

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
функционального анализа
и операторных уравнений

ka Каменский М.И.
подпись, расшифровка подписи
20.03.2025

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.О.35 Технологии программирования

9. Цели и задачи учебной дисциплины

Целями освоения учебной дисциплины являются:

- овладение базовыми навыками построения и анализа алгоритмов, необходимых для разработки эффективных прикладных программ;
- подготовка к работе с различными структурами данных.

Задачи учебной дисциплины:

- получение практического опыта реализации алгоритмов, выбора подходящей структуры данных для конкретной задачи и проверки корректности эффективности работы заданного алгоритма.

10. Место учебной дисциплины в структуре ООП:

Дисциплина Технологии программирования относится к обязательной части блока Б1. Для изучения и освоения дисциплины нужны первоначальные знания из курсов математического анализа, алгебры, аналитической геометрии. Знания и умения, приобретенные студентами в результате изучения дисциплины, будут использоваться при изучении курсов численных методов, вычислительного практикума, при выполнении курсовых и дипломных работ, связанных с математическим моделированием и обработкой наборов данных.

Дисциплина "Технологии программирования" является предшествующей для изучения следующей дисциплины: «Численные методы».

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ОП К-3	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-3.1	Осуществляет поиск, сбор, хранение, обработку, представление информации при решении задач профессиональной деятельности	знать: принципы работы с поисковыми системами интернета и электронными ресурсами уметь: применять теоретические знания на практике владеть: навыками управления информацией
		ОПК-3.2	Подбирает и использует информационные технологии при решении задач профессиональной деятельности	знать: алгоритмы сортировки, поиска, алгоритмы работы со строками, алгоритмы на графах. уметь: применять перечисленные в графе знать понятия для создания программ владеть: навыками поиска информации по особенностям конкретной поставленной задачи
ОП	Способен	ОПК-	Использует	знать: методы оценки эффективности

K-4	разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	4.1.	основные принципы алгоритмизации задач в рамках профессиональной деятельности и разработки компьютерных программ	алгоритмов уметь: реализовывать алгоритмы с использованием языка С++ владеть: навыками создания эффективных прикладных программ на С++
		ОПК-4.2.	Проводит тестирование и отладку компьютерных программ с целью апробации разработанных моделей и алгоритмов	знать: принципы работы с IDE на примере Visual Studio уметь: использовать механизмы отладки в Visual Studio владеть: навыками тестирования

12. Объем дисциплины в зачетных единицах/час. — 4/144.

Форма промежуточной аттестации экзамен

13. Трудоемкость по видам учебной работы

Вид учебной работы	Трудоемкость		
	Всего	По семестрам	
		2 семестр	
Аудиторные занятия	64	64	
в том числе:	лекции	32	32
	практические		
	лабораторные	32	32
Самостоятельная работа		44	
в том числе: курсовая работа (проект)	44		
Форма промежуточной аттестации	36	36	
Итого:	144	144	

13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК*
1. Лекции			
1.1	Основы анализа	Общая характеристика Классов входных данных. Основные понятия сложности.	https://edu.vsu.ru/user/vie

	алгоритмов	Выбор подходящего языка программирования.	w.php?id=57296&course=5773
1.2	Необходимые математические сведения	Логарифмы, бинарные деревья, О-символика. Классификация скоростей роста	
1.3	Алгоритмы поиска и выборки	Последовательный поиск, двоичный поиск, выборка	
1.4	Алгоритмы сортировки	Сортировка вставками, пузырьковая сортировка, сортировка Шелла, корневая сортировка, сортировка слиянием.	
1.5	Численные алгоритмы	Вычисление значений многочленов, Схема Горнера, умножение матриц, решение линейных уравнений, метод Гаусса-Жордана	
1.6	Алгоритмы сравнения с образцом	Сравнение строк, Конечные автоматы, алгоритм Кнута-Морриса-Пратта, алгоритм Бойера-Мура, приблизительное сравнение строк	
1.7	Алгоритмы на графах	Основные понятия теории графов, структуры данных для представления графов. Определение Матрицы примыканий и Списка примыканий. Алгоритмы обхода в глубину и по уровням. Алгоритм поиска минимального остовного дерева. Алгоритм поиска кратчайшего пути.	
1.8	Параллельные алгоритмы	Введение в параллелизм. Категории компьютерных систем, примеры параллельных архитектур, принципы анализа параллельных алгоритмов. Простые параллельные операции. Параллельный поиск. Параллельная сортировка.	
1.9	Недетерминированные алгоритмы	Определение NP. Сведение данной задачи к другой. NP-полные задачи. Раскраска графа. Раскладка по ящикам. Задача о суммах элементов подмножеств. Проверка возможных решений.	
1.10	Жадные приближенные алгоритмы	Приближения в задаче о коммивояжере. Приближения в задаче о раскладке по ящикам. Приближения в задаче об упаковке рюкзака. Приближения в задаче о раскраске графа.	
2. Практические занятия			
2.1			
3. Лабораторные занятия			
3.1	Основы анализа	Алгоритмы «грубой силы»	https://edu.vs

