



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГЕОДЕЗИИ И КАРТОГРАФИИ
(МИИГАиК)

Факультет геоинформатики и информационной безопасности
Кафедра информатики и геоинформационных технологий



**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5387FD2AF5B17FED816FDE9A1D4973B1
Владелец: Степанченко Алексей Леонидович
Действителен: с 21.07.2025 по 14.10.2026

«УТВЕРЖДАЮ»

Проректор
А. Л. Степанченко
16 февраля 2024 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
ОСНОВЫ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
«Б1.О.06.01 АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»**

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.03 Прикладная информатика

Направленность(профиль) подготовки: Прикладная информатика

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2024

Срок получения образования: 4 года

Объем:
в зачетных единицах: 4 з.е.
в академических часах: 144 ак.ч.

Москва, 2024

Разработчики:

Преподаватель кафедры Информатики и геоинформационных технологий Лебедев Е. Д.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика, утвержденного приказом Минобрнауки от 19.09.2017 № 922

Согласование и утверждение

№	Подразделение	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1	факультет Геоинформатики и информационной безопасности	Председатель ученого совета факультета	Матерухин А. В.	Рассмотрено	14.12.2023, № 7-23

Содержание

1. Цель и задачи освоения дисциплины (модуля)
2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы
3. Место дисциплины в структуре ОП
4. Объем дисциплины и виды учебной работы
5. Содержание дисциплины
 - 5.1. Разделы, темы дисциплины и виды занятий
 - 5.2. Содержание разделов, тем дисциплины и формы текущего контроля
6. Оценочные материалы текущего контроля
7. Оценочные материалы промежуточной аттестации
8. Порядок проведения промежуточной аттестации
9. Материально-техническое и учебно-методическое обеспечение дисциплины
 - 9.1. Перечень основной и дополнительной учебной литературы
 - 9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся
 - 9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине
 - 9.4. Оборудование и технические средства обучения
10. Методические указания по освоению дисциплины (модуля)

1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - является формирование у обучающихся знаний о принципах построения и анализа алгоритмов, а также развитие практических умений их реализации на языках программирования высокого уровня. . Освоение курса направлено на развитие способности выбирать и реализовывать алгоритмические методы решения прикладных задач информатики с учётом требований к производительности и надёжности программных решений.

Задачи изучения дисциплины:

- Освоение базовых принципов построения алгоритмов и написания кода;
- Разработка действующих программных решений на C++ и Python в соответствии с заданными алгоритмами;
- Изучение принципов модульного программирования: разделение кода на функции и их интеграция в единую структуру;
- Освоение основных алгоритмических стратегий и принципов построения эффективных решений..

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

Компетенции, индикаторы и результаты обучения

ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения

ОПК-7.1 Анализирует поставленную задачу и требования к программному продукту, определяет необходимые алгоритмические и программные решения, а также оценивает их применимость для практического использования

Знать:

ОПК-7.1/Зн5 Принципы формализации вычислительных задач, понятие входных и выходных данных, критерии корректности и эффективности алгоритма, а также основные подходы к постановке и декомпозиции задач в программировании.

Уметь:

ОПК-7.1/Ум5 Проводить анализ исходных данных и условий задачи, формулировать требования к алгоритму и программной реализации, определять ограничения и оценивать применимость выбранного решения.

Владеть:

ОПК-7.1/Нв5 Навыками анализа поставленной задачи, включающим формализацию исходных данных в виде алгоритмической модели задачи, выбор рациональных методов решения и обоснование предпочтительной алгоритмической стратегии.

ОПК-7.2 Разрабатывает эффективные алгоритмы для решения задач, а также выбирает оптимальные структуры данных для их реализации

Знать:

ОПК-7.2/Зн5 Классы алгоритмических стратегий; стандартные структуры данных (массив, список, дерево, граф, куча и др.); понятия сложности алгоритма.

Уметь:

ОПК-7.2/Ум5 Конструировать алгоритмы с приемлемой асимптотикой, выбирать структуры данных под конкретную задачу.

Владеть:

ОПК-7.2/Нв5 Навыками разработки, реализации и оптимизации алгоритмов с использованием подходящих структур данных

ОПК-7.3 Пишет программный код на одном или нескольких языках программирования, следуя принципам модульности, читаемости, документирования и соответствия стандартам кодирования

Знать:

ОПК-7.3/Зн5 Синтаксис и основные конструкции языков C++ и Python, включая типы данных, операторы, функции, циклы, условные выражения и механизмы обработки исключений, а также принципы структурного и модульного программирования, обеспечивающие читаемость, повторное использование и сопровождение программного кода.

