Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Т. В. Снытникова

ТЕОРИЯ
СЛОЖНОСТИ ВЫЧИСЛЕНИЙ

Учебно-методическое пособие по изучению дисциплины для студентов
магистратуры по направлению подготовки
09.04.01 Информатика и вычислительная техника

Калининград
Издательство ФГБОУ ВО «КГТУ»

2023

УДК 338.24

Рецензент:
кандидат педагогических наук, доцент кафедры прикладной информатики
ФГБОУ ВО «Калининградский государственный технический университет»
Заболотнова Елена Юрьевна

Снытникова, Т. В.

Теория сложности вычислений: учебно-методическое пособие по изучению дисциплины для студентов магистратуры по направлению подготовки 09.04.01 Информатика и вычислительная техника / Т. В. Снытникова. – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2023. – 26 с.

Учебно-методическое пособие является руководством по изучению дисциплины «Теория сложности вычислений» для студентов магистратуры по направлению подготовки 09.04.01 – «Информатика и вычислительная техника». Содержит характеристику дисциплины (цель и планируемые результаты изучения дисциплины, место дисциплины в структуре основной профессиональной образовательной программы), тематический план с описанием для каждой темы формы проведения занятия, вопросы для изучения, методические материалы к занятиям.

Учебно-методическое пособие рассмотрено и одобрено в качестве локального электронного методического материала кафедрой прикладной информатики 03 июля 2023 г., протокол № 13

Учебно-методическое пособие по изучению дисциплины рекомендовано к использованию в учебном процессе в качестве локального электронного методического материала методической комиссией ИЦТ от 05.07.2023 г., протокол № 8

УДК 338.24

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 4
1 ТЕМАТИЧЕСКИЙ ПЛАН 5
2 СОДЕРЖАНИЕ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ 6
3 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ 15
Тема 1. Неформальное определение алгоритма. Арифметические и интуитивно вычислимые функции. 15
Тема 2. Машины Тьюринга. Конечные языки. Регулярные языки. 15
Тема 3. Регистровые машины. Частично вычислимые функции. Кодирование алгоритмов. 16
Тема 4. Кодирование алгоритмов. 17
Тема 5. Алгоритмические задачи. Алгоритмы разрешения. Трудноразрешимые задачи 17
Тема 6. Сложность алгоритмов и массовых проблем. Асимптотические оценки сложности. 18
Тема 7. Класс сложности NP. Сводимость и NP-полные задачи 22
Тема 8. Некоторые NP-полные задачи на графах. 23
Тема 9. Анализ сложности рекурсивных алгоритмов. Точная оценка сложности алгоритмов. 23
Тема 10. Алгоритмы сортировки и их анализ. 23
4 ТРЕБОВАНИЯ К АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ 24
4.1. Текущая аттестация 24
4.2. Порядок применения рейтинговой системы 24
4.3. Условия получения положительной оценки 24
БИБЛИОГРАФИЧЕСКИЙ СПИСОК 25

ВВЕДЕНИЕ

Дисциплина «Теория сложности вычислений» является частью образовательной программы магистратуры по направлению подготовки 09.04.01 Информатика и вычислительная техника.

Целью изучения дисциплины является формирование у учащихся способности разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем, на основе анализа сложности алгоритмов программного обеспечения и проводить оптимизацию в соответствии с поставленными условиями.

В результате освоения дисциплины обучающийся должен:

знать:

- основные понятия теории сложности вычислений,
- определения и свойства математических объектов, используемых в этой области,
- формулировки утверждений, методы их доказательства, возможные сферы их приложений;

уметь:

- решать задачи теоретического и прикладного характера из различных разделов теории сложности вычислений,
- доказывать утверждения,
- строить модели объектов и понятий;

владеть:

- математическим аппаратом теории сложности вычислений,
- методами доказательства утверждений в этой области,
- навыками решения основных задач.

Дисциплина опирается на компетенции, полученные при изучении дисциплин «Программирование», «Операционные системы», «Высокоуровневые технологии программирования», «Программная инженерия», «Теория графов», «Дискретная математика».

Промежуточная аттестация по дисциплине проходит в форме зачета по итогам текущей аттестации по дисциплине.