	алгоритмов		u.ru/user/view.php?id=57296&course=5773
3.2	Необходимые математические сведения	Рекурсия. Логарифмы, бинарные деревья, О-символика.	
3.3	Алгоритмы поиска и выборки	Линейный поиск. Бинарный поиск. Интерполяционный поиск. Асимптотическая сложность алгоритма	
3.4	Алгоритмы сортировки	Базовый случай и рекурсивный случай. Стек. Стек вызовов. Стек вызовов с рекурсией. «Разделяй и властвуй»	
3.5	Численные алгоритмы	Рандомизация данных. Генерирование случайных величин. Рандомизация массивов. Работа с простыми числами: нахождение простых множителей, нахождение простых элементов, проверка на простоту.	
3.6	Алгоритмы сравнения с образцом	Сравнение строк. Конечные автоматы. Приближенное сравнение строк	
3.7	Алгоритмы на графах	Обход в глубину. Обход по уровням. Анализ алгоритмов обхода. Алгоритм поиска минимального остовного дерева. Алгоритм поиска кратчайшего пути	
3.8	Параллельные алгоритмы	Сортировка на линейных сетях. Четно-нечетная сортировка перестановками. Параллельный алгоритм поиска кратчайшего пути.	
3.9	Недетерминированные алгоритмы	Задача планирования работ. Задача о суммах элементов подмножеств. Упаковка рюкзака.	
3.10	Жадные приближенные алгоритмы	Преобразование решения динамического программирования в жадное решение. Оптимальная подструктура задачи о выборе процессов	

13.2. Темы (разделы) дисциплины и виды занятий

№ п/ п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1. 1	Основы анализа алгоритмов	2		2	2	6
1. 2	Необходимые математические сведения	2		2	2	6
1. 3	Алгоритмы поиска и выборки	4		4	6	14
1.	Алгоритмы	4		4	6	14

4	сортировки					
1. 5	Численные алгоритмы	4		4	4	12
1. 6	Алгоритмы сравнения с образцом	4		2	4	10
1. 7	Алгоритмы на графах	4		4	6	14
1. 8	Параллельные алгоритмы	2		4	6	12
1. 9	Недетерминированные алгоритмы	2		4	6	12
1. 10	Жадные приближенные алгоритмы	4		2	2	8
	Итого:	32		32	44	108

14. Методические указания для обучающихся по освоению дисциплины:

Аудиторные занятия, лекции и лабораторные занятия, предполагают самостоятельную работу студентов по данному курсу. Ряд тем выносятся для самостоятельного изучения, предлагаются темы для создания докладов с презентациями. Предусмотрены домашние задания и оформление отчетов выполнения лабораторных заданий, а также дополнительные задания для сильных студентов.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1	Левитин, Ананий. Алгоритмы: введение в разработку и анализ. / Ананий Левитин ; [пер. с англ. Издательский дом «Вильямс»] .— 3-е изд. — Москва ; Санкт-Петербург ; Киев : Вильямс, 2006 .— 576 с.
2	Кормен, Томас. Алгоритмы: построение и анализ. / Т. Кормен, Ч. Лейзерсон, Р. Ривест, — М: МЦНМО, 2001. — 960 с

б) дополнительная литература:

№ п/п	Источник
1	Кнут, Дональд. Искусство программирования, том 1. Основные алгоритмы.. = The Art of Computer Programming. Volume 1. Fundamental Algorithms / под ред. С. Г. Тригуб (гл. 1), Ю. Г. Гордиенко (гл. 2) и И. В. Красикова (разд. 2.5 и 2.6). — 3. — Москва: Вильямс, 2002. — Т. 1. — 720 с.

