять соглашения по оформлению кода, организовывать файлы проекта в логическую структуру, использовать систему контроля версий для управления изменениями и читать, анализировать и модифицировать существующий код в соответствии с принятыми стандартами
		Проектирует текстовый, программный или графический интерфейс программной системы исходя из ее назначения	знать: основные приемы оценки сложности программных алгоритмов уметь: оценивать сложность алгоритма по исходному коду, псевдокоду или эквивалентным описаниям

3. Место дисциплины в структуре образовательной программы

Дисциплина «Теория алгоритмов» относится к «Общефакультетскому (предпрофильному) циклу».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 4 (в часах)
Общая трудоёмкость дисциплины	3/108	108
Контактная работа- Аудиторные занятия	50	50
Лекции	16	16
Семинары, практические занятия	34	34
Самостоятельная работа	58	58
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Элементы теории информации

Дискретные сообщения, алфавит сообщения, объем алфавита. Количественная мера информации. Энтропия, формула Шеннона. Информационная и физическая энтропия. Базовые правила комбинаторики. Количество информации по Колмогорову. Единицы количества информации.

Тема 2. Передача и кодирование информации

Основные определения каналов связи и источников информации. Характеристики источников: равномерность, избыточность и производительность. Теорема Шеннона для дискретного канала связи. Соотношение сигнал-шум. Непрерывные каналы связи. Дискретизация.

Понятие и типы кодирования. Статистическое кодирование, кодирование по Хаффману. Принципы и методы сжатия данных. Кодирование и сжатие информации в современных компьютерных сетях. Сжатие информации с потерями и без потерь.

Тема 3. Основы теории алгоритмов

Определение алгоритма. Универсальные модели алгоритма. Рекурсивные функции. Тезис Чёрча. Машина Тьюринга. Тезис Тьюринга. Проблема остановки.

Тема 4. Теория сложности алгоритмов

Временная сложность алгоритмов. Асимптотический анализ трудоемкости алгоритмов. Классы сложности алгоритмов. Классы P, NP, EXP. Проблема P=NP. Понятие NP-полных задач. Важные примеры NP-полных задач. Пространственная сложность. Классы PSPACE и NPSPACE. PSPACE-полнота.

Тема 5. Структуры данных

Понятие о структурах данных. Структурное программирование. Простые и составные структуры данных. Динамические структуры. Линейные списки. Деревья. Накопители данных: стеки и очереди. Операции со стеком, двусторонней очередью. Строки. Задача поиска подстроки в строке.

Тема 6. Сортировка и поиск

Внешние и внутренние сортировки. Простые методы сортировки массивов: простое включение, простой выбор, метод пузырька. Улучшенные методы сортировки массивов: сортировка Шелла, пирамидальная сортировка, быстрая сортировка Хоара. Внешние сортировки: сортировка слиянием, естественное слияние Вирта, многофазная сортировка и ее анализ. Цифровая сортировка. Поиск элемента: в упорядоченном массиве, хеширование, деревья.

Тема 7. Алгоритмы на графах и деревьях

Понятие графа, основные задачи теории графов. Представление графов в ЭВМ. Графы и бинарные отношения. Деревья. Обходы графов. Поиск в глубину и поиск в ширину. Эйлеров и гамильтонов пути. Поиск компонент связности и бикомпонентов. Оптимизационные задачи на графах. Минимальный остов (алгоритмы Краскала, Прима), минимальное паросочетание (венгерский алгоритм). Поиск кратчайшего пути (алгоритм Дейкстры). Специальные виды деревьев: красно-черное дерево, двоичное дерево поиска, В-дерево. Алгоритмы и применение таких структур данных на практике.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия	
1	Элементы теории информации	8	4	2	2	4
2	Передача и кодирование информации	12	4	2	2	8
3	Основы теории алгоритмов	8	4	2	2	4
4	Теория сложности алгоритмов	16	6	2	4	10
5	Структуры данных	18	8	2	6	10
6	Сортировка и поиск	18	8	2	6	10
7	Алгоритмы на графах и деревьях	28	16	4	12	12
	Итого	108	50	16	34	58