1 ТЕМАТИЧЕСКИЙ ПЛАН

Таблица 1 – Трудоёмкость освоения дисциплины в третьем семестре по очной форме обучения

№ п/п	Раздел (модуль) дисциплины	Контактная работа с преподавателем					СРС	Подготовка и аттестация в период сессии
		ЛК	ЛР	ПР	РЭ	КА		
1	Неформальное определение алгоритма. Арифметические и интуитивно вычислимые функции	2	2					
2	Машины Тьюринга. Конечные автоматы. Регулярные языки	4	4					
3	Регистровые машины. Частично вычислимые функции. Кодирование алгоритмов	4	4					
4	Алгоритмические задачи. Алгоритмы разрешения. Трудноразрешимые задачи	2	4					
5	Алгоритмически неразрешимые задачи. Универсальные функции	2	0					
6	Некоторые теоремы теории алгоритмов	2	0					
7	Сложность алгоритмов и массовых проблем. Асимптотические оценки сложности	2	2					
8	Классы сложности P и EXP	1	0					
9	Класс сложности NP. Сводимость и NP-полные задачи	3	2					
10	Теорема Кука-Левина: NP-полнота задачи выполнимости	2	0					
11	Некоторые NP-полные задачи на графах	2	4					
12	Классы PSPACE и DLOG	1	0					
13	Анализ сложности рекурсивных алгоритмов. Точная оценка сложности алгоритмов	1	4					
14	Алгоритмы сортировки и их анализ	2	4					
15	Алгоритмы вычислительной математики	2	0					
ИТОГО:		**30**	**30**		**2**	**0,15**	**45,85**	
Всего		**108**						

2 СОДЕРЖАНИЕ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

Ниже приведен тематический план лекционных и лабораторных занятий.

Тематический план лекционных занятий

Тема 1	Неформальное определение алгоритма. Арифметические и интуитивно вычислимые функции
Тема 2	Машины Тьюринга. Конечные автоматы. Регулярные языки
Тема 3	Регистровые машины. Частично вычислимые функции. Кодирование алгоритмов
Тема 4	Алгоритмические задачи. Алгоритмы разрешения. Трудноразрешимые задачи
Тема 5	Алгоритмически неразрешимые задачи. Универсальные функции
Тема 6	Некоторые теоремы теории алгоритмов
Тема 7	Сложность алгоритмов и массовых проблем. Асимптотические оценки сложности
Тема 8	Классы сложности P и EXP
Тема 9	Класс сложности NP. Сводимость и NP-полные задачи
Тема 10	Теорема Кука-Левина: NP-полнота задачи выполнимости
Тема 11	Некоторые NP-полные задачи на графах
Тема 12	Классы PSPACE и DLOG
Тема 13	Анализ сложности рекурсивных алгоритмов. Точная оценка сложности алгоритмов
Тема 14	Алгоритмы сортировки и их анализ
Тема 15	Алгоритмы вычислительной математики

Тематический план практических (лабораторных занятий)

Тема 1	Неформальное определение алгоритма. Арифметические и интуитивно вычислимые функции
Тема 2	Машины Тьюринга. Конечные автоматы. Регулярные языки
Тема 3	Регистровые машины. Частично вычислимые функции
Тема 4	Кодирование алгоритмов
Тема 5	Алгоритмические задачи. Алгоритмы разрешения. Трудноразрешимые задачи
Тема 6	Сложность алгоритмов и массовых проблем. Асимптотические оценки сложности
Тема 7	Класс сложности NP. Сводимость и NP-полные задачи
Тема 8	Некоторые NP-полные задачи на графах
Тема 9	Анализ сложности рекурсивных алгоритмов. Точная оценка сложности алгоритмов
Тема 10	Алгоритмы сортировки и их анализ

Тема 1. Неформальное определение алгоритма. Арифметические и интуитивно вычислимые функции

Ключевые вопросы темы

1 Определение массовой проблемы. Примеры.

2 Неформальное определение алгоритма и его характерные свойства.

3 Арифметические и вычислимые функции: определение, примеры.

Источники

[4, с. 5–6; 5, с. 3–7; 6, с. 3–11]

Методические рекомендации

Освоение данной темы предполагает знание и умение оперировать следующими понятиями: массовая проблема, алгоритм, конструктивное множество, арифметические и вычислимые функции.

Необходимо уметь доказывать существование неконструктивного множества и строить нумерацию конструктивных множеств.

Тема 2. Машины Тьюринга. Конечные автоматы. Регулярные языки

Ключевые вопросы темы

1 Модель машины Тьюринга.

2 Задание машины Тьюринга.

3 Алгоритм Тьюринга. Вычислимость по Тьюрингу.

4 Детерминированные и недетерминированные конечные автоматы. Эквивалентность ДКА и НКА.

5 Регулярные выражения и языки. Теорема Клини. Лемма о разрастании регулярных языков.

Источники

[1, с. 4–14; 3, с. 13–28; 4, с. 6–17]

Методические рекомендации

Освоение данной темы предполагает знания и умения оперировать следующими понятиями: модель машины Тьюринга с бесконечной лентой, модель машины Тьюринга с полубесконечной лентой, модель k-ленточной машины Тьюринга, язык, допускаемый машиной Тьюринга, модель детерминированных конечных автоматов, модель недетерминированных конечных автоматов, язык, допускаемый конечным автоматом, определение регулярных выражений и регулярных языков.

Необходимо уметь доказывать эквивалентность разных моделей машин Тьюринга, эквивалентность ДКА и НКА, уметь строить регулярное выражение по заданному конечному автомату, и недетерминированный конечный автомат по заданному регулярному выражению или языку.