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет)*:

№ п/п	Ресурс
1	lib.vsu.ru
2	Электронно-библиотечная система "Лань"
3	Электронно-библиотечная система "Университетская библиотека online"
4	Образовательный портал "Электронный университет ВГУ"

16. Перечень учебно-методического обеспечения для самостоятельной работы

№	Источник

п/п	
	Сборник задач по программированию / Н.А. Тюкачев [и др.] — Воронеж : ИПЦ ВГУ, 2010 .— 76 с. <URL: http://www.lib.vsu.ru/elib/texts/method/vsu/m10-212.pdf >

17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ, электронное обучение (ЭО), смешанное обучение):

Дисциплина может реализовываться с применением электронного обучения и дистанционных образовательных технологий. При проведении занятий в дистанционной форме используются технические и информационные ресурсы Образовательного портала "Электронный университет ВГУ" (<https://edu.vsu.ru>), базирующегося на системе дистанционного обучения Moodle, развернутой в университете, а также другие доступные ресурсы в сети Интернет.

18. Материально-техническое обеспечение дисциплины:

Компьютеры, с установленным программным обеспечением Microsoft Visual Studio 2015 Community, LibreOffice.

Для проведения лекционных и лабораторных занятий используются аудитории, соответствующие действующим санитарно-техническим нормам и противопожарным правилам.

Для проведения лабораторных занятий и самостоятельной работы используется класс с компьютерной техникой, оснащенный необходимым программным обеспечением, электронными учебными пособиями и законодательно - правовой и нормативной поисковой системой, имеющий выход в глобальную сеть.

19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1.	Основы анализа алгоритмов	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Практическое задание, контрольная работа
2.	Необходимые математические сведения	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Практическое задание, контрольная работа
3.	Алгоритмы поиска и выборки	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Практическое задание, контрольная работа
4.	Алгоритмы сортировки	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2, ОПК-4.1,ОПК-4.2	Практическое задание, контрольная работа
5.	Численные	ОПК-3, ОПК-4	ОПК-3.1,ОПК-3.2,	Практическое задание, контрольная работа

№ п/п	Наименование раздела дисциплины (модуля)	Компете- нция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
	алгоритмы		ОПК- 4.1,ОПК-4.2	
6.	Алгоритмы сравнения с образцом	ОПК-3, ОПК-4	ОПК- 3.1,ОПК-3.2, ОПК- 4.1,ОПК-4.2	Практическое задание, контрольная работа
7.	Алгоритмы на графах	ОПК-3, ОПК-4	ОПК- 3.1,ОПК-3.2, ОПК- 4.1,ОПК-4.2	Практическое задание, контрольная работа
8.	Параллельные алгоритмы	ОПК-3, ОПК-4	ОПК- 3.1,ОПК-3.2, ОПК- 4.1,ОПК-4.2	Практическое задание, контрольная работа
9.	Недетерминирован ные алгоритмы	ОПК-3, ОПК-4	ОПК- 3.1,ОПК-3.2, ОПК- 4.1,ОПК-4.2	Практическое задание, контрольная работа
10.	Жадные приближенные алгоритмы	ОПК-3, ОПК-4	ОПК- 3.1,ОПК-3.2, ОПК- 4.1,ОПК-4.2	Практическое задание, контрольная работа
Промежуточная аттестация форма контроля – экзамен				Перечень вопросов Практическое задание

20. Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1. Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

Перечень практических заданий

Перечень индивидуальных заданий

1. Разработайте алгоритм для нахождения наибольшего элемента в несортированном однодиаправленном списке с ячейками, содержащими целые числа..
2. Создайте алгоритм для добавления элемента в конец двунаправленного списка.
3. Напишите алгоритм, где значение каждого элемента прямоугольного массива — это расстояние до ближайшего края массива.
4. Напишите программу, которая реализует пузырьковую сортировку.
5. Напишите программу, которая реализует пирамидальную сортировку.
6. Напишите псевдокод, выполняющий обратный симметричный обход в связном сортированном дереве.

