

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
(КГУ)

Кафедра «Фундаментальная математика»



УТВЕРЖДАЮ:
Первый проректор
/ Т. Р. Змызгова/
сен 2022 г.

Рабочая программа учебной дисциплины

АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ

образовательной программы высшего образования –
программы специалитета

01.05.01 Фундаментальные математика и механика
Направленность (профиль) «Математическое и программное обеспечение
информационных систем»

Форма обучения: очная

Курган 2022

Рабочая программа дисциплины «Алгоритмы и структуры данных» составлена в соответствии с учебными планами по программе специалитета «Фундаментальная математика и механика» (Математическое и программное обеспечение информационных систем), утвержденной:
- для очной формы обучения «30» 08 2022 года,

Рабочая программа дисциплины одобрена на заседании кафедры «Фундаментальной математики» «31» 08 2022 года, протокол № 1.

Рабочую программу составил:
К. пед. наук, доцент кафедры
«Фундаментальная математика»



А.В. Чернышова

Согласовано:

Заведующий кафедрой
«Фундаментальная математика»



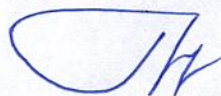
М.В. Гаврильчик

Специалист по учебно-методической
работе учебно-методического отдела



Г.В. Казанкова

Начальник управления
образовательной деятельности



И.В. Григоренко

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 3 зачетных единицы трудоемкости (108 академических часов)

Очная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		4
Аудиторные занятия (контактная работа с преподавателем), всего часов	46	46
в том числе:		
Лекции	16	16
Лабораторные работы	30	30
Практические занятия	-	-
Самостоятельная работа, всего часов	62	62
в том числе:		
Подготовка к зачету	18	18
Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины)	44	44
Вид промежуточной аттестации	зачет	зачет
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	108	108

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Алгоритмы и структуры данных» относится к части, формируемой участниками образовательных отношений блока Б1.

Изучение дисциплины базируется на знаниях, умениях и навыках, приобретенных студентами, при изучении следующих дисциплин: Информатика, Языки программирования, Дискретная математика, Математическая логика.

Результаты обучения по дисциплине необходимы для изучения дисциплин: «Методы принятия оптимальных решений», «Проектирование информационных систем» и выполнения выпускной квалификационной работы.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Цель освоения дисциплины: формирование знаний и практических навыков разработки и анализа алгоритмов программ и выбора структур данных.

Задачи дисциплины: изучение основ теории структур, методов представления данных на логическом (абстрактном) и физическом (машинном) уровнях, методов разработки эффективных алгоритмов обработки структур данных.

Компетенции, формируемые в результате освоения дисциплины:

- способен разрабатывать, внедрять и адаптировать прикладное программное обеспечение (ПК-1);
- способен выполнять работы по обслуживанию информационно-коммуникационной системы (ПК-2);
- способен выполнять работы по созданию (модификации) и сопровождению информационных систем, автоматизирующих задачи организационного управления и бизнес-процессы (ПК-3).

В результате изучения дисциплины обучающийся должен *знать:*

- современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности (ПК-1, ПК-2, ПК-3);

уметь:

- использовать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности; разрабатывать алгоритмы и программы, пригодные для практического применения (ПК-1, ПК-2, ПК-3);

владеть:

- современными информационными технологиями и программными средствами, в том числе отечественного производства при решении задач профессиональной деятельности; методами разработки алгоритмов и программ, пригодных для практического применения (ПК-1, ПК-2, ПК-3).

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Учебно-тематический план.

Рубеж	Номер раздела, темы	Наименование раздела, темы	Количество часов контактной работы с преподавателем	
			Лекции	Лабораторные работы
Рубеж 1	1	Типы данных и базовые структуры данных	2	4
	2	Линейные структуры данных	2	4
	3	Нелинейные структуры данных	4	5
		Рубежный контроль №1	-	1
Рубеж 2	4	Алгоритмы сортировки данных	2	4
	5	Алгоритмы поиска данных	2	4
	6	Алгоритмы на графах	4	7
		Рубежный контроль №2	-	1
Всего:			16	30

4.2. Содержание лекционных занятий

Тема 1. Типы данных и базовые структуры данных

Алгоритмы и данные. Свойства алгоритма. Анализ сложности алгоритма. Семантика, синтаксис, прагматика. Структура данных. Структуры хранения данных: вектор, список, сеть. Массивы. Структуры данных массивов. Структуры хранения массивов. Строки. Операции над строками. Записи. Операции над записями. Множества.