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Элементы теории информации	Решение задач по комбинаторике и расчету количества информации. Реализация на языке высокого уровня алгоритмов вычисления комбинаторных формул и мер информации (Хартли, Шеннон). Отработка синтаксиса языка высокого уровня и анализа сложности алгоритмов для построения обоснованных выводов.	Решение задач в интерактивной форме, проверка самостоятельной работы и разбор ошибок, аудиторная проверочная работа.
Передача и кодирование информации	Решение задач по определению характеристик источников информации и соотношения сигнал/шум. Практика кодирования информации с реализацией алгоритмов обнаружения и коррекции ошибок на языке высокого уровня. Анализ эффективности кодов и помехоустойчивости каналов связи с использованием инструментов разработки.	Решение задач в интерактивной форме, проверка самостоятельной работы и разбор ошибок, аудиторная проверочная работа.
Основы теории алгоритмов	Решение задач по проектированию машин Тьюринга для различных вычислительных проблем. Реализация моделей Тьюринга на языке высокого уровня с использованием объектно-ориентированного подхода.	Решение задач в интерактивной форме, проверка самостоятельной работы и разбор ошибок, аудиторная проверочная работа.
Теория сложности алгоритмов	Решение задач по определению временной и пространственной сложности алгоритмов. Практика асимптотического анализа с использованием Big O нотации на примерах кода языка высокого уровня. Сравнение теоретических расчетов с практическими замерами производительности алгоритмов.	Решение задач в интерактивной форме, проверка самостоятельной работы и разбор ошибок, аудиторная проверочная работа.
Структуры данных	Решение задач по реализации базовых структур данных (списки, стеки, очереди, деревья) на языке высокого уровня. Сравнение эффективности различных структур данных и анализ сложности операций. Практика объектно-ориентированного проектирования и тестирования реализованных структур.	Решение задач в интерактивной форме, проверка самостоятельной работы и разбор ошибок, аудиторная проверочная работа.
Сортировка и поиск	Реализация и сравнение алгоритмов сортировки (быстрая, слиянием, пирамидальная) и поиска на языке высокого уровня. Анализ временной и пространственной сложности алгоритмов на практических примерах. Оптимизация кода с	Решение задач в интерактивной форме, проверка самостоятельной работы и разбор

	использованием инструментов профилирования и тестирования производительности.	ошибок, аудиторная проверочная работа.
Алгоритмы на графах и деревьях	Реализация алгоритмов обхода графов и работы с деревьями на языке высокого уровня. Решение прикладных задач на поиск кратчайших путей и минимальных остовных деревьев. Анализ эффективности графовых алгоритмов и оптимизация структур данных для хранения графов.	Решение задач в интерактивной форме, проверка самостоятельной работы и разбор ошибок, аудиторная контрольная работа.

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Элементы теории информации	Условная энтропия и взаимная информация. Энтропия как мера неопределенности. Определение и свойства дифференциальной энтропии.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия. Выполнение домашних заданий к каждому занятию.
Передача и кодирование информации	Словарные методы кодирования. Методы LZ77 и LZ78. Неравенство Крафта. Сжатие и кодирование графической и звуковой информации.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия. Выполнение домашних заданий к каждому занятию.
Основы теории алгоритмов	Другие модели абстрактных машин: многоленточные и недетерминированные машины Тьюринга, машина Поста, машины с произвольным доступом к памяти. Лямбда-исчисление Чёрча.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия. Выполнение домашних заданий к каждому занятию.
Теория сложности алгоритмов	Альтернативные подходы к определению класса NP. Теорема Кука-Левина. Дополнения языков из NP.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия. Выполнение домашних заданий к каждому занятию.
Структуры данных	Персистентные структуры данных. Хэширование. Хэширование с открытой адресацией. Матрицы и списки смежности. Квадродеревья и октодеревья.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия. Выполнение домашних заданий к каждому занятию.
Сортировка и поиск	Алгоритмы сортировки и поиска на больших	Работа с учебной

	объемах данных. Внешняя сортировка. Параллельная сортировка, сортировка с приоритетами.	литературой. Решение типовых задач. Разбор вопросов по теме занятия. Выполнение домашних заданий к каждому занятию.
Алгоритмы на графах и деревьях	Алгоритмы поиска минимального остовного дерева. Алгоритмы топологической сортировки. Алгоритмы поиска мостов и точек сочленения.	Работа с учебной литературой. Решение типовых задач. Разбор вопросов по теме занятия. Выполнение домашних заданий к каждому занятию.