Тема 3. Регистровые машины. Частично вычислимые функции. Кодирование алгоритмов

Ключевые вопросы темы

1 Модель машин Шёнфильда и Минского. Макросы.

2 Простейшие арифметические функции. Операции над функциями: суперпозиция, примитивная рекурсия, минимизация.

3 Классы функций: частично рекурсивные (вычислимые) функции, общерекурсивные функции, примитивно рекурсивные функции.

4 Тезис Чёрча.

5 Кодирование алгоритмов.

Источники

[1, с. 44–51, 23–38; 3, с. 82–121; 4, с. 121–128; 6, с. 68–70; 7, с. 12–19]

Методические рекомендации

Освоение данной темы предполагает знания и умения оперировать следующими понятиями: регистровые машины, простейшие арифметические функции, операторы суперпозиции, примитивной рекурсии и минимизации.

Необходимо знать следующих определений: вычислимая по Шёнфильду функция, частично рекурсивная функция, общерекурсивная функция, примитивная функция.

Необходимо уметь обосновывать эквивалентность различных определений вычислимости. Также уметь доказывать строгое вложение класса примитивно рекурсивных функций в класс общерекурсивных функций и класса общерекурсивных функций в класс частично рекурсивных функций.

Уметь работать с канторовской нумерацией n-ок неотрицательных чисел и геделевской нумерацией последовательностей.

Тема 4. Алгоритмические задачи. Алгоритмы разрешения. Трудноразрешимые задачи

Ключевые вопросы темы

1 Алгоритмические задачи: функциональные, логические, задачи принадлежности.

2 Алгоритмы разрешения.

Источники

[2, с. 10–16; 3, с. 10; 4, с. 22–24]

Методические рекомендации

Освоение данной темы предполагает знания следующих понятий: функциональные задачи, логические задачи, задачи принадлежности.

Требуется уметь формулировать массовые проблемы как функциональные, таки логические задачи и понимать, как связаны алгоритмы разрешения этих формулировок.

Тема 5. Алгоритмически неразрешимые задачи. Универсальные функции

Ключевые вопросы темы

1 Алгоритмически неразрешимые задачи. Определение. Примеры.

2 Универсальные функции.

Источники

[2, с. 6–17; 3, с. 121–148]

Методические рекомендации

Освоение данной темы предполагает знание и умение оперировать следующими понятиями: алгоритмически неразрешимая проблема, универсальная функция.

Необходимо уметь доказывать следующие утверждения: неразрешимость задачи останова машины Тьюринга, существование универсальной функции для множества ч.в.ф. от n аргументов и несуществование универсальной функции для множества о.р.ф. от n аргументов.

Тема 6. Некоторые теоремы теории алгоритмов

Ключевые вопросы темы

1 Доопределение функции.

2 Теорема о существовании доопределения частично рекурсивной функции.

3 Лемма о параметризации.

4 Теорема Клини о неподвижной точке.

Источники

[1, с. 21–24; 7, с. 21–24]

Методические рекомендации

Освоение данной темы предполагает знание следующих определений и формулировок: доопределение функции, рекурсивное множество, теоремы о доопределении, лемма о параметризации, теорема Клини о неподвижной точке, теорема Райса.

Необходимо уметь доказывать приведенные выше теоремы.

Тема 7. Сложность алгоритмов и массовых проблем. Асимптотические оценки сложности

Ключевые вопросы темы

1 Проблема сложности алгоритмов на примере алгоритмов поиска.

2 Асимптотические оценки сложности.

Источники

[2, с. 4–8; 6, с. 4–6; 7, с. 25–27]

Методические рекомендации

Освоение данной темы предполагает знание следующих понятий: асимптотически верхняя граница, асимптотически нижняя граница, асимптотически точная граница, свойства асимптотических скоростей роста.

Необходимо уметь находить асимптотические границы для следующих функций: полиномиальные, логарифмические, экспоненциальные.

Тема 8. Классы сложности P и EXP

Ключевые вопросы темы

1 Определение классов сложности.

2 Примеры задач класса P.

3 Примеры задач класса EXP.

Источники

[4, с. 25–26; 7, с. 27–28]

Методические рекомендации

Освоение данной темы предполагает знание и умение оперировать следующими понятиями: класс сложности, класс полиномиальной сложности, класс экспоненциальной сложности.

Необходимо уметь приводить примеры задач данных классов.

Тема 9. Класс сложности NP. Сводимость и NP-полные задачи

Ключевые вопросы темы

1 Класс сложности NP.

2 Понятие сводимости.

3 NP-полнота.

Источники

[1, с. 102–122; 4, с. 26–43; 7, с. 28–29]

Методические рекомендации

Освоение данной темы предполагает знание и умение оперировать следующими понятиями: недетерминированная машина Тьюринга, класс сложности NP, понятие полиномиальной сводимости задач.