Уметь:

ОПК-7.3/Ум5 Реализовывать алгоритмы в виде программных модулей, документировать код и соблюдать стандарты оформления

Владеть:

ОПК-7.3/Нв5 Навыками программирования и реализации алгоритмов на языках C++ и Python с соблюдением принципов читаемости и модульности

ОПК-7.4 Проводит отладку и тестирование программного кода, выявляет и устраняет ошибки, а также оптимизирует производительность программы для обеспечения её стабильной и эффективной работы в реальных условиях

Знать:

ОПК-7.4/Зн5 Методы и стратегии тестирования (юнит-тесты, граничные случаи); приёмы отладки и оптимизации; основные метрики производительности.

Уметь:

ОПК-7.4/Ум5 Проводить тестирование, находить ошибки применять оптимизацию узких мест.

Владеть:

ОПК-7.4/Нв5 Навыками отладки и оптимизации программного кода, включающим умение использовать инструменты анализа выполнения программы, выявлять и устранять ошибки, а также применять методы оптимизации вычислений и управления памятью для повышения производительности реализованных алгоритмов.

3. Место дисциплины в структуре ОП

Дисциплина (модуль) Б1.О.06.01 «Алгоритмизация и программирование» относится к обязательной части образовательной программы и изучается в семестре(ах): 2.

В процессе изучения дисциплины студент готовится к решению типов задач профессиональной деятельности, предусмотренных ФГОС ВО и образовательной программой.

4. Объем дисциплины (модуля) и виды учебной работы

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Лекционные занятия (часы)	Практические занятия (часы)	Самостоятельная работа (часы)	Промежуточная аттестация (часы)

2 семестр	144	4	36	36	72	Зачет с оценкой
Всего	144	4	36	36	72	

5. Содержание дисциплины (модуля)

5.1. Разделы, темы дисциплины и виды занятий

(часы промежуточной аттестации не указываются)

Наименование раздела, темы	Всего	Лекционные занятия	Практические занятия	Самостоятельная работа	Планируемые результаты обучения, соотношенные с результатами освоения программы
Раздел 1. Понятие алгоритма. Методы оценки алгоритмов.	20	4	4	12	ОПК-7.1 ОПК-7.3
Тема 1.1. Введение в асимптотический анализ	10	2	2	6	
Тема 1.2. Анализ рекурсивных алгоритмов	10	2	2	6	
Раздел 2. Методы оптимизации алгоритмов	62	16	16	30	ОПК-7.1 ОПК-7.2 ОПК-7.3
Тема 2.1. Метод декомпозиции	14	4	4	6	
Тема 2.2. Динамическое программирование	14	4	4	6	
Тема 2.3. Жадные алгоритмы	10	2	2	6	
Тема 2.4. Поиск с возвратом	10	2	2	6	
Тема 2.5. Эвристические алгоритмы	14	4	4	6	
Раздел 3. Введение в параллельные алгоритмы	28	8	8	12	ОПК-7.1 ОПК-7.2 ОПК-7.3
Тема 3.1. Векторные инструкции процессора	14	4	4	6	
Тема 3.2. Параллелизация на уровне данных	14	4	4	6	
Раздел 4. Приложение алгоритмов на примере графовых задач	34	8	8	18	ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4
Тема 4.1. Деревья. Алгоритмы индексирования данных.	10	2	2	6	
Тема 4.2. Задача поиска кратчайшего пути между двумя точками	10	2	2	6	
Тема 4.3. Задача коммивояжера	14	4	4	6	
Итого	144	36	36	72	

5.2. Содержание разделов, тем дисциплин и формы текущего контроля

Раздел 1. Понятие алгоритма. Методы оценки алгоритмов.

Тема 1.1. Введение в асимптотический анализ

Асимптотический анализ — это метод исследования поведения функций или систем в предельных случаях, например, при бесконечном увеличении входных данных или приближении к бесконечно удаленной точке. Он позволяет аппроксимировать сложные функции, определять временную сложность алгоритмов с помощью асимптотических нотаций (например, O , Ω , Θ), а также анализировать предельные свойства случайных величин и поведение физических систем.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 1.2. Анализ рекурсивных алгоритмов