Перечень заданий для контрольных работ

Форма контрольно-измерительного материала № 1

УТВЕРЖДАЮ
Заведующий кафедрой функционального анализа и операторных уравнений

подпись расшифровка подписи

—. —. 20 —

Направление подготовки 01.03.04 Прикладная математика
шифр, наименование

Дисциплина Б1.О.35 Технологии программирования

Форма обучения
очная
очное, очно-заочное, заочное

Вид контроля
экзамен
экзамен, зачет;

Вид аттестации
промежуточная
текущая, промежуточная

Контрольно-измерительный материал № 1

- Напишите программу, которая разбирает и вычисляет арифметические выражения, содержащие вещественные числа и операторы +, -, * и /.
- Создайте программу, применяющую интерполяционный поиск.

Преподаватель С.Н. Ушаков
подпись расшифровка подписи

Форма контрольно-измерительного материала № 2

УТВЕРЖДАЮ
Заведующий кафедрой функционального анализа и операторных уравнений

подпись расшифровка подписи

—. —. 20 —

Направление подготовки 01.03.04 Прикладная математика
шифр, наименование

Дисциплина Б1.О.35 Технологии программирования

Форма обучения
очная
очное, очно-заочное, заочное

Вид контроля
экзамен
экзамен, зачет;

Вид аттестации
промежуточная
текущая, промежуточная

Контрольно-измерительный материал № 2

1. Реализуйте алгоритм линейного поиска в отсортированных связных списках.
2. Реализуйте программу рекурсивного интерполяционного поиска.

Преподаватель С.Н. Ушаков
подпись расшифровка подписи

примерные задания для контрольной работы

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Кафедра функционального анализа и операторных уравнений

Комплект заданий для контрольной работы № 1

по дисциплине *Технологии программирования*
(наименование дисциплины)

Темы:..... Алгоритмы сортировки. Сортировка вставкой в массивах

Задание. Составьте алгоритм и напишите программу для решения ниже приведенной задачи. Ввод данных и вывод результатов оформляйте с выводом на экран поясняющих сообщений.

Создайте программу, которая реализует сортировку выбором.

Критерии оценки:

- оценка «отлично» выставляется студенту, если правильно и рационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «хорошо» выставляется студенту, если правильно, но нерационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «удовлетворительно» выставляется студенту, если составлен алгоритм с негрубыми ошибками и написана работающая программа, соответствующая составленному алгоритму;
- оценка «неудовлетворительно» выставляется студенту, если алгоритм не составлен или составлен алгоритм с грубыми ошибками; не написана программа или написана неработающая программа.

Комплект заданий для контрольной работы № 2

по дисциплине *Технологии программирования*

Тема Алгоритмы на графах

Задание. Напишите псевдокод, выполняющий обратный симметричный обход в связном сортированном дереве.

- оценка «отлично» выставляется студенту, если правильно и рационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «хорошо» выставляется студенту, если правильно, но нерационально составлен алгоритм, написана работающая программа, соответствующая составленному алгоритму;
- оценка «удовлетворительно» выставляется студенту, если составлен алгоритм с негрубыми ошибками и написана работающая программа, соответствующая составленному алгоритму;

- оценка «неудовлетворительно» выставляется студенту, если алгоритм не составлен или составлен алгоритм с грубыми ошибками; не написана программа или написана неработающая программа.

20.2. Промежуточная аттестация

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств: экзаменационные билеты.

Билет на экзамене состоит из пяти вопросов. Каждый вопрос оценивается в 2 балла. Для получения оценки отлично нужно набрать 10 баллов, для хорошо - 8-9, удовлетворительно - 6-7. На выполнение заданий выделяется 120 минут. После будет разговор по этому билету. Если студент не сможет объяснить своё решение или доказательство, то оценка будет снижена. За работу на практических занятиях вы можете заработать дополнительные (до 5) баллы.