Необходимо уметь приводить примеры задачи, сводимой к другой.

Тема 10. Теорема Кука-Левина: NP-полнота задачи выполнимости

Ключевые вопросы темы

1 Постановка проблемы выполнимости булевых функций.

2 Теорема Кука-Левина.

Источники

[1, с. 96–101; 2, с. 24–28]

Методические рекомендации

Освоение данной темы предполагает знание постановки проблемы выполнимости булевых функций.

Необходимо уметь формулировать и доказывать теорему Кука-Левина.

Тема 11. Некоторые NP-полные задачи на графах

Ключевые вопросы темы

1 Задача о вершинном покрытии.

2 Задача о гамильтоновом цикле.

3 Задача коммивояжера.

Источники

[1, с. 123–138; 4, с. 63–75]

Методические рекомендации

Освоение данной темы предполагает умение формулировать следующие задачи на графах: о вершинном покрытии, о нахождении гамильтонова цикла, задача коммивояжера.

Необходимо знать методы построения решения переборных задач: метод динамического программирования, метод “разделяй и властвуй”, перебор с возвратом, метод ветвей и границ.

Тема 12. Классы PSPACE и DLOG

Ключевые вопросы темы

1 Определение классов сложности по памяти (классы емкостной сложности).

2 Вложенность классов.

Источники

[6, с. 74–78]

Методические рекомендации

Освоение данной темы предполагает знание и умение оперировать следующими понятиями: емкостная сложность алгоритмов, класс PSPACE, класс DLOG.

Необходимо уметь вычислять емкостную сложность алгоритма, понимать связь между сложностью алгоритма по времени и по памяти.

Тема 13. Анализ сложности рекурсивных алгоритмов. Точная оценка сложности алгоритмов

Ключевые вопросы темы

1 Вычисление сложности рекурсивных алгоритмов.

2 Точная оценка сложности.

Источники

[7, с. 30–36]

Методические рекомендации

Освоение данной темы предполагает знание и умение оперировать следующими понятиями: рекурсивный алгоритм, глубина рекурсии, рекуррентное соотношение, рекурсия с мемоизацией.

Необходимо уметь вычислять точную оценку сложности алгоритма.

Тема 14. Алгоритмы сортировки и их анализ

Ключевые вопросы темы

1 Сортировка пузырьком.
2 Быстрая сортировка.
3 Сортировка подсчетом.
4 Сортировка вставками.
5 Сортировка слиянием.

Источники

[6, с. 7–10; 7, с. 37–41]

Методические рекомендации

Освоение данной темы предполагает знание различных алгоритмов сортировки и умение реализовывать эти алгоритмы на языке программирования.

Необходимо знать временные и емкостные характеристика алгоритмов сортировки.

Тема 15. Алгоритмы вычислительной математики

Ключевые вопросы темы

1 Алгоритм Карацубы для умножения чисел.
2 Алгоритм Тоома для умножения чисел.
3 Алгоритм Штрассена для умножения матриц.
4 Алгоритм Винограда для умножения матриц.

Источники

[6, с. 16–21; 7, с. 42–52]

Методические рекомендации

Освоение данной темы предполагает знание алгоритмов Карацубы и Тоома для умножения чисел и алгоритмы Штрассена и Винограда для умножения матриц.

Необходимо уметь вычислять оценку сложности этих алгоритмов.

3 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

Лабораторные работы проводятся с целью формирования у студентов знаний, умений и навыков, а также соответствующих компетенций.

Ниже представлен краткий план лабораторных работ с основными вопросами. Подготовку к лабораторным работам можно осуществлять с помощью рекомендованных литературных источников.

Тема 1. Неформальное определение алгоритма. Арифметические и интуитивно вычислимые функции

Для овладения понятийной базой темы предлагается для следующих массовых проблем доказать, что исходные и искомые множества являются конструктивными:

1) массовая проблема сортировки массива из n натуральных чисел;

2) по тройке натуральных чисел (a, b, c) требуется выдавать ответ «да», если уравнение $ax^2 + bx + c = 0$ имеет действительные корни, и ответ «нет» в противном случае;

3) по тексту программы, записанной на языке Python, определить, «зацикливается» она (ответ «да») или нет (ответ «нет»).

Поскольку исходные объекты массовой проблемы задаются конструктивными множествами, предлагается задать нумерацию для следующих множеств:

1) NxN;
2) все подмножества конечного множества.

Тема 2. Машины Тьюринга. Конечные языки. Регулярные языки

Построить следующие машины Тьюринга:

1. Дана десятичная запись натурального числа n>1. Построить МТ, вычисляющую функцию f(n)=n-1 так, чтобы запись n-1 не содержала левый ноль (не допускается запись 09). Начальное положение головки над крайним правым символом.
2. Дан массив из открывающих и закрывающих скобок. Построить МТ, которая удаляет пары взаимных скобок. Например, дано «][[][[]», на выходе «][[». Считывающая головка над крайним левым символом.