Тема 2. Линейные структуры данных

Списки. Структура и классификация списков. Операции над линейными списками. Применение списков. Стеки. Структура стека. Операции над стеками. Применение стеков. Очереди. Деки. Операции над очередями и деками. Применение очередей и деков.

Тема 3. Нелинейные структуры данных

Алгоритм преобразования m -арного дерева в бинарное дерево. Представление деревьев в памяти ЭВМ. Идеально-сбалансированное бинарное дерево. Бинарные (двоичные) деревья поиска. Сбалансированные деревья поиска. Сбалансированные AVL-деревья поиска. Рандомизированные деревья поиска. Оптимальные деревья поиска. Операции над деревьями. В-деревья. Операции над В-деревьями.

Тема 4. Алгоритмы сортировки данных

Основные понятия и классификация алгоритмов сортировки. Внутренняя сортировка. Метод прямого включения. Метод прямого выбора. Метод прямого обмена. Шейкерная сортировка. Быстрые (улучшенные) методы сортировки. Метод Шелла. Метод пирамиды. Метод Хоара. Поразрядная сортировка.

Внешняя сортировка. Прямое слияние. Естественное слияние. Сбалансированное многопутевое слияние. Многофазная сортировка. Каскадная сортировка.

Тема 5. Алгоритмы поиска данных

Классификация алгоритмов поиска. Поиск в последовательно организованных структурах. Последовательный поиск. Двоичный и Фибоначчиев поиск. Интерполяционный поиск. Индексно-последовательный поиск. Поиск в деревьях. Случайные двоичные деревья поиска. Оптимальные двоичные деревья поиска. Сбалансированные деревья поиска. Хеширование. Хеш-функции. Разрешение коллизий методом цепочек. Разрешение коллизий методом открытой адресации. Идеальное хеширование.

Тема 6. Алгоритмы на графах

Основные понятия и определения. Представление графов. Матрица смежности. Векторы смежности. Списки смежности. Матрица инцидентности. Пути в графе. Путевая матрица (матрица достижимости). Кратчайшие пути. Алгоритм Дейкстры. Алгоритм Флойда. Остовные деревья графа. Обходы графов. Поиск в глубину. Поиск в ширину. Остовное дерево наименьшей стоимости. Алгоритм Прима. Алгоритм Крускала. Упорядочение графа

4.3. Лабораторные занятия

Номер раздела, темы	Наименование раздела, темы	Наименование лабораторной работы	Норматив времени, час.
			Спая форма обучения
1	Типы данных и базовые структуры данных	Структуры данных списки, очереди	4
2	Линейные структуры данных	Структуры данных стеки, деки	4
3	Нелинейные структуры данных	Двоичные деревья поиска	5
		Рубежный контроль №1	1
4	Алгоритмы сортировки данных	Методы упорядочивания данных.	4
5	Алгоритмы поиска данных	Поиск в последовательно организованных структурах.	4
6	Алгоритмы на графах	Алгоритм определения кратчайших путей на графах	7
		Рубежный контроль №2	1
Всего:			30

4.4 Практические занятия

Практические занятия не предусмотрены учебным планом.

4.5. Курсовая работа

Не предусмотрена.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Лекционный курс основывается на методе обучения, использующем технологию, при которой студенты конспектируют теоретический материал, участвуют в опросах и дискуссиях. В этом случае задействованы зрительная, слуховая, моторная и ассоциативная виды памяти.

При прослушивании лекций рекомендуется в конспекте отмечать все важные моменты, на которых заостряет внимание преподаватель, в частности те, которые направлены на качественное выполнение соответствующей лабораторной работы.

Преподавателем запланировано использование при чтении лекций технологии учебной дискуссии. Поэтому рекомендуется фиксировать для себя интересные моменты с целью их активного обсуждения на дискуссии в конце лекции.

Залогом качественного выполнения лабораторных работ и занятий является самостоятельная подготовка к ним накануне путем повторения материалов лекций. Рекомендуется подготовить вопросы по неясным моментам и обсудить их с преподавателем в начале занятия.