Анализ рекурсивных алгоритмов включает оценку их временной и пространственной сложности, что является сложной задачей из-за рекурсивного характера алгоритмов и использования стека вызовов. Для анализа используются методы, такие как метод подстановки, рекуррентные соотношения (или рекуррентные уравнения), а также основная теорема (для разделяй и властвуй), которая позволяет оценивать сложность алгоритмов, разбивающих задачу на подзадачи. Визуализация рекурсивного дерева помогает понять структуру и количество вызовов, а анализ стека вызовов позволяет оценить потребление памяти.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 2. Методы оптимизации алгоритмов

Тема 2.1. Метод декомпозиции

Метод декомпозиции в алгоритмах — это стратегия "разделяй и властвуй", где сложная задача разбивается на более мелкие, управляемые и идентичные подзадачи, которые решаются рекурсивно, а затем их решения объединяются для получения общего результата. Этот метод упрощает разработку, облегчает отладку и улучшает организацию кода, а также является основой для создания эффективных алгоритмов в информатике.

Процесс декомпозиции алгоритмов

1. Разбивка задачи на меньшие. Основная сложная задача делится на меньшие подзадачи, которые должны быть более простыми для решения.
2. Каждый из меньших экземпляров задачи решается, часто с помощью рекурсивного вызова того же алгоритма, пока задачи не станут настолько малы, чтобы их можно было решить легко.
3. Решения отдельных подзадач объединяются для получения окончательного решения исходной задачи.

Ряд эффективных алгоритмов построены на основе метода декомпозиции, среди них:

- а) Сортировка слиянием (Merge Sort): Разбивает массив на подмассивы, сортирует их, а затем сливает отсортированные подмассивы.
- б) Быстрая сортировка (Quick Sort): Выбирает опорный элемент и разбивает массив на две части, после чего рекурсивно сортирует эти части.

Текущий контроль
Вид (форма) контроля
Тестирование

Тема 2.2. Динамическое программирование

Динамическое программирование — это метод решения задач путём разбиения их на более простые, перекрывающиеся подзадачи, решения которых сохраняются и используются повторно для избежания лишних вычислений.

Текущий контроль
Вид (форма) контроля
Тестирование

Тема 2.3. Жадные алгоритмы

Жадные алгоритмы — это алгоритмы оптимизации, которые на каждом шаге делают локально оптимальный выбор, чтобы получить в итоге глобально оптимальное решение. Они отличаются простотой реализации и скоростью работы. Однако, жадные алгоритмы не всегда гарантируют оптимальность и работают только для задач с определенной структурой.

Текущий контроль
Вид (форма) контроля
Тестирование

Тема 2.4. Поиск с возвратом

Алгоритм backtracking (возврат к исходным данным) — это метод решения задачи перебора всех возможных вариантов с последующим выбором оптимального решения. Данный подход широко применяется в области разработки программного обеспечения, особенно в решении задач комбинаторной оптимизации, где требуется найти наилучшее решение из множества возможных комбинаций. Принцип работы алгоритма backtracking основывается на рекурсивном итеративном процессе, который проходит по всем возможным ветвям решения задачи. Основной идеей является перебор всех возможных решений путем последовательного выбора вариантов и проверки их на соответствие заданным условиям.

Текущий контроль
Вид (форма) контроля
Тестирование

Тема 2.5. Эвристические алгоритмы

Эвристический алгоритм — алгоритм решения задачи, включающий практический метод, не являющийся гарантированно точным или оптимальным, но достаточный для решения поставленной задачи. Позволяет ускорить решение задачи в тех случаях, когда точное решение не может быть найдено.

Текущий контроль
Вид (форма) контроля
Тестирование

Раздел 3. Введение в параллельные алгоритмы

Тема 3.1. Векторные инструкции процессора

Векторные инструкции процессора – это команды, позволяющие выполнять одну и ту же операцию одновременно над несколькими элементами данных (вектором), что значительно ускоряет вычисления по сравнению со скалярными инструкциями, обрабатывающими по одному элементу за раз. Эта технология реализуется в виде векторных расширений, таких как AVX, в современных процессорах архитектур x86 и ARM.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 3.2. Параллелизация на уровне данных

Параллелизация на уровне данных — это выполнение одной и той же операции параллельно над различными элементами данных, например, над частями одного массива или коллекции. Этот подход разбивает большие наборы данных на меньшие части, каждая из которых обрабатывается одновременно на отдельных вычислительных узлах или ядрах процессора, что значительно ускоряет выполнение задач, таких как численные моделирования или архивация.

Текущий контроль

Вид (форма) контроля
Тестирование

Раздел 4. Приложение алгоритмов на примере графовых задач

Тема 4.1. Деревья. Алгоритмы индексирования данных.