3. Дана строка из алфавита {a,b}. Построить МТ, которая переместит все буквы «a» в левую часть, а буквы «b» - в правую.
4. Построить МТ, которая преобразует запись числа из двоичной системы исчисления в восьмеричную.
5. На ленте МТ в трех клетках записаны три различных буквы А, В, С. Считывающая головка обозревает последнюю правую букву. Построить МТ, которая поменяет местами крайние буквы.
6. На ленте число в десятичной системе исчисления. Головка над правой цифрой. Записать цифры в обратном порядке.
7. Какой язык допускается конечным автоматом $M=(\{a,b\},\{q_0\},q_0,\{q_0\},\varnothing)$
8. Построить НКА, допускающий цепочки в алфавите {1,2}, у которых последний символ цепочки уже встречался в ней раньше. Построить эквивалентный ДКА.
9. Для следующих регулярных выражений построить НКА:
   1) b(ba+b)*+b;
   2) a(ab+b)*+ba;
   3) (ab+b)*(ba+ab);
   4) (a*b)*ab*a.
10. Найти регулярные выражения, обозначающие языки, все слова которых – элементы множества {0,1}*:
    1) оканчивающиеся на 011, 101, 110;
    2) начинающиеся с 110, 101 или 011;
    3) у которых каждый третий символ есть 0 или каждый второй – 1;
    4) содержащие каждую из подстрок 001 и 101;
    5) начинающиеся с 011 и содержащие вхождения подстроки 110;
    6) содержащее четное число символов 0 и нечетное число символов 1.

Тема 3. Регистровые машины. Частично вычислимые функции. Кодирование алгоритмов

Используя операции машины Минского a^0, a’, $a^-(n)$: (a≠0)? {a--, count++} : count=n написать макросы, вычисляющие следующие функции:

1 f(x,y)=x+y;

2 f(x,y)=min(x,y).

Для следующих арифметических функций выписать частично рекурсивную форму:

1 $c_m(x_1, x_2, ..., x_n) = m$;

2 $s_{m,k}(x_1, x_2, ..., x_n) = x_m + k$;

3 $sgn(x) = \{0, \wedge x = 0\ 1, \wedge x \neq 0$

4 $\neg sgn(x) = \{1, \wedge x = 0\ 0, \wedge x \neq 0$

5 sum(x,y)=x+y

6 mult(x,y)=x*y

7 power(x,y)= x^y

8 pred(x)=x-1;

9 $x \dot{-} y = \{0, \wedge x \leq y\ x - y, \wedge x > y$

10 |x-y|=$\{0, \wedge x = y\ x - y, \wedge x > y\ y - x, \wedge x < y$

11 f(x,y)=min(x,y)

12 f(x,y)=max(x,y)

Выписать аналитический вид для следующей функции

13 f(x,y)=μ z (|(y+z) –x|=0)

Тема 4. Кодирование алгоритмов

На любом языке программирования реализовать прямую и обратную нумерацию троек неотрицательных чисел. За основу взять на выбор одну из нумераций: Кантора или $\pi(x,y)=2^x(2y+1)-1$.

1) Вывести номера первых 6 пар, упорядоченных в лексикографическом порядке.
2) Восстановить координаты пар у полученных номеров.
3) Вычислить номер тройки (1,1,1).

Тема 5. Алгоритмические задачи. Алгоритмы разрешения. Трудноразрешимые задачи

Переформулировать функциональные задачи в логические.

1 УСЛОВИЕ. Задан граф G = <V, E>.

ВОПРОС. В какое минимальное число цветов можно раскрасить вершины графа так, чтобы любые две соседние вершины были окрашены в различные цвета?

2 УСЛОВИЕ. Задан граф G = <V, E>.

ВОПРОС. Найти клику максимальной мощности (подграф графа G такой, что любые две вершины в нём соединены ребром).

3 УСЛОВИЕ. Заданы граф G = <V, E>, целые положительные длины l(e) для всех ребер, выделенные вершины s и t – начало и конец.

ВОПРОС. Найти в G элементарный путь (все вершины различны) из s в t максимальной длины.

4 УСЛОВИЕ. Задано конечное множество A, положительные целые веса w(a), стоимости s(a) для каждого a ϵ A и общее ограничение на вес K.

ВОПРОС. Найти из всевозможных выборок A' ∈ A такую, чтобы суммарный вес входящих в него элементов не превосходил K, а суммарная стоимость была максимальна.

5 УСЛОВИЕ. Заданы конечное множество предметов A и размер s(a) ϵ (0, 1] для каждого предмета a ϵ A.

ВОПРОС. Найти такое разбиение множества A на непересекающиеся A1, A2, …, Ak, чтобы сумма размеров предметов в каждом Ai не превосходила 1 и k было минимальным.