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерные вопросы контрольной работы

1. Какая информация содержится в сообщении о том, что монетка упала гербом?
2. В ящике 10 гранат, из которых 8 без взрывателя. Из ящика наудачу выбираются 3 гранаты. Какое количество информации содержится в сообщении о том, что все выбранные гранаты оказались без взрывателя?
3. В группе 20 курсантов, среди которых 4 отличника, 6 хорошистов, 7 троечников, остальные двоечники. По списку наудачу отбираются 5 курсантов. Какое количество информации содержится в сообщении о том, что среди отобранных курсантов 3 отличника, 1 хорошист и 1 троечник?

Примерные задания контрольной работы

1. Дана строка из букв «а» и «b». Разработать машину Тьюринга, которая переместит все буквы «а» в левую, а буквы «b» – в правую части строки. Каретка находится над крайним левым символом строки.
2. На ленте машины Тьюринга находится целое положительное число, записанное в десятичной системе счисления. Найти произведение этого числа на число 11. Каретка обозревает крайнюю правую цифру числа.
3. На ленте машины Тьюринга находится десятичное число. Определить, делится ли это число на 5 без остатка. Если делится, то записать справа от числа слово «у», если нет – «п».
4. На ленте машины Тьюринга записано число в пятеричной системе счисления. Каретка находится над крайней правой цифрой. Записать цифры этого числа в обратном порядке.
5. Даны два натуральных числа n и m , представленные в унарной системе счисления. Между этими числами стоит знак «*». Построить машину Тьюринга, определяющую, равны эти числа или нет.
6. Даны два натуральных числа n и m , представленные в двоичной системе счисления. Между этими числами стоит знак «*». Найти разность этих чисел.

7. Дано число в двоичной системе счисления. Разработать машину Тьюринга, которая будет умножать это число на два (в двоичной системе счисления со сдвигом влево).

8. Сконструировать машину Тьюринга, которая выступит в качестве двоично-восьмеричного дешифратора.

9. Даны два натуральных числа n и m , заданных в унарной системе счисления. Числа n и m представлены наборами символов «1», разделенных «/». В конце набора стоит знак « = ». Разработать машину Тьюринга, которая будет производить деление нацело двух натуральных чисел n и m и находить остаток от деления.

10. Определите временную и пространственную сложность алгоритма:

```
for (int i = 0; i < n; ++i) { // В
for (int j = 1; j < m; ++j) { // A
    a[i][j]++;                // C
}
```

11. Реализовать алгоритм поиска наименьшего элемента в неотсортированном массиве и определить его временную и пространственную сложность.

12. Реализовать алгоритм сортировки массива методом пузырька и определить его временную и пространственную сложность.

13. Реализовать алгоритм поиска среднего значения в отсортированном массиве и определить его временную и пространственную сложность.

14. Реализовать алгоритм нахождения суммы всех элементов в двумерном массиве и определить его временную и пространственную сложность.

15. Реализовать алгоритм поиска наибольшего простого числа в заданном диапазоне и определить его временную и пространственную сложность.

16. Реализовать алгоритм поиска всех повторяющихся элементов в массиве и определить его временную и пространственную сложность.

17. Реализовать алгоритм проверки, является ли заданное число палиндромом, и определить его временную и пространственную сложность.

18. Реализовать алгоритм поиска наибольшей возрастающей подпоследовательности в массиве и определить его временную и пространственную сложность.
19. Реализовать алгоритм нахождения суммы всех элементов в связном списке и определить его временную и пространственную сложность.
20. Реализовать алгоритм поиска цикла в связном списке и определить его временную и пространственную сложность.
21. Реализовать алгоритм проверки, является ли заданная строка анаграммой другой строки, и определить его временную и пространственную сложность.
22. Реализовать алгоритм нахождения всех перестановок заданной строки и определить его временную и пространственную сложность.
23. Реализовать алгоритм поиска наибольшего количества одинаковых элементов в матрице и определить его временную и пространственную сложность.
24. Реализовать алгоритм нахождения наименьшего пути в графе от одной вершины к другой и определить его временную и пространственную сложность.
25. Реализовать алгоритм проверки, является ли заданное дерево двоичным поисковым деревом, и определить его временную и пространственную сложность.
26. Реализовать алгоритм нахождения всех простых чисел в заданном диапазоне и определить его временную и пространственную сложность.
27. Реализовать алгоритм нахождения всех подстрок заданной строки и определить его временную и пространственную сложность.
28. Реализовать алгоритм проверки, является ли заданное число совершенным (сумма его делителей равна самому числу), и определить его временную и пространственную сложность.
29. Реализовать алгоритм нахождения наибольшего общего делителя двух чисел и определить его временную и пространственную сложность.
30. Реализовать алгоритм нахождения наименьшего общего кратного двух чисел и определить его временную и пространственную сложность.