Деревья являются в некотором смысле простейшим классом графов. Для них выполняются многие интересные утверждения, которые не всегда выполняются для графов в общем случае. Применительно к деревьям многие доказательства и рассуждения оказываются намного проще. Выдвигая какие-то гипотезы при решении задач теории графов, целесообразно сначала их проверять на деревьях. Деревья являются самым распространённым классом графов, применяемых в программировании, причём в самых разных ситуациях.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 4.2. Задача поиска кратчайшего пути между двумя точками

Задача поиска кратчайшего пути между двумя точками — это задача нахождения пути с минимальной суммой весов (например, расстояния или времени) между двумя вершинами на графе, представленном как узлы и связи. Классическим решением для графов с неотрицательными весами является алгоритм Дейкстры, а для графов, где ищут кратчайший путь между всеми парами вершин, существует алгоритм Флойда-Уоршелла.

Текущий контроль

Вид (форма) контроля
Тестирование

Тема 4.3. Задача коммивояжера

Задача коммивояжёра — найти самый короткий маршрут, проходящий через все заданные города ровно по одному разу, с возвращением в исходный город. Эта задача относится к комбинаторной оптимизации, где необходимо посетить N городов, чтобы минимизировать суммарное расстояние или стоимость поездки, что формально сводится к поиску гамильтонова цикла минимальной длины в графе. Для решения задачи коммивояжера используются точные методы (полный перебор, метод ветвей и границ) для небольшого числа городов, и эвристические алгоритмы (ближайшего соседа, жадный алгоритм) для более крупных задач, чтобы найти приближенное, но быстрое решение.

Текущий контроль

Вид (форма) контроля
Тестирование

6. Порядок проведения промежуточной аттестации

Порядок проведения промежуточной аттестации регламентируется локальными нормативными актами Университета:

- Положением о формах, периодичности, порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по основным образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.6 «Организация и проведение промежуточной аттестации».
- Положением о порядке осуществления образовательного процесса, проведения текущего контроля успеваемости и промежуточной аттестации обучающихся с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.8 «Проведение промежуточной аттестации с применением ЭО и ДОТ».

Промежуточная аттестация: Зачет с оценкой, 2 семестр.

7. Оценочные материалы текущего контроля

Раздел 1. Понятие алгоритма. Методы оценки алгоритмов.

Контролируемые ИДК: ОПК-7.1 ОПК-7.3

Тема 1.1. Введение в асимптотический анализ

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Впишите пропущенное слово: Согласно тезису Черча, каждая вычислимая функция ... рекурсивна.

Впишите пропущенное слово: Согласно тезису Черча, каждая вычислимая функция ... рекурсивна.

2. Укажите основные ресурсы, используемые для оценки работы алгоритма

1. Машинное время
2. Оплата труда программистов
3. Оперативная память
4. Долговременная память
5. Пропускная способность сети
6. Стоимость лицензии

Тема 1.2. Анализ рекурсивных алгоритмов

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Соотнесите этапы рекурсии с их описанием

Этап Описание

- 1 Прямой ход А Критическая часть, предотвращающая бесконечность
- 2 Обратный ход Б Углубление вызовов до базового случая;
- 3 Условие выхода В Возврат значений по цепочке

2. Выберите все утверждения, которые верно характеризуют дерево рекурсии

1. Отображает систему рекурсивных вызовов функции
2. Используется только для итеративных алгоритмов
3. Не зависит от глубины рекурсии
4. Каждый узел дерева соответствует одному вызову функции
5. листья дерева представляют базовые случаи
6. Не отражает количество операций

Раздел 2. Методы оптимизации алгоритмов

Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3

Тема 2.1. Метод декомпозиции

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какие из перечисленных алгоритмов реализуют принцип метода «разделяй и властвуй»?

1. Быстрая сортировка (QuickSort)
2. Сортировка слиянием (MergeSort)
3. Линейный поиск
4. Сортировка вставками
5. Алгоритм бинарного поиска

2. Вставьте в нужном порядке слова: В алгоритме Карацубы умножение сводится к ... умножениям меньшего размера, с последующим ... результатов.

- 1) Трём;
- 2) Двум;
- 3) Объединением;
- 4) Разделением;

Тема 2.2. Динамическое программирование

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какие из перечисленных преимуществ характерны для динамического программирования?

- A. Уменьшение временной сложности алгоритма за счёт исключения повторных вычислений.
- B. Компактность кода: требует меньше строк кода, чем рекурсия.
- C. Возможность решать задачи, которые невозможно разбить на подзадачи.
- D. Работа с одним типом данных: работа возможна только с массивами
- E. Эффективное использование памяти для хранения промежуточных результатов.