6 УСЛОВИЕ. Заданы конечное множество X, целые положительные расстояния d(x, y) для каждой пары x, y элементов из X, целое положительное B.

ВОПРОС. Найти такое разбиение множества X на непересекающиеся X1, X2, …, Xk, что для всех x, y∈ Xi d(x, y) ≤ B и k было минимальным.

7 УСЛОВИЕ. Задано семейство C подмножеств конечного множества S.

ВОПРОС. Найти минимальное покрытие множества S. Иными словами, найти в C такое подмножество C', что любой элемент из S принадлежит, по крайней мере, одному подмножеству из C', и количество подмножеств в C' минимально.

8 УСЛОВИЕ. Задан граф G = <V, E>.

ВОПРОС. Найти в V минимальное доминирующее множество. Иными словами, найти подмножество V' ∈ V такое, что для любой вершины u∈V\V' существует такая вершина v∈ V', что u и v соединены ребром из E, и мощность |V'| минимальна.

Тема 6. Сложность алгоритмов и массовых проблем. Асимптотические оценки сложности

Для выполнения следующих заданий используйте правила асимптотического анализа:

- O(k*f) = O(f) – постоянный множитель k (константа) отбрасывается;
- O(f*g) = O(f)*O(g) – оценка сложности произведения двух функций равна произведению их сложностей;

- $O(f/g)=O(f)/O(g)$ – оценка сложности частного двух функций равна частному их сложностей;
- $O(f+g)$ равна доминанте $O(f)$ и $O(g)$ – оценка сложности суммы функций определяется как оценка сложности доминанты первого и второго слагаемых.

1. В приведенном ниже списке указано время выполнения пяти алгоритмов (предполагается, что указано точное время выполнения). Насколько медленнее будет работать каждый из алгоритмов, если: увеличить размер входных данных вдвое; увеличить размер входных данных на 1?

(a) n^2

(b) n^3

(c) $100n^2$

(d) $n \log n$

(e) 2^n

2. Есть шесть алгоритмов, время выполнения которых приведено ниже. (Предполагается, что указано точное количество выполняемых операций как функция размера данных n.) Предположим, имеется компьютер, способный выполнять 10^{10} операций в секунду, и результат должен быть вычислен не более чем за час.

При каком наибольшем размере входных данных n результат работы каждого алгоритма может быть вычислен за час?

(a) n^2

(b) n^3

(c) $100n^2$

(d) $n \log n$

(e) 2^n

3. Расположите функции из следующего списка по возрастанию скорости роста. Другими словами, если функция $g(n)$ в вашем списке следует непосредственно после $f(n)$, из этого следует, что $f(n)=O(g(n))$.

$f_1(n) = n^{2,5}$

$f_2(n) = \sqrt{2n}$

$f_3(n) = n + 10$

$f_4(n) = 10^n$

$f_5(n) = 100^n$

$f_6(n) = n^2 \log n$

4. Расположите функции из следующего списка по возрастанию скорости роста. Другими словами, если функция g(n) в вашем списке следует непосредственно после f (n), из этого следует, что f (n)=O(g(n)).

$g_1(n) = 2^{\sqrt{\log n}}$

$g_2(n) = 2^n$

$g_4(n) = n^{4/3}$

$g_3(n) = n(\log n)^3$

$g^5(n) = n^{\log n}$

$g_6(n) = 2^{(2^n)}$

$g_7(n) = 2^{(n^2)}$

5. Допустим, имеются функции f и g, такие что f(n)= O(g(n)). Для каждого из следующих утверждений укажите, является оно истинным или ложным, и приведите доказательство или контрпример.

(a) log2 f (n) = O(log2 g(n)).

(b) 2f (n) = O(2g(n)).

(c) $f(n)^2 = O(g(n)^2)$.

6. Рассмотрим следующую задачу: имеется массив A, состоящий из n целых чисел A[1], A[2], ..., A[n]. Требуется вывести двумерный массив B размером n × n, в котором B[i, j] (для i < j) содержит сумму элементов массива от A[i] до A[j], т. е. сумму A[i] + A[i + 1] + ... + A[j]. (Для i ≥ j значение элемента массива B[i, j] остается неопределенным, поэтому выводиться для таких элементов может что угодно.)

Ниже приведен простой алгоритм решения этой задачи.

```
For i = 1, 2, ..., n
    For j = i + 1, i + 2, ..., n
        Просуммировать элементы массива от A[i] до A[j]
        Сохранить результат в B[i, j]
    Конец For
Конец For
```

(a) Для некоторой выбранной вами функции f приведите границу времени выполнения этого алгоритма в форме O(f (n)) для входных данных размера n (то есть границу количества операций, выполняемых алгоритмом).

(b) Для той же функции f покажите, что время выполнения алгоритма для входных данных размера n также равно Ω(f (n)). (Тем самым определяется асимптотически точная граница Θ(f (n)) для времени выполнения.)