Преподавателем запланировано применение на лабораторных занятиях технологий развивающейся кооперации, коллективного взаимодействия, разбора конкретных ситуаций.

Для текущего контроля успеваемости по очной форме обучения преподавателем используется балльно-рейтинговая система контроля и оценки академической активности. Поэтому настоятельно рекомендуется тщательно прорабатывать материал дисциплины при самостоятельной работе, участвовать во всех формах обсуждения и взаимодействия, как на лекциях, так и на лабораторных занятиях в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Выполнение самостоятельной работы подразумевает самостоятельное изучение разделов дисциплины, подготовку к лабораторным занятиям, к рубежным контролям, подготовку к зачёту.

Рекомендуемая для очной формы обучения трудоемкость самостоятельной работы представлена в таблице:

Рекомендуемый режим самостоятельной работы

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.
	очная форма обучения
Самостоятельное изучение тем дисциплины:	28
Алгоритмы решения задач выбора	4
Красно-черные деревья	3
Базовое восходящее скошенное дерево	4
Нисходящие скошенные деревья	3

Поиск кратчайшего отрицательно взвешенного пути на графе	4
Алгоритм нахождения максимального пути на графе	3
Алгоритм Беллмана-Мура нахождения кратчайшего пути на графе.	4
Определение экстремальных путей на графах. Метод Шимбелла	3
Подготовка к лабораторным занятиям (по 1 часу на каждое занятие)	15
Подготовка к рубежным контролям (по 0,5 часа на каждый рубеж)	1
Подготовка к зачёту	18
Всего:	62

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности студентов в КГУ (для очной формы обучения).
2. Отчеты студентов по лабораторным работам.
3. Банк заданий к рубежным контролям № 1, № 2.
4. Вопросы к зачёту.

6.2. Система балльно-рейтинговой оценки работы студентов по дисциплине

№	Наименование	Содержание					
		Распределение баллов					
		Вид учебной работы:	Посещение лекций	Выполнение и защита отчетов по лабораторным работам	Рубежный контроль №1	Рубежный контроль №2	Зачёт
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения студентов на первом учебном занятии)	Балльная оценка:	До 16	До 30	До 12	До 12	До 30
	Примечания:	8 лекций по 2 балла	15 лабораторных занятий по 2 балла	На 7 лабораторном занятии	На 15 лабораторном занятии		

2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и зачета	60 и менее баллов – незачёт 61... 100 – зачтено
3	Критерии допуска к промежуточной аттестации, возможности получения автоматического зачета (экзаменационной оценки) по дисциплине, возможность получения бонусных баллов	<p>Для допуска к промежуточной аттестации студент должен выполнить все работы рубежного контроля и лабораторные работы и набрать не менее 50 баллов.</p> <p>Для получения экзаменационной оценки «автоматически» студенту необходимо набрать следующее минимальное количество баллов:</p> <ul style="list-style-type: none"> - 61 балл для получения зачета «автоматически». <p>По согласованию с преподавателем студенту могут быть добавлены дополнительные (бонусные) баллы за активное участие в научной и методической работе, оригинальность принятых решений в ходе выполнения заданий текущего и рубежного контроля, за участие в значимых учебных и внеучебных мероприятиях кафедры.</p>
4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) обучающихся для получения недостающих баллов в конце семестра	<p>В случае если к промежуточной аттестации набрана сумма менее 50 баллов, студенту необходимо выполнить дополнительные задания, до конца последней (зачетной) недели семестра. При этом необходимо проработать материал всех пропущенных лекционных и лабораторных занятий.</p> <p>Формы дополнительных заданий (назначаются преподавателем):</p> <ul style="list-style-type: none"> - прохождение рубежного контроля (баллы в зависимости от рубежа); - выполнение лабораторных работ – до 15 баллов. <p>Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.</p>

6.3. Процедура оценивания результатов освоения дисциплины

Рубежные контроли проводятся в форме письменного тестирования, зачёт в устной форме виде ответов на вопросы в билетах к зачёту.

Перед проведением рубежного контроля преподаватель прорабатывает со студентами основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии.

Варианты заданий для рубежных контролей № 1, № 2 состоят из 12 вопросов по 1 баллу каждый.

На каждую работу при рубежном контроле студенту отводится время не менее 40 минут.