2. Алгоритм вычисления значения функции $F(n)$, где n - натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(n) = F(n-1) * (n + 1), \text{ при } n > 1$$

Чему равно значение функции $F(5)$?

Алгоритм вычисления значения функции $F(n)$, где n - натуральное число, задан следующими соотношениями:

$$F(1) = 1$$

$$F(n) = F(n-1) \cdot (n + 1), \text{ при } n > 1$$

Чему равно значение функции $F(5)$?

Тема 2.3. Жадные алгоритмы

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какие преимущества характерны для жадных алгоритмов? Выберите один или несколько верных ответов.

- A. Простота реализации
- B. Гарантированная оптимальность решения
- C. Высокая скорость работы на больших объёмах данных
- D. Обязательное использование рекурсии
- E. Подходит для задач с независимыми шагами

2. Дополните предложение, ответы напишите через запятую: «Жадные алгоритмы основаны на последовательном выборе _____, которое кажется _____ на каждом шаге».

- базового случая;
- локального решения;
- рекурсивным;
- точным;
- наилучшим;

Тема 2.4. Поиск с возвратом

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Дополните предложение, записав правильный ответ через запятую: «Поиск с возвратом эффективен при наличии _____, которые позволяют _____ неподходящие ветви дерева решений».

- границ;
- отсекаль;
- повторять;
- ограничений;
- результатов;
- объединять;

2. Выберите все задачи, в которых целесообразно применять метод поиска с возвратом.

- A. Расстановка восьми ферзей на шахматной доске так, чтобы ни один из них не находился под боем другого
- B. Поиск кратчайшего пути в графе с неотрицательными весами
- C. Подбор комбинаций элементов, сумма которых равна заданному числу
- D. Решение sudoku
- E. Сортировка массива методом пузырька

Тема 2.5. Эвристические алгоритмы

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какие характеристики отличают эвристические алгоритмы?

- A. Всегда находят точное решение
- B. Не зависят от параметров задачи
- C. Основаны на приближённых решения
- D. Используют интуитивные правила

Е. Не гарантируют оптимальности результата

2. Соотнесите тип эвристического алгоритма и его краткую характеристику.

Тип алгоритма Характеристика

1. Генетический алгоритм

2. Поиск с имитацией отжига

3. Муравьиный алгоритм

А. Использует вероятностный выбор для выхода из локальных минимумов

С. Использует коллективное поведение агентов для нахождения пути

В. Имитация эволюционных процессов — скрещивания и мутаций

Раздел 3. Введение в параллельные алгоритмы

Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3

Тема 3.1. Векторные инструкции процессора

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Дополните предложение, ответы напишите через запятую: «. .. инструкции, такие как SSE и AVX, обрабатывают несколько элементов данных ... ».

- Скалярные;
- Векторные;
- Последовательно;
- Одновременно;

2. Выберите утверждения, справедливые для интрисиков.

А. Интрисики позволяют использовать SIMD-инструкции на уровне языка C/C++.

В. Интрисики автоматически оптимизируются компилятором без участия программиста.

С. Использование интрисиков может повысить производительность при параллельной обработке данных.

Д. Интрисики являются макросами, не зависящими от архитектуры процессора

Тема 3.2. Параллелизация на уровне данных

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Соотнесите фреймворки с описанием:

Фреймворк Описание

1. OpenMP А. Реализует GPU-вычисления

2. PyCUDA В. Обеспечивает параллелизацию на уровне CPU с помощью директив в коде.

3. Numba С. Ускоряет вычисления с помощью JIT-компиляции.

2. Выберите утверждения, справедливые для мьютексов.

А. Обеспечивают взаимное исключение при доступе к разделяемым данным.

В. Позволяют нескольким потокам одновременно владеть одним ресурсом.

С. Поток, заблокировавший мьютекс, должен сам его разблокировать.

Д. Используются исключительно для синхронизации потоков в разных процессах.

Е. Мьютексы предотвращают состояния гонки (race conditions).

Раздел 4. Приложение алгоритмов на примере графовых задач

Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4

Тема 4.1. Деревья. Алгоритмы индексирования данных.

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какие утверждения справедливы для бинарного дерева поиска (BST)?
- A. Левое поддереву узла содержит элементы, меньшие ключа узла.
 - B. Правое поддереву узла содержит элементы, большие ключа узла.
 - C. Элементы в поддеревьях могут располагаться в произвольном порядке.
 - D. Бинарное дерево поиска обеспечивает логарифмическое время поиска при сбалансированной структуре.
 - E. В BST невозможно удаление узлов.