(c) Хотя алгоритм, проанализированный в частях (a) и (b), является наиболее естественным подходом к решению задачи (в конце концов, он просто перебирает нужные элементы массива B и заполняет их значениями), в нем присутствуют избыточные источники неэффективности. Предложите другой алгоритм для решения этой задачи с асимптотически лучшим временем выполнения. Другими словами, вы должны разработать алгоритм со временем выполнения O(g(n)), где $g\ \frac{(n)}{f}(n) = 0$.

7 Вы занимаетесь экстремальными испытаниями различных моделей стеклянных банок для определения высоты, при падении с которой они не разобьются. Тестовый стенд для этого эксперимента выглядит так: имеется лестница с n ступенями, и вы хотите найти самую высокую ступень, при падении с которой банка останется целой. Назовем ее наивысшей безопасной ступенью.

На первый взгляд естественно применить бинарный поиск: сбросить банку со средней ступени, посмотреть, разобьется ли она, а затем рекурсивно повторить попытку от ступени n/4 или 3n/4 в зависимости от результата. Но у такого решения есть недостаток: вероятно, при поиске ответа будет разбито слишком много банок.

С другой стороны, если вы стремитесь к экономии, можно выбрать следующую стратегию. Сначала банка сбрасывается с первой ступени, потом со второй и т. д. Высота каждый раз увеличивается на одну ступень, пока банка не разобьется. При таком подходе достаточно всего одной банки (как только она разобьется, вы получаете правильный ответ), но для получения ответа может потребоваться до n попыток (вместо log n, как при бинарном поиске).

Похоже, намечается компромисс: можно выполнить меньше попыток, если вы готовы разбить больше банок. Чтобы лучше понять, как этот компромисс работает на количественном уровне, рассмотрим проведение эксперимента с ограниченным «бюджетом» из $k \geq 1$ банок. Другими словами, нужно найти правильный ответ (наивысшую безопасную ступень), использовав не более k банок.

(a) Допустим, вам выделен бюджет k = 2 банки. Опишите стратегию поиска наивысшей безопасной ступени, требующую не более f (n) бросков для некоторой функции f (n) с менее чем линейной скоростью роста. (Иначе говоря, в этом случае $f\ \frac{(n)}{n} = 0$.)

(b) Теперь допустим, что доступен бюджет k > 2 банок с некоторым заданным k. Опишите стратегию поиска наивысшей безопасной ступени с использованием не более k банок. Если обозначить количество бросков по вашей стратегии $f_k(n)$, то функции f_1, f_2, f_3, ... должны обладать тем свойством, что каждая из них растет асимптотически медленнее предыдущей:

$f_k\ \frac{(n)}{f_{k-1}}(n) = 0$для всех k.

Тема 7. Класс сложности NP. Сводимость и NP-полные задачи

Доказать, что следующие задачи принадлежат классу NP.

Задача 3-выполнимости (3 SAT). Требуется узнать, выполнима ли КНФ, в которой каждая элементарная дизъюнкция имеет ранг 3 (состоит из трёх литералов). NP-полнота этой задачи доказывается полиномиальным сведением к ней задачи SAT КНФ.

Задача независимого множества (НМ). Подмножество вершин в графе называется независимым, если в нём нет смежных вершин. Требуется выяснить, есть ли в графе независимое множество заданной мощности. NP-полнота задачи доказывается полиномиальным сведением к ней задачи 3 SAT.

Задача вершинного покрытия (ВП). Подмножество V вершин в графе называется вершинным покрытием графа, если любое ребро графа инцидентно хотя бы одной вершине в V . Требуется выяснить, есть ли в графе вершинное покрытие заданной мощности. NP-полнота задачи доказывается полиномиальным сведением к ней задачи НМ.

Задача о клике. Подмножество вершин в графе называется кликой, если любые две вершины в нём смежные. Задача: есть ли в графе клика заданной мощности? Её NP-полнота доказывается полиномиальным сведением к ней задачи ВП.

Задача изоморфизма: есть ли в графе подграф, изоморфный другому графу. Её NP-полнота доказывается полиномиальным сведением к ней задачи о клике.

Задача раскраски: является ли граф k-раскрашиваемым, т. е. можно ли каждую вершину графа окрасить в один из k цветов, так, что никакие две смежные вершины не будут окрашены в один цвет? NP-полнота задачи доказывается полиномиальным сведением к ней задачи 3SAT.

Задача ориентированного гамильтонова цикла (ОГЦ). Требуется узнать, есть ли в ориентированном графе гамильтонов цикл (проходит через каждую вершину ровно один раз). NP-полнота задачи доказывается полиномиальным сведением к ней задачи 3 SAT.

Задача неориентированного гамильтонова цикла (ГЦ). Требуется узнать, есть ли в неориентированном графе гамильтонов цикл. NP-полнота задачи доказывается сведением к ней задачи ОГЦ.