Преподаватель оценивает в баллах результаты рубежных контролей каждого студента по количеству правильных ответов и заносит в ведомость учета текущей успеваемости.

Билет к зачету состоит из 1 вопроса. Вопросы к зачёту доводятся до студентов на последней лекции в семестре. На подготовку ответа по вопросам билета к зачёту студенту отводится 1 астрономический час. Вопрос оценивается в 30 баллов.

Результаты текущего контроля успеваемости и зачёта заносятся преподавателем в зачётную ведомость, которая сдаётся в организационный отдел ин-

ститута в конце зачётной недели, а также выставляется в зачетную книжку студента.

6.4 Примеры оценочных средств для рубежных контролей, зачёта

6.4.1 Примеры заданий для рубежного контроля №1

1. Что называется алгоритмом?

1. Система формальных правил, четко и однозначно определяющая процесс решения поставленной задачи в виде конечной последовательности действий или операций.

2. Набор команд.

3. Последовательность действий.

4. Совокупность правил и инструкций, оформленных в соответствии стандарта.

2. Какие характеристики алгоритма?

1. Время работы алгоритма.

2. Количество операторов.

3. Количество переходов.

4. Объем памяти компьютера.

3. Что называется вычислительной сложностью (трудоемкостью) алгоритма $T(n)$, где n – размер задачи?

1. Объем памяти компьютера, необходимый для его выполнения.

2. Количество операций, необходимых для его выполнения.

3. Количество переходов.

4. Количество операторов.

4. Что называется ёмкостной сложностью алгоритма?

1. Количество операций, необходимых для его выполнения.

2. Количество переходов.

3. Количество операторов.

4. Объем памяти компьютера, требуемый для реализации алгоритма.

5. Что является свойством алгоритма?

1. Финитность.

2. Динамичность.

3. Гибкость.

4. Эффективность.

6. Какие случаи проведения анализа алгоритма существуют?

1. Сравнение алгоритмов.

2. Оценка производительности.

3. Установка значений параметров алгоритма.

4. Просмотр алгоритма.

7. Что называется O-нотацией?

1. Математическое выражение.

2. Свойство функции.

3. Математическая запись, позволяющая отбрасывать подробности при анализе алгоритмов.

4. Алгоритмическая запись.

8. Каковы причины использования O-нотации при оценке функции вычислительной сложности алгоритма?

1. Не учитывать вклад малых слагаемых в математических формулах.
2. Удобство восприятия.
3. Простота записи.
4. Классифицировать алгоритмы согласно верхней границе их общего времени выполнения.

9. Что означает выражение $O(1)$?

1. Время выполнения алгоритма равно 1.
2. Константное время выполнения алгоритма, не зависящее от n .
3. Ошибка в алгоритме.
4. Нулевой алгоритм.

10. Что называется данными?

1. Сведения, полученные путем измерения наблюдения логических или арифметических операций и представленные в форме, пригодной для хранения, передачи и обработки.

2. Множество числовых величин.
3. Совокупность сообщений.
4. Характеристики случайных величин.

11. Что называется типом данных?

1. Набор данных, определяющий диапазон значений.
2. Набор данных, определяющий способ представления.
3. Набор данных, определяющий допустимые операции.
4. Набор данных, определяющий диапазон возможных значений, способ представления, допустимые операции над данными.

12. Что называется структурой данных?

1. Набор правил и ограничений, определяющих связи между отдельными элементами данных.
2. Набор правил и ограничений, определяющих связи между отдельными группами данных.
3. Набор правил и ограничений, определяющих связи между отдельными элементами и группами данных.
4. Некоторую иерархию данных.

6.4.2 Примеры заданий для рубежного контроля №2

1. Что такое функция сложности алгоритма?

1. Функция, характеризующая структуру алгоритма.
2. Функция, характеризующая объем алгоритма.
3. Функция, выражающая относительную скорость алгоритма в зависимости от некоторой переменной (переменных).
4. Функция, характеризующая эффективность алгоритма.

2. Сортировка – упорядочение элементов множества в возрастающем или убывающем порядке?

1. Да.
2. Нет.

3. Какова сущность сортировки выбором?

1. В неупорядоченном списке выбирается и отделяется от остальных наименьший элемент. Исходный список оказывается измененным. Измененный список принимается за исходный и процесс продолжается до тех пор, пока все элементы не будут выбраны.