31. Построить двухленточную машину Тьюринга, допускающую язык всех слов из 0 и 1, в которых этих символов поровну. Первая лента содержит вход и просматривается слева направо. Вторая лента используется для запоминания излишка нулей по отношению к единицам и наоборот в прочитанной части входа. Описать состояния и переходы, а также неформальное предназначение каждого состояния.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. *Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.*

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

ПКН-2 Способность разрабатывать алгоритмы и программы с использованием современных технологий программирования

1) Владеет объектно-ориентированным языком программирования на уровне знания синтаксиса и семантики, основ стандартной библиотеки

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: синтаксис и семантику базовых конструкций объектно-ориентированного языка программирования

Уметь: строить обоснованные выводы, понимать концепции доказуемости

Типовые контрольные задания

Задание: Реализация операций с графом

Реализовать на языке высокого уровня программу для работы с графом:

1. Создание графа по заданной структуре
2. Визуализация графа
3. Подсчет количества вершин и ребер
4. Вычисление степени вершины и смежных вершин
5. Удаление вершины с перестроением графа
6. Визуализация модифицированного графа

Требования:

- Использовать ООП: классы Graph, Vertex, Edge

- Применить коллекции стандартной библиотеки (ArrayList, HashMap)
- Реализовать методы с соблюдением семантики ООП (инкапсуляция, наследование)
- Обеспечить корректность работы с памятью и исключениями
- Доказать корректность операций через тестирование

Критерии оценки:

- Правильное использование синтаксиса Java
- Грамотное применение ООП принципов
- Эффективное использование стандартной библиотеки
- Корректность работы с памятью и исключениями
- Качество тестов, доказывающих правильность реализации

2) Использует инструментальные средства программирования (IDE, SDK, API, популярные фреймворки и библиотеки)

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные понятия теории алгоритмов и основные алгоритмы на графах

Уметь: применять методы анализа сложности алгоритмов для решения прикладных задач профессиональной деятельности

Типовые контрольные задания

Задание: Нахождение максимального потока в графе

Реализовать на языке высокого уровня программу для:

1. Поиска максимального потока из вершины 1 в вершину 10 с помощью алгоритма Форда-Фалкерсона
2. Нахождения минимального разреза с помощью алгоритма расстановки пометок

Требования:

- Создать генератор случайных графов по заданному примеру
- Реализовать алгоритмы нативно без использования графовых библиотек
- Протестировать на графах разного размера и сложности

- Проанализировать временную сложность алгоритмов
- Визуализировать результаты работы алгоритмов

Критерии оценки:

- Корректность реализации алгоритмов Форда-Фалкерсона
- Умение анализировать сложность алгоритмов на практике
- Эффективное использование инструментов разработки (IDE, отладка)
- Качество тестирования и верификации результатов
- Соответствие теоретической и практической сложности

3) Организует кодовую базу, ориентируется в существующем коде, демонстрирует знание общепринятых соглашений и политик в области оформления кода

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: принципы организации кодовой базы, общепринятые соглашения по оформлению кода и основы работы с системами контроля версий

Уметь: применять соглашения по оформлению кода, организовывать файлы проекта в логическую структуру, использовать систему контроля версий для управления изменениями и читать, анализировать и модифицировать существующий код в соответствии с принятыми стандартами

Типовые контрольные задания

Задание: Анализ характеристик графа

Реализовать на языке высокого уровня программу для анализа графа со следующими функциями:

1. Вычисление степеней вершин
2. Построение матрицы смежности
3. Построение матрицы инцидентности
4. Вычисление эксцентриситета вершин
5. Определение радиуса, диаметра и центра графа
6. Поиск циклов в графе

Требования:

- Оформить каждый анализ в отдельном методе
- Реализовать структуру данных для представления графа
- Протестировать на графах разного размера (10, 50, 100 вершин)
- Сравнить время выполнения разных операций
- Визуализировать зависимости времени работы от размера графа
- Использовать Git для контроля версий с осмысленными коммитами