Структура билета:

1. Теоретический вопрос.
2. Решение рекуррентного соотношения или, если это невозможно, оценка скорости его роста.
3. Доказательство теоремы или примеры применения алгоритма, или исправление кода известного алгоритма.
4. Анализ эффективности алгоритма.
5. Написать функцию по структуре данных (списки, деревья).

Теоретический вопрос обязателен, в случае неудовлетворительного ответа на него экзамен прекращается с оценкой неудовлетворительно.

Перечень вопросов к экзамену:

1. Понятие алгоритма. Основы решения алгоритмической задачи.
2. Важные типы алгоритмических задач.
3. Последовательности и рекуррентные соотношения. Определения последовательности, рекуррентного уравнения и соотношения, общего и частного решений рекуррентного соотношения. Методы решения рекуррентных соотношения, метод прямой подстановки, метод обратной подстановки.
4. Линейные рекуррентные соотношения второго порядка с постоянными коэффициентами. Теоремы о решении рекуррентного соотношения второго порядка с постоянными коэффициентами. Примеры.
5. Распространенные типы рекуррентных соотношений в анализе алгоритмов. Уменьшение на единицу. Уменьшение на постоянный множитель. Декомпозиция.
6. Теорема 4 (правило гладкости). Теорема 5 (основная теорема).
7. Основы анализа эффективности алгоритмов. Оценка размера входных данных. Единицы измерения времени выполнения алгоритма. Эффективность в разных случаях. Общее рекуррентное соотношение декомпозиции
8. Асимптотические обозначения. Полезные свойства сделанных асимптотических обозначений (теорема с доказательством!).
9. Основные классы эффективности.
10. Математический анализ нерекурсивных алгоритмов. Общий план анализа нерекурсивных алгоритмов. Примеры.
11. Математический анализ рекурсивных алгоритмов. Общий план анализа нерекурсивных алгоритмов. Примеры.

12. Метод грубой силы. Сортировка выбором.
13. Метод грубой силы. Пузырьковая сортировка.
14. Метод грубой силы. Последовательный поиск и поиск подстрок методом грубой силы.
15. Метод грубой силы. Поиск пары ближайших точек.
16. Исчерпывающий перебор. Задача коммивояжера. Задача о рюкзаке.
17. Метод декомпозиции. Основная теорема. Сортировка слиянием.
18. Метод декомпозиции. Основная теорема. Быстрая сортировка.
19. Метод декомпозиции. Основная теорема. Бинарный поиск.
20. Метод декомпозиции. Основная теорема. Задача о паре ближайших точках.
21. Базовые структуры данных. Линейные структуры данных.
22. Базовые структуры данных. Графы. Представление графов. Взвешенные графы. Пути и циклы.
23. Базовые структуры данных. Графы. Поиск в глубину.
24. Базовые структуры данных. Графы. Поиск в ширину.
25. Базовые структуры данных. Деревья. Корневые деревья.
26. Базовые структуры данных. Упорядоченные деревья.
27. Базовые структуры данных. Обход бинарного дерева.
28. Базовые структуры данных. Поиск и вставка в бинарное дерево поиска.
29. Метод уменьшения размера задач. Сортировка вставкой.
30. Метод преобразования. Понятие пирамиды. Восходящие построение пирамиды.
31. Метод преобразования. Понятие пирамиды. Нисходящие построение пирамиды.
32. Метод преобразования. Пирамидальная сортировка.
33. Метод преобразования. Схема Горнера.

Примерный вариант билета

1. Метод декомпозиции. Основная теорема. Быстрая сортировка.
2. Найдите порядок роста следующего рекуррентного соотношения:

$$T(n) = 3 T\left(\frac{n}{9}\right) + n^2 + 15n, \quad T(1) = 1.$$

ИЛИ

Решите рекуррентное соотношение второго порядка:

$$\begin{cases} a(n) = a(n-1) + n, & n > 1, \\ a(1) = 2. \end{cases}$$

ИЛИ

Решите рекуррентное соотношение второго порядка:

$$\begin{cases} a(n) = a(n-1) + a(n-2), & n > 2, \\ a(1) = 1, \quad a(2) = 1. \end{cases}$$

3. Отсортируйте список с помощью пирамидальной сортировки с использованием представления пирамид в виде массивов:

S, O, R, T, I, N, G (в алфавитном порядке).