2. Из неупорядоченной последовательности элементов выбирается поочередно каждый элемент, сравнивается с предыдущим, уже упорядоченным, и помещается на соответствующее место.

3. Анализируются первые элементы обоих массивов. Меньший элемент переписывается в новый массив. Оставшийся элемент последовательно сравнивается с элементами из другого массива. В новый массив после каждого сравнения попадает меньший элемент. Процесс продолжается до исчерпания элементов одного из массивов. Затем остаток другого массива дописывается в новый массив.

4. Элементы последовательно сравниваются между собой и меняются местами в случае, если предшествующий элемент больше последующего.

4. Какова сущность сортировки вставкой?

1. В неупорядоченном списке выбирается и отделяется от остальных наименьший элемент. Исходный список оказывается измененным. Измененный список принимается за исходный и процесс продолжается до тех пор, пока все элементы не будут выбраны.

2. Из неупорядоченной последовательности элементов выбирается поочередно каждый элемент, сравнивается с предыдущим, уже упорядоченным, и помещается на соответствующее место.

3. Анализируются первые элементы обоих массивов. Меньший элемент переписывается в новый массив. Оставшийся элемент последовательно сравнивается с элементами из другого массива. В новый массив после каждого сравнения попадает меньший элемент. Процесс продолжается до исчерпания элементов одного из массивов. Затем остаток другого массива дописывается в новый массив.

4. Элементы последовательно сравниваются между собой и меняются местами в случае, если предшествующий элемент больше последующего.

5. Какова сущность сортировки слиянием?

1. В неупорядоченном списке выбирается и отделяется от остальных наименьший элемент. Исходный список оказывается измененным. Измененный список принимается за исходный и процесс продолжается до тех пор, пока все элементы не будут выбраны.

2. Из неупорядоченной последовательности элементов выбирается поочередно каждый элемент, сравнивается с предыдущим, уже упорядоченным, и помещается на соответствующее место.

3. Анализируются первые элементы обоих массивов. Меньший элемент переписывается в новый массив. Оставшийся элемент последовательно сравнивается с элементами из другого массива. В новый массив после каждого сравнения попадает меньший элемент. Процесс продолжается до исчерпания элементов одного из массивов. Затем остаток другого массива дописывается в новый массив.

4. Элементы последовательно сравниваются между собой и меняются местами в случае, если предшествующий элемент больше последующего.

6. Какова сущность сортировки обменом?

1. В неупорядоченном списке выбирается и отделяется от остальных наименьший элемент. Исходный список оказывается измененным. Измененный список принимается за исходный и процесс продолжается до тех пор, пока все элементы не будут выбраны.

2. Из неупорядоченной последовательности элементов выбирается поочередно каждый элемент, сравнивается с предыдущим, уже упорядоченным, и помещается на соответствующее место.

3. Анализируются первые элементы обоих массивов. Меньший элемент переписывается в новый массив. Оставшийся элемент последовательно сравнивается с элементами из другого массива. В новый массив после каждого сравнения попадает меньший элемент. Процесс продолжается до исчерпания элементов одного из массивов. Затем остаток другого массива дописывается в новый массив.

4. Элементы последовательно сравниваются между собой и меняются местами в случае, если предшествующий элемент больше последующего.

7. Внутренняя сортировка – сортировка данных, элементы которых располагаются в оперативной памяти?

1. Да.
2. Нет.

8. Какие сложные методы сортировки?

1. Метод фон Неймана.
2. Сортировка Шелла.
3. Сортировка Хоара.
4. Метод «пузырька».

9. Внешняя сортировка – сортировка данных, элементы которых располагаются на внешнем носителе?

1. Да.
2. Нет.

10. Какова сущность поиска в глубину?