Критерии оценки:

- Корректность реализации алгоритмов анализа графа
- Качество оформления кода и модульность архитектуры
- Эффективность работы с графовыми структурами данных
- Точность вычислений и анализа результатов
- Соответствие теоретической и практической сложности алгоритмов
- Логичная история коммитов в Git и правильная организация репозитория

4) Проектирует текстовый, программный или графический интерфейс программной системы исходя из ее назначения

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные приемы оценки сложности программных алгоритмов

Уметь: оценивать сложность алгоритма по исходному коду, псевдокоду или эквивалентным описаниям

Типовые контрольные задания

Задание: Анализ алгоритмов сортировки

Реализовать на языке высокого уровня три алгоритма сортировки:

1. Быстрая сортировка (QuickSort)
2. Сортировка выбором (SelectionSort)
3. Пузырьковая сортировка (BubbleSort)

Требования:

- Оформить каждый алгоритм в отдельном методе
- Проверить корректность работы на массивах размером 10, 100, 1000 элементов
- Измерить время выполнения каждого алгоритма
- Построить график зависимости времени от размера массива
- Сравнить экспериментальные результаты с теоретической сложностью алгоритмов

Критерии оценки:

- Корректность реализации алгоритмов
- Качество оформления кода и модульность
- Точность измерений и анализа
- Соответствие теоретических и практических результатов

Примеры практико-ориентированных заданий

1. В отделе работают 6 мужчин и 4 женщины. По табельным номерам наудачу отобраны 7 человек. Какое количество информации содержится в сообщении о том, что среди отобранных лиц окажутся 3 женщины?
2. В лифт 7-этажного дома сели 3 пассажира. Каждый независимо от других с одинаковой вероятностью может выйти на любом (начиная со второго) этаже. Какое количество информации содержится в сообщении о том, что все вышли на разных этажах?
3. Имеются изделия четырех сортов, причем число изделий каждого сорта равно $n_1 = 2$, $n_2 = 3$, $n_3 = 1$, $n_4 = 3$. Для контроля наудачу берутся 5 изделий. Какое количество информации содержится в сообщении о том, что среди них $m_1 = 2$ - первосортных; $m_2 = 1$ - второго, $m_3 = 0$ - третьего и $m_4 = 2$ - четвертого сорта?
4. Таможенный контроль проходят 3 таджика, 4 грузина и 6 азербайджанцев. На досмотр наудачу вызывают трех человек. Какое количество информации содержится в сообщении о том, что все три вызванных являются грузинами?
5. Лектор произносит в среднем около сорока шестибуквенных слов в минуту. Рассматривая его как источник дискретных сообщений, определить его

производительность. Для простоты принять, что все буквы алфавита равновероятны и статистически независимы.

6. Двоичный источник с равновероятными элементами имеет производительность 1000 бит-с. При передаче по каналу в среднем один из переданных 100 символов искажается. Определить скорость передачи информации по данному каналу.

7. Имеется источник информации с энтропией в единицу времени $H = 100$ бит и два канала связи: каждый из них может передавать в единицу времени 70 бит информации, при этом в результате помехи, действующей на каждый из этих каналов, значение бита может быть заменено на противоположное с вероятностью 0.1. Вопрос - достаточна ли пропускная способность этих каналов для передачи информации, поставляемой источником?

8. Дан алфавит $A = \{1, 0\}$ и состояния $Q = \{q_1\}$. Построить машину Тьюринга, удаляющую 0.

9. Построить машину Тьюринга T^+ для сложения двух натуральных чисел, записанных в унарной системе. Построить диаграмму машины Тьюринга T^+ , $A = \{1\}$.

10. Дан алфавит $A = \{1\}$. Построить машину Тьюринга для получения следующего натурального числа.

11. Дано слово в алфавите $A = \{a, b, c, d\}$. Построить машину Тьюринга, удаляющую все буквы a .

12. Построить машину Тьюринга T^{++} для суммирования N натуральных подряд идущих чисел, записанных в унарной системе через «*». Пример: $111*11*1111* \dots *11$.

13. Построить машину Тьюринга T_- , которая вычисляет разность двух натуральных чисел, записанных в унарной системе через разделитель. Например: $11111-111$.

