

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Богдалова Елена Вячеславовна

Должность: Проректор по образовательной деятельности

Дата подписания: 21.08.2025 12:33:21

Уникальный программный ключ:

ec85dd5a839619d48ea76b2d23dba88a9c83070c

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение инклюзивного высшего образования
«Российский государственный университет
социальных технологий»
(ФГБОУ ИВО «РГУ СоцТех»)

УТВЕРЖДАЮ

Проректор по образовательной деятельности

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Б1.В.ДЭ.05.02 Теория алгоритмов

наименование дисциплины

09.03.03 «Прикладная информатика»

шифр и наименование направления подготовки

Цифровая трансформация

направленность (профиль)

Москва 2024

Содержание

- 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ**
- 2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ**
- 3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ 4.**
МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ
ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ
ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ
- 5. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И**
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Теория алгоритмов»

Оценочные средства составляются в соответствии с рабочей программой дисциплины и представляют собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.), предназначенных для измерения уровня достижения обучающимися установленных результатов обучения.

Оценочные средства используются при проведении текущего контроля успеваемости и промежуточной аттестации.

Таблица 1 - Перечень компетенций, формируемых в процессе освоения дисциплины

Код компетенции	Наименование результата обучения
ПК-10	Способен применять системный подход и математические методы в формализации решения прикладных задач
	ПК-10.1. Знает базовые положения фундаментальных разделов системного анализа и математики в объеме, необходимом для обработки информации и анализа данных в прикладной области; принципы и методы проведения исследований в области информационных систем и технологий; техники планирования и проведения вычислительного эксперимента.
	ПК-10.2. Умеет формулировать и доказывать наиболее важные результаты в прикладных областях; применять численные методы для решения прикладных задач; программно реализовать вычислительный эксперимент посредством языков программирования или с использованием специализированных пакетов прикладных программ; разрабатывать алгоритмы решения конкретных задач.
	ПК-10.3. Владеет навыками постановки задачи; навыками работы с библиографическими источниками информации; навыками решения поставленных задач в предметной области в рамках выбранного профиля.

Конечными результатами освоения дисциплины являются сформированные когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям. Формирование дескрипторов происходит в течение всего семестра по этапам в рамках контактной работы, включающей различные виды занятий и самостоятельной работы, с применением различных форм и методов обучения (табл.2).

Таблица 2 - Формирование компетенций в процессе изучения дисциплины:

Код компетенции	Уровень освоения компетенций	Индикаторы достижения компетенций	Вид учебных занятий ¹ , работы, формы и методы обучения, способствующие формированию и развитию компетенций ²	Контролируемые разделы и темы дисциплины ³	Оценочные средства, используемые для оценки уровня сформированности компетенции ⁴
ПК-10		<i>Знает</i>			
	Недостаточный уровень	ПК-10. Студент не знает базовые положения фундаментальных разделов математики в объеме, необходимом для обработки информации и анализа данных в прикладной области; принципы и методы проведения исследований в области информационных систем и технологий; техники планирования и проведения вычислительного эксперимента. Студент не знает основные принципы построения поисковых, сортирующих и вычислительных алгоритмов, основы стратегии разработки алгоритмов. Студент не умеет определять наихудшее и среднее время работы	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации.	Раздел 1. Исчисление высказываний. Раздел 2. Логика предикатов первого порядка. Раздел 3. Исчисление предикатов.	Текущий контроль – устный опрос, контрольная работа.

¹ Лекционные занятия, практические занятия, лабораторные занятия, самостоятельная работа...

² Необходимо указать активные и интерактивные методы обучения (например, интерактивная лекция, работа в малых группах, методы мозгового штурма и т.д.), способствующие развитию у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств.

³ Наименование темы (раздела) берется из рабочей программы дисциплины.

⁴ Оценочное средство должно выбираться с учетом запланированных результатов освоения дисциплины, например:

«Знать» – собеседование, коллоквиум, тест...

«Уметь», «Владеть» – индивидуальный или групповой проект, кейс-задача, деловая (ролевая)

игра, портфолио...

		алгоритмов; выбирать и применять			
--	--	----------------------------------	--	--	--

		различные методы для разработки алгоритмов; проводить сравнительный Не владеет технологиями оценки эффективности алгоритмов; навыками доказательства корректности алгоритмов; навыками производить алгоритмизацию и анализ времени работы алгоритмов.			
--	--	--	--	--	--

Базовый уровень	ПК-10.1. Студент имеет несистематизированные знания базовых положений фундаментальных разделов математики в объеме, необходимом для обработки информации и анализа данных в прикладной области; принципах и методах проведения исследований в области информационных систем и технологий; техники планирования и проведения вычислительного эксперимента. Студент имеет несистематизированные знание основных принципов построения поисковых, сортирующих и вычислительных алгоритмов, основы стратегии разработки алгоритмов.	Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации.	Раздел 1. Исчисление высказываний. Раздел 2. Логика первого предикатов Раздел 3. Исчисление предикатов.	Текущий контроль – устный опрос, контрольная работа.
Средний уровень	ПК-10.1. Студент знает основное содержание материала дисциплины. Студент самостоятельно выделяет главные положения в изученном материале. Знает основные базовые положения фундаментальных разделов	Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся,	Раздел 1. Исчисление высказываний. Раздел 2. Логика первого предикатов Раздел 3. Исчисление предикатов.	Текущий контроль – устный опрос, контрольная работа.

