

МИНОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ  
Директор института прикладной  
математики и компьютерных наук  
А.В. Замятин  
2022 г.



**Фонд оценочных средств по дисциплине**

**Алгоритмы и структуры данных**

по направлению подготовки

**02.03.03 Математическое обеспечение и администрирование информационных систем**

Направленность (профиль) подготовки :

**DevOps-инженерия в администрировании инфраструктуры ИТ-разработки**

Томск–2022

ФОС составил:

канд. физ.-мат. наук, доцент  
доцент кафедры теоретических основ информатики



А.Л. Фукс

Рецензент:

д-р техн. наук, профессор,  
профессор кафедры теоретических основ информатики



Ю.И. Косток

Фонд оценочных средств одобрен на заседании учебно-методической комиссии  
института прикладной математики и компьютерных наук (УМК ИПМКН)

Протокол от 24 февраля 2022 г. № 01

Председатель УМК ИПМКН,  
д-р техн. наук, профессор



С.П. Суценко

**Оценочные средства (ОС)** являются элементом системы оценивания сформированности компетенций у обучающихся в целом или на определенном этапе ее формирования.

ОС разрабатываются в соответствии с рабочей программой (РП) дисциплины и включает в себя набор оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине.

### **1. Компетенции и результаты обучения, формируемые в результате освоения дисциплины**

<b>Компетенция</b>	<b>Индикатор компетенции</b>	<b>Код и наименование результатов обучения</b> (планируемые результаты обучения, характеризующие этапы формирования компетенций)	<b>Критерии оценивания результатов обучения</b>			
			Отлично	Хорошо	Удовлетворительно	Неудовлетворительно

<p>ОПК-2 – Способен применять современный математический аппарат, связанный с проектированием, разработкой, реализацией и оценкой качества программных продуктов и программных комплексов в различных областях человеческой деятельности</p>	<p>ИОПК-2.1 Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем ИОПК-2.2 Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий</p>	<p>ОР-2.1.1. Знать методы исследования и теорию сложности алгоритмов ОР-2.2.1. Уметь применять методы исследования алгоритмов при создании и анализе программ</p>	<p>Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий Сформированные систематические знания методов исследования и теорию сложности алгоритмов и умение. Сформированные систематические умения применять методы исследования алгоритмов при создании и анализе</p>	<p>Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем с неточностями Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий с неточностями Сформированные, но содержащие отдельные пробелы знания методов исследования и теорию сложности алгоритмов и умение. Сформированные, но содержащие отдельные пробелы</p>	<p>Использует методы построения и анализа алгоритмов при проектировании и разработке программных систем с ошибками Использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий с ошибками Фрагментарные знания методов исследования и теорию сложности алгоритмов, частично освоенное умение. Частично освоенное умение применять методы исследования алгоритмов</p>	<p>Не использует методы построения и анализа алгоритмов при проектировании и разработке программных систем Не использует фундаментальные знания для реализации алгоритмов пригодных для практического применения в области информационных систем и технологий Отсутствие знаний методов исследования и теорию сложности алгоритмов и умений. Отсутствие умений применять методы исследования алгоритмов при создании и анализе программ.</p>
--	---	---	--	---	--	--

	<p>ИОПК-2.3 Разрабатывает алгоритмы и программы при решении профессиональной деятельности задач</p>	<p>ОР-2.3.1. Знать принципы разработки эффективных алгоритмов, набор базовых алгоритмов и базовые структуры данных  ОР-2.3.2. Уметь применять полученные знания при создании и анализе программ</p>	<p>Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности Сформированные систематические знания методов исследования и теорию сложности алгоритмов и умение. Сформированные систематические умения применять методы исследования алгоритмов при создании и анализе программ.</p>	<p>Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности с неточностями Сформированные, но содержащие отдельные пробелы знания методов исследования и теорию сложности алгоритмов и умение. Сформированные, но содержащие отдельные пробелы умения применять методы исследования алгоритмов при создании и анализе программ.</p>	<p>Разрабатывает алгоритмы и программы при решении задач профессиональной деятельности с ошибками Фрагментарные знания методов исследования и теорию сложности алгоритмов, частично освоенное умение. Частично освоенное умение применять методы исследования алгоритмов при создании и анализе программ.</p>	<p>Не разрабатывает алгоритмы и программы при решении задач профессиональной деятельности  Отсутствие знаний методов исследования и теорию сложности алгоритмов и умений.  Отсутствие умений применять методы исследования алгоритмов при создании и анализе программ.</p>
--	---	---	---	--	---	--

## 2. Этапы формирования компетенций и виды оценочных средств

№	Этапы формирования компетенций (разделы дисциплины)	Код и наименование результатов обучения	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1.	Методы анализа алгоритмов	ОР-2.1.1, ОР-2.2.1, ОР-2.3.1,	Вопросы по теории раздела 1.
2.	Поиск и сортировка	ОР-2.1.1, ОР-2.2.1, ОР-2.3.1,	Вопросы по теории раздела 2. Задания и вопросы к лабораторной работе №1
3.	Структуры данных	ОР-2.1.1, ОР-2.2.1, ОР-2.3.1,	Вопросы по теории раздела 3. Задания и вопросы к лабораторной работе №2
4.	Оптимизационные алгоритмы и задачи на графах	ОР-2.1.1, ОР-2.2.1, ОР-2.3.1,	Вопросы по теории раздела 4. Задания и вопросы к лабораторной работе №3
5.	Задача поиска подстроки	ОР-2.1.1, ОР-2.2.1, ОР-2.3.1,	Вопросы по теории раздела 5.
6.	Теория алгоритмов и NP-полные задачи	ОР-2.1.1, ОР-2.2.1, ОР-2.3.1,	Вопросы по теории раздела 6.