ИЛИ

Докажите теорему ниже

Теорема. Пусть $f(n)$ — гладкая функция. Тогда для любого целого $b \geq 2$,

$$f(bn) \in O(f(n))$$

т.е. существует положительная константа c_b и неотрицательное целое число n_0 такие, что

$$f(bn) \leq c_b f(n)$$

4. Рассмотрим следующий алгоритм.

Алгоритм Secret(A[0..n-1, 0..n-1])

for $i \leftarrow 0$ **to** $n-1$ **do**

for $j \leftarrow i$ **to** $n-1$ **do**

swap(A[i,j], A[j,i]) // swap – функция обмена

а) Что делает этот алгоритм?

б) Назовите основную операцию алгоритма.

в) Сколько раз выполняется основная операция?

г) К какому классу эффективности относится этот алгоритм?

д) Усовершенствуйте этот алгоритм или предложите другой и оцените его класс эффективности.

5. Для решения (обязательно!!!) опишите структуру данных, используемую в функции

Напишите функцию поиска разности максимального и минимального элемента в бинарном дереве поиска.

Для оценивания результатов обучения на экзамене используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
<i>Обучающийся в полной мере владеет понятийным аппаратом теории алгоритмов. Способен иллюстрировать ответ примерами, применять теоретические знания для создания алгоритмов и реализации в виде программ на языке C++</i>	<i>Повышенный уровень</i>	<i>Отлично</i>
<i>Обучающийся владеет понятийным аппаратом теории алгоритмов. Способен иллюстрировать ответ примерами допускает ошибки при создании алгоритмов и реализации в виде программ на языке C++ или способен применять теоретические знания для создания алгоритмов и реализации в виде программ на языке C++, но допускает неточности при применении понятийного аппарата, но отвечает на дополнительные вопросы</i>	<i>Базовый уровень</i>	<i>Хорошо</i>
<i>Обучающийся владеет частично теоретическими основами дисциплины, фрагментарно способен иллюстрировать ответ примерами, фактами, не отвечает на дополнительные вопросы. Не умеет применять теоретические знания для создания алгоритмов и реализации в виде программ на языке C++</i>	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
<i>Ответ на контрольно-измерительный материал не соответствует любым трем(четырем) из перечисленных показателей. Обучающийся демонстрирует отрывочные, фрагментарные знания, допускает грубые ошибки.</i>	<i>–</i>	<i>Неудовлетворительно</i>

20.3 Фонд оценочных средств сформированности компетенций студентов, рекомендуемый для проведения диагностических работ

Задания закрытого типа:

1. Определите к какому асимптотическому классу эффективности по числу операций умножения относится алгоритм, реализованный в функции *f()*.

```
double f(double x, double* P, int n)
{
    double p = 0.0;
    for (int i = 0; i < n; i++) {
        int step = 1;
        for (int j = 0; j < i; j++)
        {
            step *= x;
        }
        p += P[i] * step;
    }
    return p;
}
```

1. линейный
2. квадратичный
3. кубический
4. логарифмический
5. экспоненциальный
6. константа

Ответ: 2

Решение. Посчитаем число операций в первом цикле. Это число не зависит от размера входных данных. При каждом i – операций $i+1$, то есть нужно посчитать сумму вида $1+2+\dots+n$. Это арифметическая прогрессия, её сумма равна $(n-1)*n/2$. Порядок роста этой суммы – квадратичный.