14. Найти пространственную и временную сложность алгоритма машины Тьюринга, вычисляющей в унарном коде функцию $f(x) = x^2$.

15. Найти пространственную и временную сложность алгоритма машины Тьюринга, вычисляющей в унарном коде функцию $f(x) = x^u$.

16. Найти пространственную и временную сложность алгоритма машины Тьюринга, вычисляющей в унарном коде функцию $f(x) = 2^x$.

17. Задан полный путь к файлу с таблицей вещественных (типа double) чисел, записанной с помощью стандартной функции `fwrite()`. Вначале записана ее высота h и ширина w (тип

int), а затем - элементы. Определить функцию Invert(), которая получает файл с таблицей, транспонирует таблицу и записывает ее в другой файл.

18. Текст - последовательность символов, состоящая из букв и пробелов. Слово - подпоследовательность, не содержащая пробелов. Тексты и слова хранятся в символьных строках. Создать функцию IndexWord(), обеспечивающую проверку присутствия в тексте заданного слова и вычисляющую позицию (от начала текста) первой буквы первого вхождения этого слова. При отсутствии слова в тексте функция возвращает -1.

19. Написать собственный вариант MyStrCmp() стандартной функции strcmp().

20. Текст - последовательность из букв и пробелов, хранящаяся в символьной строке. Создать функцию InsertWord(), осуществляющую вставку подпоследовательности символов (букв и пробелов) в текст, начиная с указанной позиции. Функция должна возвращать адрес символьной строки с измененным текстом. Пример. Текст: "abcd efgh ijklmo", слово: "word", позиция: 2 (нумерация начинается с нуля). Результат: "abwordcd efgh ijklmo".

21. Текст - последовательность из букв и пробелов, хранящаяся в символьной строке. Создать функцию ExluWord(), осуществляющую исключение из текста подпоследовательности символов (букв и пробелов), начинающейся с позиции start и заканчивающейся в позиции finish (start<=finish). Функция должна возвращать адрес символьной строки с измененным текстом.

Примерные вопросы для подготовки к зачету

1. Основные принципы и понятия теории информации.
2. Методы кодирования и декодирования информации.
3. Характеристики и свойства каналов связи.
4. Методы сжатия информации и их применение.
5. Теория ошибок и коррекция ошибок в передаче информации.
6. Условная энтропия и взаимная информация.
7. Дифференциальная энтропия.
8. Основные понятия и принципы кодирования информации.

9. Базовые методы кодирования: двоичный код, код Грея, коды Хаффмана.
10. Принципы сжатия информации и методы сжатия данных.
11. Кодирование аудио и видео: форматы и алгоритмы сжатия.
12. Кодирование текстовой информации: ASCII, Unicode, UTF-8.
13. Кодирование изображений: форматы и алгоритмы сжатия.
14. Кодирование сигналов в телекоммуникациях: аналоговая и цифровая модуляция.
15. Кодирование информации в компьютерных сетях: протоколы передачи данных.
16. Конструктивный подход теории алгоритмов.
17. Прimitивно-рекурсивные функции.
18. Операция «Марковской подстановки».
19. Схема нормального алгоритма Маркова
20. Принцип нормализации Маркова.
21. Основные понятия и принципы вычислимости функций.
22. Проблема останова: неразрешимость и связь с вычислимостью функций.
23. Формальное определение алгоритма и его связь с вычислимостью функций.
24. Машина Тьюринга как универсальная модель вычислений.
25. Функциональное представление алгоритма с помощью машины Тьюринга.
26. Понятие вычислимости и различные классы вычислимых функций.
27. Расширение машины Тьюринга: многоленточные и неоднозначные машины.
28. Теорема Поста: неразрешимость проблемы останова для машин Тьюринга.
29. Рекурсивные функции и их вычислимость с помощью машин Тьюринга.
30. Недетерминированная машина Тьюринга и связанные с ней проблемы вычислимости.
31. Понятие полиномиальной вычислимости и класс P.
32. Проблема коммивояжера и связь с вычислимостью функций.
33. Класс NP и связанные с ним проблемы вычислимости.