2. Сколько узлов в полном бинарном дереве, состоящем из 10 уровней?
Сколько узлов в полном бинарном дереве, состоящем из 10 уровней?

Тема 4.2. Задача поиска кратчайшего пути между двумя точками

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Какие алгоритмы решают задачу поиска кратчайшего пути?
- A. Алгоритм Дейкстры
 - B. Алгоритм Беллмана–Форда
 - C. Алгоритм Кнута–Морриса–Пратта
 - D. Алгоритм A*
 - E. Алгоритм Дельта-шага
 - F. Алгоритм Краскала
2. Какие способы представления графа используются в компьютерных структурах данных?
- A. Матрица смежности
 - B. Список смежности
 - C. Матрица инцидентности
 - D. Таблица достижимости
 - E. Массив ключей

Тема 4.3. Задача коммивояжера

Форма контроля/оценочное средство: Тестирование

Вопросы/Задания:

1. Соотнесите тип метода решения задачи коммивояжера с его характеристикой
- | Тип метода | Характеристика |
|--------------------------|---|
| 1. Полный перебор | A. Использует вероятностный выбор маршрутов и мутации для поиска оптимума |
| 2. Метод ветвей и границ | B. Последовательно исключает заведомо невыгодные маршруты, сокращая пространство поиска |
| 3. Жадный алгоритм | C. На каждом шаге выбирает ближайший доступный город |
| 4. Генетический алгоритм | D. Гарантированно находит оптимальный путь, но требует экспоненциального времени |
2. Дополните предложение, ответы напишите через запятую: «В задаче коммивояжера требуется ... все города ровно один раз и ... длину маршрута».
- Минимизировать
 - Посетить
 - Увеличить
 - Исключить

8. Оценочные материалы промежуточной аттестации

2 семестр, Зачет с оценкой

Контролируемые ИДК: ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-7.4

Вопросы/Задания:

1. Раскройте проблему формализации понятия алгоритма. Опишите основные модели формализации: машину Тьюринга, нормальные алгоритмы Маркова и λ -исчисление Черча.
2. Сформулируйте тезис Черча и объясните его значение. Дайте определение алгоритмически неразрешимым задачам. Сформулируйте теорему о рекурсии.
3. Охарактеризуйте вычислительные ресурсы (время и память). Опишите метод асимптотического анализа алгоритмов с использованием нотации «big-O» (O-большое).
4. Опишите способы представления чисел (целых и вещественных) в памяти компьютера.
5. Объясните проблемы компьютерных вычислений, связанные с использованием стандарта IEEE754 (потеря точности, машинный эпсилон, некорректные сравнения).
6. Опишите назначение и приведите примеры применения побитовых операторов.
7. Опишите уровни параллелизации: на уровне процессора (конвейеризация, суперскалярность), векторные операции (SIMD) и понятие интринсиков (интринзик-функций).
8. Дайте определения понятиям «процесс» и «поток». Опишите модель параллелизации на уровне данных.
9. Сформулируйте проблему разделения ресурсов (race condition) на примере задачи об обедающих философах.
10. Перечислите и охарактеризуйте основные средства синхронизации: атомарные переменные, мьютексы, семафоры.
11. Охарактеризуйте класс алгоритмов численных приближений.
12. Опишите стратегию «разделяй и властвуй» как метод оптимизации алгоритмов.
13. Проиллюстрируйте стратегию "разделяй и властвуй" на примере бинарного возведения в степень.
14. Опишите подходы к оценке эффективности параллельных вычислений. Введите модель вычислений в виде графа «операции-операнды».
15. Перечислите основные показатели эффективности параллельного алгоритма (ускорение, эффективность, масштабируемость).
16. Сформулируйте закон Амдала и закон Густавсона-Барсиса, объясните их практическое применение и ограничения.
17. Опишите особенности реализации параллельных алгоритмов в Python. Объясните назначение и механизм работы Global Interpreter Lock (GIL), его влияние на многопоточность.