2. Определите к какому асимптотическому классу эффективности в лучшем случае относится алгоритм, реализованный в функции $f()$.

```
bool f(double* a, int n)
{
    for (int i = 0; i < n - 2; i++)
        for (int j = i + 1; j < n - 1; j++)
            if (a[i] == a[j])
                return false;
    return true;
}
```

- 1. линейный
- 2. квадратичный
- 3. кубический
- 4. логарифмический
- 5. экспоненциальный
- 6. константа

Ответ: 6

Решение. Функция $f()$, исходя из представленного кода, возвращает $false$, если в массиве есть одинаковые элементы и $true$ – в противном случае. Наименьшее число операций сравнения будет, если два первых элемента массива совпадают. В этом случае произойдёт ровно одно сравнение, поэтому асимптотический класс эффективности этого алгоритма – константа.

3. Определите к какому асимптотическому классу эффективности в худшем случае относится алгоритм, реализованный в функции $f()$.

```
bool f(double* a, int n)
{
    for (int i = 0; i < n - 2; i++)
        for (int j = i + 1; j < n - 1; j++)
            if (a[i] == a[j])
                return false;
    return true;
}
```

- 1. линейный
- 2. квадратичный
- 3. кубический
- 4. логарифмический
- 5. экспоненциальный
- 6. константа

Ответ: 2

Решение. Функция $f()$, исходя из представленного кода, возвращает $false$, если в массиве есть одинаковые элементы и $true$ – в противном случае. Наибольшее число операций сравнения будет, если все элементы массива различны. В этом случае будут выполнены все операции сравнения в двух циклах. При каждом i – число операций во втором цикле равно $n-1-i-1$. То есть нужно посчитать эту сумму при i от 0 до $n-3$ включительно. Это сумма чисел $(n-2)+(n-1)+\dots+1$. Это арифметическая прогрессия, её сумма равна $(n-1)*(n-2)/2$. Порядок роста этой суммы – квадратичный.

Задания открытого типа:

1. Будем считать, что одна операция сравнения a и b позволяет установить меньше ли a b , больше ли a b или они равны. Определите максимальное число таких сравнений при бинарном поиске в массиве [5, 16, 29, 33, 41, 44, 57, 72, 76, 83, 87, 95, 100].

Ответ: 4

Решение. Наибольшее число сравнений при бинарном поиске в отсортированном массиве можно найти по формуле $\lceil \log_2 n \rceil + 1$, где $\lceil \cdot \rceil$ – обозначение для целой части числа. В нашем случае $\lceil \log_2 13 \rceil + 1 = 4$. Получить ответ можно и эмпирическим путём, например, каждый раз при поиске выбирать самый длинный путь (ищем число 44, тогда $44 < 57$, $44 > 29$, $44 > 41$, $44 = 44$).

2. Найдите $x(123)$, которое задано рекуррентным соотношением

$$x(n) = x(n - 1) + 6, \text{ для всех } n > 1, x(1) = 0.$$

Ответ: 732

Решение. Выпишем в начале несколько членов этой последовательности

$$x(2) = x(1) + 6 = 6, x(3) = x(2) + 6 = 12, x(4) = x(3) + 6 = 18.$$

Можно заметить, что каждый новый член последовательности больше предыдущего на одно и тоже число, на 6. Таким образом, это арифметическая последовательность. Общая формула в этом случае имеет вид:

$$x(n) = 6(n - 1) \\ x(123) = 6(123 - 1) = 732$$

3. Найдите $x(128)$, которое задано рекуррентным соотношением

$$x(n) = x\left(\frac{n}{2}\right) + n, \text{ для всех } n > 1, x(1) = 1.$$

Ответ: 255

Решение. Выпишем в начале несколько членов этой последовательности

$$x(2^1) = x(2/2) + 2 = 1 + 2 = 3, \quad x(2^2) = x(2^2/2) + 2^2 = x(2) + 4 = 1 + 2 + 4 = 7, \\ x(2^3) = x(2^3/2) + 2^3 = x(4) + 8 = 1 + 2 + 4 + 8 = 15.$$

Можно заметить, что каждый новый член последовательности равен соответствующей степени двойки минус один. Общая формула в этом случае имеет вид:

$$x(2^k) = 2^{k+1} - 1 \\ x(128) = x(2^7) = 2^8 - 1 = 256 - 1 = 255$$

Задания раздела 20.3 рекомендуются к использованию при проведении диагностических работ с целью оценки остаточных результатов освоения данной дисциплины (знаний, умений, навыков).