Задача коммивояжёра (КОМ). Требуется узнать, есть ли в неориентированном графе с целочисленными весами рёбер гамильтонов цикл с заданной суммой весов рёбер. NP-полнота задачи доказывается сведением к ней задачи ГЦ.

Тема 8. Некоторые NP-полные задачи на графах

Решить задачу «КОММИВОЯЖЕР» в оптимизационном варианте (найти в неориентированном нагруженном графе минимальный по стоимости гамильтоновый цикл) с помощью алгоритма с возвратом. Модельный граф не менее чем из 5 вершин придумать самому. Граф можно взять не полный, но обязательно содержащий хотя бы два различных гамильтоновых цикла.

Для решения использовать [4, с. 63–68]

Тема 9. Анализ сложности рекурсивных алгоритмов. Точная оценка сложности алгоритмов

Написать алгоритм, считающий числа Фибоначчи:

F(0)=1; F(1)=1; F(n)=F(n-1)+F(n-2)

1) используя рекурсию;

2) используя мемоизированную рекурсию;

3) используя цикл;

4) используя матричное возведение в степень

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n = \begin{pmatrix} F_{n-1} & F_n \\ F_n & F_{n+1} \end{pmatrix}$$

Оценить для всех алгоритмов временную и пространственную сложность.

Тема 10. Алгоритмы сортировки и их анализ

На любом языке программирования реализовать следующие алгоритмы сортировки:

1 сортировка пузырьком или вставками,

2 быстрая сортировка или сортировка слиянием.

Для реализованных алгоритмов вычислить временную сложность в лучшем, среднем и худшем случае, оценить сложность по памяти.

Провести численный эксперимент, подтверждающий полученные теоретические оценки.

Для выполнения задания рекомендуется пользоваться источниками литературы [6, с. 7–9; 7, с. 37–42].

4 ТРЕБОВАНИЯ К АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

4.1. Текущая аттестация

В ходе изучения дисциплины студентам предстоит пройти следующие этапы текущей аттестации: защита лабораторных работ на протяжение всего курса.

Преподаватель вправе выбрать методику оценивания знаний студентов: традиционная зачетно-экзаменационная, либо балльно-рейтинговая.

4.2. Порядок применения рейтинговой системы

В рамках балльно-рейтинговой системы выставляется оценка за качество выполнения и защиту лабораторных и контрольных работ.

Вид деятельности	Доля	Кол-во ед.	Макс. балл за ед.	Всего
Обязательные виды деятельности				
Посещаемость занятий	40%	N1	=400/N1	400
Выполнение лаб. работ	60%	10	60	600
Итого:				1000

4.3. Условия получения положительной оценки

Для получения оценки зачтено необходимо выполнить все лабораторные работы и защитить их, или в рамках БРС набрать не менее 60% установленных баллов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гашков, С. Б. Теория алгоритмов и вычислений / С. Б. Гашков. – Санкт-Петербург: Лань, 2023. – 168 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/352274 – Режим доступа: для авториз. пользователей.
2. Агибалов, Г. П. Теория вычислительной сложности: учеб. пособие / Г. П. Агибалов. – Томск: ТГУ, 2018. – 42 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/148674. – Режим доступа: для авториз. пользователей.
3. Широков, Д. В. Теория алгоритмов: учеб. пособие / Д. В. Широков. – Киров: ВятГУ, 2017. – 163 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/134610. – Режим доступа: для авториз. пользователей.
4. Сибилева, Н. С. Алгоритмы и теория сложности: учеб. пособие / Н. С. Сибилева, А. С. Файнштейн, С. И. Файнштейн. – Магнитогорск: МГТУ им. Г.И. Носова, 2021. – 93 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/263753. – Режим доступа: для авториз. Пользователей.
5. Белоусов, В. Е. Алгоритмы и анализ сложности: метод. указания / сост.: В. Е. Белоусов [и др.]. – Воронеж: ВГТУ, 2023. – 30 с. – Текст: электронный // Лань: электронно-библиотечная система. – URL: https://e.lanbook.com/book/340367
6. Алексеев, В. Б. Введение в теорию сложности алгоритмов: учеб. пособие для студентов / В. Б. Алексеев. – Москва: Издательский отдел ф-та ВмиК МГУ (лицензия ИД N 05899 от 24.09.2001), 2002 г. – 82 с.
7. Волосевич, А. А. Основы теории алгоритмов: учеб.-метод. пособие по курсу «Теория алгоритмов» для студ. спец. I-31 03 04 «Информатика» всех форм обуч. / А. А. Волосевич. – Минск: БГУИР, 2007. – 54 с.

Локальный электронный методический материал

Татьяна Валентиновна Снытникова

ТЕОРИЯ СЛОЖНОСТИ ВЫЧИСЛЕНИЙ

Редактор Е. Билко

Уч.-изд. л. 1,2. Печ. л. 1,6.

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Калининградский государственный технический университет».
236022, Калининград, Советский проспект, 1