18. Перечислите готовые решения и библиотеки для создания многопоточных и многопроцессорных приложений в Python (threading, multiprocessing, asyncio, concurrent.futures).
19. Проведите сравнительный асимптотический анализ алгоритмов сортировки: сортировки пузырьком, поразрядной сортировки и быстрой сортировки.
20. Охарактеризуйте метод полного перебора (brute force) и укажите его недостатки.
21. Опишите методы оптимизации перебора: алгоритмы поиска с возвратом (backtracking) и метод ветвей и границ.
22. Дайте определение графа. Приведите классификацию видов графов (ориентированные/неориентированные, взвешенные/невзвешенные и др.).
23. Опишите способы представления графов в памяти: матрица смежности, списки смежности, а также специализированные форматы хранения (например, CSR — Compressed Sparse Row).
24. Охарактеризуйте граф как инструмент научного анализа и приведите примеры графовых задач.
25. Опишите алгоритмы обхода графов: поиск в глубину (DFS) и поиск в ширину (BFS).
26. Дайте определение дерева как структуры данных. Опишите способы обхода деревьев (в глубину, в ширину).
27. Охарактеризуйте структуру «двоичная куча» (binary heap) и ее свойства.
28. Раскройте суть метода динамического программирования.
29. Охарактеризуйте класс жадных алгоритмов. Раскройте суть задачи о рюкзаке и продемонстрируйте применение жадного подхода к ее решению.
30. Охарактеризуйте класс эвристических алгоритмов.
31. Опишите принцип работы, основные этапы и примеры использования следующего эвристического алгоритма: Алгоритм имитации отжига.
32. Опишите принцип работы, основные этапы и примеры использования следующего эвристического алгоритма: Генетический алгоритм.
33. Опишите принцип работы, основные этапы и примеры использования следующего эвристического алгоритма: Муравьиный алгоритм.

9. Материально-техническое и учебно-методическое обеспечение дисциплины

9.1. Перечень основной и дополнительной учебной литературы

Основная литература

1. Страуструп Б. Язык программирования C++ для профессионалов: учебник: учебник / Б. Страуструп. - Язык программирования C++ для профессионалов - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. - 670 с. - 978-5-4497-0922-6. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/146412.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

2. Тарасов В. Н. Численные методы. Теория, алгоритмы, программы: учебное пособие: учебное пособие / В. Н. Тарасов, Н. Ф. Бахарева. - Численные методы. Теория, алгоритмы, программы - Самара: Поволжский государственный университет телекоммуникаций и информатики, 2017. - 266 с. - 5-7410-0451-2. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/71903.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

3. Ягьяева Л. Т. Теория алгоритмов и программ: учебное пособие: учебное пособие / Л. Т. Ягьяева, М. Ю. Валеев. - Теория алгоритмов и программ - Казань: Казанский национальный исследовательский технологический университет, 2019. - 116 с. - 978-5-7882-2737-5. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/109603.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

4. Алексеев А. А. Основы параллельного программирования с использованием Visual Studio 2010: учебное пособие: учебное пособие / А. А. Алексеев. - Основы параллельного программирования с использованием Visual Studio 2010 - Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. - 330 с. - 978-5-4497-0341-5. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/89456.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

5. Шелудько В. М. Основы программирования на языке высокого уровня Python: учебное пособие: учебное пособие / В. М. Шелудько. - Основы программирования на языке высокого уровня Python - Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. - 146 с. - 978-5-9275-2649-9. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/87461.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

Дополнительная литература

1. Павлов Л. А. Структуры и алгоритмы обработки данных: учебник для вузов: учебник для вузов / Л. А. Павлов, Н. В. Первова. - 3-е изд., стер. - Санкт-Петербург: Лань, 2021. - 256 с. - 978-5-8114-7259-8. - Текст: электронный. // Издательство Лань: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/156929.jpg> (дата обращения: 16.03.2025). - Режим доступа: по подписке

2. Костюкова Н. И. Комбинаторные алгоритмы для программистов: учебное пособие: учебное пособие / Н. И. Костюкова. - Комбинаторные алгоритмы для программистов - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 216 с. - 978-5-4497-2406-9. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/133945.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

3. Моисеенкова Т. В. Дискретная математика в примерах и задачах: учебное пособие: учебное пособие / Т. В. Моисеенкова. - Дискретная математика в примерах и задачах - Красноярск: Сибирский федеральный университет, 2018. - 132 с. - 978-5-7638-3967-8. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/100011.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

4. Борзунов С. В. Языки программирования. Python: решение сложных задач: учебное пособие для вузов: учебное пособие для вузов / С. В. Борзунов, С. Д. Кургалин. - Санкт-Петербург: Лань, 2023. - 192 с. - 978-5-507-45923-0. - Текст: электронный. // Издательство Лань: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/319394.jpg> (дата обращения: 16.03.2025). - Режим доступа: по подписке