1. Поиск начинаем с некоторой фиксированной вершины v_0 . Затем выбираем произвольную вершину u , смежную с v_0 , и повторяем просмотр от u . Предположим, что находимся в некоторой вершине v . Если существует ещё не просмотренная вершина u , $u-v$, то рассматриваем её, затем продолжаем поиск с нее. Если не просмотренной вершины, смежной с v , не существует, то возвращаемся в вершину, из которой попали в v , и продолжаем поиск (если $v=v_0$, то поиск закончен);

2. Поиск начинаем с некоторой фиксированной вершины v_0 . Затем выбираем произвольную вершину u , смежную с v_0 , и повторяем просмотр от u . Предположим, что находимся в некоторой вершине v . Если существует ещё не просмотренная вершина u , $u-v$, то рассматриваем её, затем продолжаем поиск с нее. Если не просмотренной вершины, смежной с v , не существует, то возвра-

щаемся в вершину, из которой попали в v , и продолжаем поиск (если $v=u$, то поиск закончен);

3. Поиск начинаем с некоторой фиксированной вершины v_0 . Затем выбираем произвольную вершину u , смежную с v_0 , и повторяем просмотр от u . Предположим, что находимся в некоторой вершине v . Если существует ещё не просмотренная вершина u , то рассматриваем её, затем продолжаем поиск с нее. Если не просмотренной вершины, смежной с v , не существует, то возвращаемся в вершину, из которой попали в v , и продолжаем поиск (если $v=v_0$, то поиск закончен).

4. Поиск начинаем с некоторой фиксированной вершины v_0 . Затем выбираем произвольную вершину u , смежную с v_0 , и повторяем просмотр от u . Предположим, что находимся в некоторой вершине v . Если существует ещё не просмотренная вершина u , то рассматриваем её, затем продолжаем поиск с нее. Если не просмотренной вершины, смежной с v , не существует, то возвращаемся в вершину, из которой попали в v , и продолжаем поиск (если $v=v_0$ и $v=u$ то поиск закончен).

11. Что является стандартным способом устранения рекурсии при поиске в глубину?

1. Использование массива.
2. Использование очереди.
3. Использование стека.
4. Использование циклического списка.

12. Что используется при поиске в ширину?

1. Массив.
2. Очередь
3. Стек.
4. Циклический список

6.4.3 Примерный перечень вопросов для зачёта

1. Алгоритм. Свойства, виды и структуры алгоритмов. Этапы решения задач на ЭВМ.
2. Данные. Типы данных. Структуры хранения данных: вектор, список и сеть.
3. Массивы. Структуры данных и структуры хранения массивов.
4. Сложность алгоритма. Функции сложности алгоритмов. Оценка эффективности алгоритмов.
5. Линейный список. Операции над линейным списком. Применение линейного списка.
6. Стек. Структура хранения стека. Операции над стеками. Применение стека.
7. Очереди. Структура хранения очереди. Операции над очередями. Применение очереди.
8. Нелинейные структуры данных. Представление m -арного дерева бинарным деревом. Идеально сбалансированное бинарное дерево.
9. Бинарные деревья поиска. Сбалансированные деревья поиска.

10. Операции над деревьями. Алгоритмы обхода дерева.
11. В- деревья. В+ – деревья. Операции над В – деревьями.
12. Простые методы сортировки. Методы прямого включения, прямого обмена, прямого выбора.
13. Сложные методы сортировки. Метод Шелла. Сортировка с помощью дерева (пирамиды). Быстрая сортировка Хоара.
14. Внешняя сортировка. Методы прямого и естественного слияния.
15. Методы поиска, основанные на сравнении ключей: последовательный, бинарный, интерполяционный.
16. Методы поиска, основанные на цифровых свойствах ключей: хеширование.
17. Основные определения теории графов. Представление графов.
18. Алгоритмы поиска в глубину и ширину на графах.
19. Алгоритм построения остовного дерева методом Крускала.
20. Алгоритм построения остовного дерева методом Прима.
21. Алгоритмы нахождения кратчайшего пути на графе. Алгоритм Дейкстры.
22. Алгоритмы нахождения кратчайшего пути на графе. Алгоритм Флойда.

6.5. Фонд оценочных средств

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1. Основная учебная литература

1. Мясникова Н. А. Алгоритмы и структуры данных. Учебное пособие. – Москва: Кнорус, 2020. – 186 с. – (сер. Бакалавриат).
2. Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ. – Москва: Вильямс, 2018. – 1328 с.
3. Седжвик Р. Алгоритмы на С++ – Москва: Изд-во «Вильямс», 2016 – 1056 с.
4. Маер, А. В. Введение в структуры и алгоритмы обработки данных : учебное пособие / А. В. Маер, О. С. Черепанов. — Курган : КГУ, 2021. — 107 с. — ISBN 978-5-4217-0576-5. — Текст : электронный // Лань : электронно-библиотечная система.