## 3. Типовые контрольные задания или иные материалы, необходимые для оценки образовательных результатов обучения

3.1. Типовые задания для проведения текущего контроля успеваемости по дисциплине (студенты должны самостоятельно подготовить и сдать программы на лабораторных занятиях):

1. Лабораторная работа «Эффективные алгоритмы внутренней сортировки».
2. Лабораторная работа «Структуры данных».
3. Лабораторная работа «Задачи на графах».

Примеры заданий для лабораторных работ:

1. Эффективные алгоритмы внутренней сортировки.

Цель работы: практическая реализация и исследование эффективных алгоритмов сортировки.

Описание: реализовать шаблонный класс с 2-3 статическими методами сортировки и проверить их работу на больших массивах разных типов.

2. Хеш-таблицы и списки.

Цель работы: практическая реализация и использование двух контейнерных классов.

Описание: реализовать классы «Список строк» и «Хеш-таблица (метод цепочек, хранит строки), использовать хеш-таблицу в задаче поиска уникальных строк в большом массиве, исследовать трудоемкость.

3. Задачи на графах.

Цель работы: реализовать и использовать несколько алгоритмов на графах для решения задачи начального разбиения набора точек на плоскости на заданное число кластеров.

Описание: реализовать алгоритмы построения матриц смежности и весов, выделения минимального остова, выделения компонент связности и использовать их при решении исходной задачи.

3.2. Типовые задания для проведения промежуточной аттестации по дисциплине «Алгоритмы и структуры данных» (экзамен проводится в 2 этапа – по разделам 1-3 в середине семестра и по разделам 4-6 в конце семестра, студенты получают 3-4 вопроса и отвечают устно):

**Раздел 1. Методы анализа алгоритмов.**

1. Рекуррентные и рекурсивные алгоритмы.
2. Трудоемкость и емкостная сложность.
3. Теоремы о трудоемкости.

**Раздел 2. Поиск и сортировка.**

4. Дихотомический поиск в упорядоченном массиве.
5. Простые алгоритмы сортировки.
6. Сортировка слиянием – рекурсивный и рекуррентный варианты.
7. Сортировка Шелла.
8. Бинарная куча и просеивание элементов.
9. Пирамидальная сортировка.
10. Быстрая сортировка: идея, трудоемкость в среднем и наихудшем.
11. Быстрая сортировка: идея, разделение опорным элементом, варианты с одним или двумя рекурсивными вызовами, емкостная сложность.

**Раздел 3. Структуры данных.**

12. Хеширование. Идея, метод цепочек.
13. Хеширование. Идея, метод открытой адресации. Варианты реализации.
14. Случайное бинарное дерево. Построение, поиск, удаление элементов.
15. AVL-деревья. Деревья Фибоначчи. Трудоемкость поиска. Структура вершины.
16. Добавление вершины к AVL-дереву.
17. Удаление вершин из AVL-дерева.
18. B-деревья. Структура вершины. Поиск значения. Оценки трудоемкости.

**Раздел 4. Оптимизационные алгоритмы и задачи на графах.**

19. Методы решения задач дискретной оптимизации.
20. Поиск в глубину.
21. Поиск в ширину.
22. Выделение минимального остова. Алгоритм Прима.
23. Выделение минимального остова. Алгоритм Крускала.
24. Поиск кратчайших путей. Алгоритм Дейкстры.

25. Поиск кратчайших путей. Алгоритм Флойда-Уоршола.
26. Варианты поиска оптимального маршрута коммивояжера.
27. Маршрут коммивояжера на основе минимального остова.
28. Задача коммивояжера. Алгоритм ближайшего города.
29. Алгоритм имитации отжига.
30. Муравьиный алгоритм.
31. Генетический алгоритм для задачи коммивояжера.
32. Задача раскраски графов. Минимальная раскраска графа по методу ветвей и границ.
33. Задача раскраски графов. Алгоритмы, основанные на степенях вершин.
34. Алгоритмы раскраски графов, основанные на склеивании вершин.
35. Раскраска транзитивно-ориентируемых графов.
36. Выделение двусвязных компонент.
37. Выделение сильно связанных компонент.

#### **Раздел 5. Задача поиска подстроки**

38. Алгоритм Бойера-Мура-Хопгуда.
39. Алгоритм на основе конечного автомата.
40. Алгоритм Кнута-Морриса-Пратта.
41. Алгоритм Рабина-Карпа.
42. Алгоритм Ахо-Корасик.

#### **Раздел 6. Теория алгоритмов и NP-полные задачи**

43. Недетерминированные и детерминированные алгоритмы.
44. NP-полные и NP-трудные задачи.
45. Основные идеи доказательства NP-полноты задачи выполнимости булевых формул.

### **4. Методические материалы, определяющие процедуры оценивания образовательных результатов обучения**

4.1. Методические материалы для оценки текущего контроля успеваемости по дисциплине.

Текущий контроль успеваемости проводится во время сдачи лабораторных работ. Каждая работа оценивается по пятибалльной системе по следующим параметрам:

- полнота реализации программы,
- ответы на вопросы по переменным, функциям, классам программы
- ответы на вопросы по теории из соответствующего раздела курса
- умение исправлять ошибки и оперативно вносить изменения в программу.

4.2. Методические материалы для проведения промежуточной аттестации по дисциплине.

Видом промежуточной аттестации является экзамен. Итоговая оценка формируется как средневзвешенная по результатам сдачи лабораторных работ и устного собеседования на экзамене. Экзамен проводится в 2 этапа – в середине и в конце семестра. Компетенции



формируются на протяжении всего курса, а итоговая оценка показывает степень их освоения.