

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Косенок Сергей Михайлович  
Должность: ректор  
Дата подписания: 11.06.2026 10:49:51  
Уникальный программный ключ:  
e3a68f3eaa1e62674b54f4998099d766b1f836

## Оценочные материалы для текущего контроля и промежуточной аттестации по дисциплине

Структуры и алгоритмы обработки данных, 4 семестр

Код, направление подготовки	27.03.04 Управление в технических системах
Направленность (профиль)	Инженерия автоматизированных, информационных и робототехнических систем
Форма обучения	очная
Кафедра-разработчик	автоматики и компьютерных систем
Выпускающая кафедра	автоматики и компьютерных систем

### Типовые задания для контрольной работы:

Контрольная работа проводится в виде теста. Вопросы с вариантами ответов формулируются в виде:

- пусть алгоритм представлен следующим псевдокодом ..., причем асимптотическая сложность фрагмента  $f(N, \text{other})$  составляет  $O(N)$ . Определите асимптотическую сложность алгоритма в зависимости от  $N$ ;
- представлено краткое описание некоторой физической (математической, алгоритмической) последовательности. Требуется выбрать/предложить наиболее подходящий алгоритм сортировки;
- представлена структура данных, предназначенная для решения некоторого класса задач. Требуется определить ее свойства, характеристики, способы внутренней организации;
- вопросы тестового вида, направленные на выявление знаний асимптотических свойств конкретных алгоритмов, структур данных.

### Типовые вопросы к зачету:

1. Структуры и алгоритмы обработки данных. Связь со структурным и объектно-ориентированным программированием. Абстрактные типы данных.
2. Сложность алгоритмов. Асимптотическая сложность.
3. Структура данных «Список» («Динамический массив», «Стек», «Очередь», «Дерево», «Множество», «Словарь», «Граф»). Операции. Способы реализации.
4. Алгоритм сортировки выбором (вставками, Шелла, подсчетом, поразрядной, Хоара, слиянием, пирамидальной). Основные свойства.
5. Хеширование. Хеш-функции. Коллизии.
6. Графы. Способы представления.
7. Алгоритмы на графах.
8. Поиск. Алгоритмы поиска в линейных структурах.
9. Алгоритмы поиска строк.
10. Деревья поиска. Сбалансированные деревья. Алгоритмы построения, добавления и удаления элемента.
11. Градиентный метод. Область применения.
12. Динамическое программирование.

13. P и NP задачи. Сводимость задач. Классы задач.

14. Методы решения NP задач.

### **Примерные практические задания промежуточной аттестации:**

Формулировка задания: заданы некоторые исходные данные, требуется сформулировать ответ, содержащий конкретные сведения, связанные с решением вычислительной задачи (свойством структуры, алгоритма).

Решаемая задача направлена на выявление следующих знаний, умений, навыков (по отдельности или в сочетаниях):

- формализации вычислительной задачи;
- умение выполнить сравнительный анализ;
- определить асимптотические характеристики алгоритма/структуры данных;
- выделить ключевые алгоритмические свойства программы;
- показать знания определенных структур, алгоритмов обработки данных.

Примеры некоторых задач промежуточной аттестации:

- построить сбалансированное дерево (АВЛ-дерево, красно-черное дерево, В-дерево и др.), которое получится в результате удаления (добавления) узла с ключом X;
- некоторый абстрактный тип данных (АТД) определяет структуру данных «очередь» («стек»). Функции Push, Pop и Count позволяют поместить узел в контейнер, извлечь узел и узнать количество узлов соответственно. Построить блок-схему алгоритма обхода бинарного дерева (графа) в ширину (глубину);
- дано дерево. Показать результат, который будет отображен на экране при работе функции прямого (обратного, симметричного) обхода;
- задан граф. Построить промежуточные матрицы смежности при работе алгоритма Флойда (Дейкстры), либо другие внутренние структуры при работе алгоритмов Крускала (Борувки, Прима и др.);
- указаны несколько способов реализации некоторого АТД, необходимо сравнить асимптотические характеристики ключевых алгоритмов работы с данным АТД;
- задано некоторое условие задачи, необходимо построить алгоритм (псевдокод, написать программу), которая реализует эту задачу.