7.2. Дополнительная учебная литература

1. Анашкина Н.В. Технологии и методы программирования. – М.: Издательский центр «Академия», 2012. – 384 с.
2. Давыдов В.Г. Технологии программирования. С++. – Санкт-Петербург: БХВ – Петербург, 2005. – 672 с.
3. Уайс, М. А. Организация структур данных и решение задач на С++ [Текст] / Уайс М. : А. пер. с англ. – М. : ЭКОМ Паблишерз, 2008. – 896 с.

4. Хусаинов Б.С. Структуры и алгоритмы обработки данных. Примеры на языке Си: Учебное пособие. – Финансы и статистика, 2004. – 464 с

5. Аксёнова Е.А., Соколов А.В. Алгоритмы и структуры данных на C++: Учебное пособие. – Петрозаводск: Изд-во ПетрГУ, 2008.- 81 с. URL: <http://window.edu.ru/resource/576/63576>.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

1. Семахин А.М. Алгоритмы и структуры данных. Методические указания к выполнению лабораторных и курсовых работ для студентов направления 231000.62«Программная инженерия». Курган, КГУ, 2012. – 64 с.

2. Алгоритмы и структуры данных. Методические указания к выполнению курсовых работ для студентов направления 231000.62«Программная инженерия». Курган, КГУ, 2014. – 74 с.

9. РЕСУРСЫ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. Федеральный портал «Российское образование» URL: <http://www.edu.ru/>

2. Сайт дистанционного обучения в НОУ «ИНТУИТ». URL: <http://www.intuit.ru/>

10. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

1. ЭБС «Лань»
2. ЭБС «Консультант студента»
3. ЭБС «Znanium.com»
4. «Гарант» - справочно-правовая система

11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение по реализации дисциплины осуществляется в соответствии с требованиями ФГОС ВО по данной образовательной программе..

12. ДЛЯ ОБУЧАЮЩИХСЯ С ИСПОЛЬЗОВАНИЕМ ДИСТАНЦИОННЫХ ОБРАЗОВАТЕЛЬНЫХ ТЕХНОЛОГИЙ

При использовании электронного обучения и дистанционных образовательных технологий (далее ЭО и ДОТ) занятия полностью или частично проводятся в режиме онлайн. Объем дисциплины и распределение нагрузки по видам работ соответствует п. 4.1. Распределение баллов соответствует п. 6.2 либо может быть изменено в соответствии с решением кафедры, в случае перехода на ЭО и ДОТ в процессе обучения. Решение кафедры об используемых технологиях и системе оценивания достижений обучающихся принимается с учетом мнения ведущего преподавателя и доводится до обучающихся.

Аннотация к рабочей программе дисциплины
«Алгоритмы и структуры данных»

образовательной программы высшего образования –
программы специалитета

01.05.01 Фундаментальные математика и механика

**Направленность (профиль) «Математическое и программное обеспечение
информационных систем»**

Трудоемкость дисциплины: 3 ЗЕ (108 академических часов)

Семестр: 4

Форма промежуточной аттестации: зачёт

Содержание дисциплины

Алгоритмы и данные. Свойства алгоритма. Анализ сложности алгоритма. Семантика, синтаксис, прагматика. Структура данных. Структуры хранения данных: вектор, список, сеть. Массивы. Структуры данных массивов. Структуры хранения массивов. Строки. Операции над строками. Записи. Операции над записями. Множества.

Списки. Структура и классификация списков. Операции над линейными списками. Применение списков. Стеки. Структура стека. Операции над стеками. Применение стеков. Очереди. Деки. Операции над очередями и деками. Применение очередей и деков.

Алгоритм преобразования m -арного дерева в бинарное дерево. Представление деревьев в памяти ЭВМ. Идеально-сбалансированное бинарное дерево. Бинарные (двоичные) деревья поиска. Операции над деревьями. В-деревья. Операции над В-деревьями.

Основные понятия и классификация алгоритмов сортировки. Внутренняя сортировка.

Классификация алгоритмов поиска. Поиск в последовательно организованных структурах. Последовательный поиск. Двоичный Поиск в деревьях.

Представление графов. Кратчайшие пути. Алгоритм Дейкстры. Алгоритм Флойда. Остовные деревья графа.