34. Теорема Кука-Левина: NP-полнота и связь с вычислимостью функций.
35. Рекурсия и итерация: различия и применение.
36. Анализ времени выполнения алгоритмов.
37. Анализ памяти и пространственной сложности алгоритмов.
38. Большая O-нотация и ее применение для оценки сложности алгоритмов.
39. Сортировка: алгоритмы и их сложность.
40. Поиск: алгоритмы и их сложность.
41. Интуитивное определение алгоритма и его временной и емкостной трудоемкости.
42. Формы представления алгоритмов. Методы разработки эффективных алгоритмов.
43. Реально-выполнимые и реально-невыполнимые алгоритмы.
44. Оценка трудоемкости. Рекуррентные теоремы.
45. Алгоритмы объединения множеств и их сравнение.
46. Верификация алгоритмов. Метод инварианта.
47. Задача сортировки и ее формы. Нижняя оценка трудоемкости методов, основанных на сравнениях.
48. Простые методы сортировки.
49. Быстрая сортировка Хоара. Поиск порядковых статистик.
50. Многофазная сортировка.
51. Цифровая сортировка и ее применение при лексикографическом упорядочивании строк.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Осокин, Александр Николаевич Теория информации : учебник для вузов / А. Н. Осокин, А. Н. Мальчуков. Электрон. дан. Москва : Юрайт, 2025 208 с (Высшее образование) URL: <https://urait.ru/bcode/561389> (дата обращения: 01.09.2025). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/561389> ISBN 978-5-534-16333-9 : 1119.00. [БИК ID: RU2fURAIT2f561389]
2. Куликов, А. А. Алгоритмы и структуры данных с использованием интерпретируемых языков [Электронный ресурс] : учебное пособие / Куликов А. А. Москва : РТУ МИРЭА, 2025 63 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/508432> ISBN 978-5-7339-2678-0. [БИК ID: RU-LAN-BOOK-508432]

Дополнительная литература:

1. Чернышев, Станислав Андреевич Принципы, паттерны и методологии разработки программного обеспечения : учебник для вузов / С. А. Чернышев. Электрон. дан. Москва : Юрайт, 2025 176 с (Высшее образование) URL: <https://urait.ru/bcode/567946> (дата обращения: 01.09.2025). Режим доступа: Электронно-библиотечная система Юрайт, для авториз. пользователей <https://urait.ru/bcode/567946> ISBN 978-5-534-14383-6 : 789.00. [БИК ID: RU2fURAIT2f567946]
2. Сыромятников, В. П. Структуры и алгоритмы обработки данных: Практикум [Электронный ресурс] / Сыромятников В. П. Москва : РТУ МИРЭА, 2020 244 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/163915>. [БИК ID: RU-LAN-BOOK-163915]

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

1. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/> (<http://library.fa.ru/files/elibfa.pdf>)
2. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>

3. Информационно-образовательный портал Финуниверситета: <https://org.fa.ru>
4. •Электронно-библиотечная система «Университетская библиотека ОНЛАЙН»
<http://biblioclub.ru/>
5. Электронный учебный курс <https://campus.fa.ru/course/view.php?id=12139>
6. Личный кабинет обучающегося <https://org.fa.ru>
7. Издательство «Открытые системы» <http://www.osp.ru>
8. Электронная библиотека (www.bibliotekar.ru).
9. Библиотечно-информационный комплекс Финансового университета при
Правительстве РФ <http://www.library.fa.ru>

10. Методические указания для обучающихся по освоению дисциплины

При изучении теоретического материала необходимо опираться на рабочую программу дисциплины, материалы лекций и литературу из основного списка.

Кроме этого, необходимо активно работать с Интернет-источниками и пособиями других авторов, помогающими усвоить материал отдельных разделов программы.

Необходимо конспектировать лекции, помечая сложные и непонятные моменты с тем, чтобы задать вопросы лектору в конце лекции или же на консультации.

При подготовке к семинарским занятиям необходимо изучить вопросы, вынесенные на самостоятельное изучение, так как семинарские занятия предполагают их обсуждение и дискуссию по теме; кроме того, задания для самостоятельной работы необходимы для того, чтобы успешно выполнить самостоятельные задания на семинарах.

Индивидуальные задания для работы на компьютере, файлы с выполненными заданиями необходимо хранить в личной сетевой папке в компьютерной сети вуза.

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Python 3.8
2. Windows
3. Astra Linux
4. Libre Office
5. Kaspersky

Современные профессиональные базы данных и информационные справочные системы:

1. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
2. Информационно-правовая система «Гарант»

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. **Учебная аудитория** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)

2. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)