5. Левин М. П. Параллельное программирование с использованием OpenMP: учебное пособие: учебное пособие / М. П. Левин. - Параллельное программирование с использованием OpenMP - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 133 с. - 978-5-4497-3330-6. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/142293.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

6. Иванов Б. Н. Дискретная математика. Алгоритмы и программы. Расширенный курс: учебное пособие для вузов: учебное пособие для вузов / Б. Н. Иванов. - 2-е изд., стер. - Санкт-Петербург: Лань, 2024. - 668 с. - 978-5-507-49205-3. - Текст: электронный. // Издательство Лань: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/382373.jpg> (дата обращения: 16.03.2025). - Режим доступа: по подписке

7. Барков И. А. Объектно-ориентированное программирование: учебник для вузов: учебник для вузов / И. А. Барков. - 2-е изд., стер. - Санкт-Петербург: Лань, 2023. - 700 с. - 978-5-507-47113-3. - Текст: электронный. // Издательство Лань: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/329549.jpg> (дата обращения: 16.03.2025). - Режим доступа: по подписке

8. Сузи Р. А. Язык программирования Python: учебное пособие: учебное пособие / Р. А. Сузи. - Язык программирования Python - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. - 350 с. - 978-5-4497-3351-1. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/142310.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

9. Абрамян М. Э. Введение в стандартную библиотеку шаблонов C++. Описание, примеры использования, учебные задачи: учебник: учебник / М. Э. Абрамян. - Введение в стандартную библиотеку шаблонов C++. Описание, примеры использования, учебные задачи - Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. - 178 с. - 978-5-9275-2374-0. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/87401.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

10. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: учебное пособие: учебное пособие / А. В. Боресков, А. А. Харламов, Н. Д. Марковский [и др.] - Параллельные вычисления на GPU. Архитектура и программная модель CUDA - Москва: Московский государственный университет имени М.В. Ломоносова, 2015. - 336 с. - 978-5-19-011058-6. - Текст: электронный. // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/54647.html> (дата обращения: 16.03.2025). - Режим доступа: по подписке

9.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся

Профессиональные базы данных

Не используются.

Ресурсы «Интернет»

1. <https://ratcatcher.ru/alg/> - Интерактивные конспекты по курсы "Алгоритмизация и программирование"
2. <https://recursion.vercel.app/> - Инструмент построения дерева рекурсии
3. <https://programforyou.ru/block-diagram-redactor> - Редактор блок-схем
4. <https://www.online-cpp.com/> - Онлайн-компилятор языка программирования C++

5. <https://lms.miiigaik.ru/> - СДО МИИГАиК
6. <https://programforyou.ru/graph-redactor> - Визуализация графа. Построение кратчайшего маршрута алгоритмом Дейкстры
7. <https://www.iprbookshop.ru/> - Электронно-библиотечная система IPR SMART
8. https://ratcatcher.ru/media/alg/practice/prac_2/Sim.html - Интерактивная иллюстрация метода Симпсона
9. https://ratcatcher.ru/media/alg/practice/prac_5/Массив.html - Интерактивный демонстратор работы алгоритмов сортировки
10. https://ratcatcher.ru/media/alg/practice/prac_9/A_звезда.html - Интерактивная визуализация Алгоритма A*

9.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

1. Microsoft Visual Studio;

Перечень информационно-справочных систем

(обновление выполняется еженедельно)

Не используется.

9.4. Оборудование и технические средства обучения

Учебная аудитория

Маркерная доска - 0 шт.

Моноблок - 16 шт.

Проектор - 1 шт.

10. Методические указания по освоению дисциплины (модуля)

С целью успешного освоения данного предмета студенту необходимо выполнять домашние работы по указанным преподавателям темам и представленным на занятиях материалам. Для более успешного выполнения работы студенту предлагается использование учебно-методических материалов, соответствующих указанной теме.

Самостоятельная работа студентов описывается и регулируется методическими рекомендациями по дисциплине.

Описание возможностей изучения дисциплины лицами с ОВЗ и инвалидами

При необходимости обучения студентов-инвалидов и лиц с ограниченными возможностями здоровья аудиторные занятия могут быть заменены или дополнены изучением полнотекстовых лекций, презентаций, видео- и аудиоматериалов в электронной информационно-образовательной среде (ЭИОС) университета. Индивидуальные задания подбираются в адаптированных к ограничениям здоровья формах (письменно или устно, в форме презентаций). Выбор методов обучения зависит от их доступности для инвалидов и лиц с ограниченными возможностями здоровья.

Рабочая программа дисциплины для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.