		математики в объеме, необходимом для обработки информации и анализа данных в прикладной области; принципы и методы проведения исследований в области информационных систем и технологий; техники планирования и проведения вычислительного эксперимента. Студент знает основные принципы построения поисковых, сортирующих и вычислительных алгоритмов, основы стратегии разработки алгоритмов, но допускает незначительные ошибки	подготовка и сдача промежуточной аттестации.		
	Высокий уровень	ПК-10.1. Студент самостоятельно выделяет главные положения в изученном материале. Знает базовые положения фундаментальных разделов математики в объеме, необходимом для обработки информации и анализа данных в прикладной области; принципы и методы проведения исследований в области информационных систем и технологий; техники планирования и проведения вычислительного эксперимента. Студент знает принципы построения поисковых, сортирующих и вычислительных алгоритмов; основы стратегии разработки алгоритмов.	Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации.	Раздел 1. Исчисление высказываний. Раздел 2. Логика первого предикатов порядка. Раздел 3. Исчисление предикатов.	Текущий контроль – устный опрос, контрольная работа.
		<i>Умеет</i>			

Базовый уровень	ПК-10.2. Студент испытывает затруднения при выборе стратегии разработки алгоритмов. Студент не умеет формулировать и доказывать наиболее важные результаты в прикладных областях; программно реализовать вычислительный эксперимент посредством языков программирования или с использованием специализированных пакетов прикладных программ; разрабатывать алгоритмы решения конкретных задач. Студент испытывает затруднения при определении наихудшего и среднего времени работы алгоритмов; выборе и применении различных методов для разработки алгоритмов; проведении сравнительного анализа времени работы алгоритмов.	Лекционные и практические занятия, малых групп, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка промежуточной аттестации.	Раздел 1. Исчисление высказываний. Раздел 2. Логика первого порядка. Раздел 3. Исчисление предикатов.	Текущий контроль – устный опрос, контрольная работа.
Средний уровень	ПК-10.2 Студент умеет формулировать и доказывать наиболее важные результаты в прикладных областях; программно реализовать вычислительный эксперимент посредством языков программирования или с использованием специализированных пакетов прикладных программ; разрабатывать алгоритмы решения конкретных задач. Студент умеет определять наихудшее	Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации.	Раздел 1. Исчисление высказываний. Раздел 2. Логика первого порядка. Раздел 3. Исчисление предикатов.	Текущий контроль – устный опрос, контрольная работа.

		и среднее время работы алгоритмов; выбирать и применять различные методы для разработки алгоритмов; проводить сравнительный анализ времени работы алгоритмов, но допускает незначительные ошибки.			
	Высокий уровень	ПК-10.2. Студент умеет анализировать прикладную задачу, находить оптимальные методы ее решения. Студент умеет формулировать и доказывать наиболее важные результаты в прикладных областях; программно реализовать вычислительный эксперимент посредством языков программирования или с использованием специализированных пакетов прикладных программ; разрабатывать алгоритмы решения конкретных задач. Студент умеет определять наихудшее и среднее время работы алгоритмов; выбирать и применять различные методы для разработки алгоритмов; проводить сравнительный анализ времени работы алгоритмов.	Лекционные и практические занятия, малых интерактивная лекция, дискуссия, самостоятельная обучающихся, подготовка промежуточной аттестации.	Раздел 1. Исчисление высказываний. Раздел 2. Логика первого предикатов порядка. Раздел 3. Исчисление предикатов.	Текущий контроль – устный опрос, контрольная работа.
		<i>Владеет</i>			
	Базовый уровень	ПК-10.3. Студент на базовом уровне владеет навыками постановки задачи; навыками работы с библиографическими источниками информации; навыками решения поставленных задач в предметной	Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа	Раздел 1. Исчисление высказываний. Раздел 2. Логика первого предикатов порядка. Раздел 3. Исчисление	Текущий контроль – устный опрос, контрольная работа.

		области в рамках выбранного профиля; технологиями оценки эффективности алгоритмов; навыками доказательства корректности алгоритмов; навыками производить алгоритмизацию.	обучающихся, подготовка и сдача промежуточной аттестации.	предикатов.	
Средний уровень	ПК-10.3. Студент владеет знаниями всего изученного материала. Владеет навыками постановки задачи; навыками работы с библиографическими источниками информации; навыками решения поставленных задач в предметной области в рамках выбранного профиля. Студент владеет технологиями оценки эффективности алгоритмов; навыками доказательства корректности алгоритмов; навыками производить алгоритмизацию, но допускает незначительные ошибки.	Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации.	Раздел 1. Исчисление высказываний. Раздел 2. Логика первого порядка. Раздел 3. Исчисление предикатов.	Текущий контроль – устный опрос, контрольная работа.	
Высокий уровень	ПК-10.3. Студент владеет концептуально-понятийным аппаратом, научным языком и терминологией. Владеет навыками постановки задачи; навыками работы с библиографическими источниками информации; навыками решения поставленных задач в предметной области в рамках выбранного профиля. Студент владеет технологиями оценки эффективности алгоритмов; навыками доказательства корректности алгоритмов; навыками производить	Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации.	Раздел 1. Исчисление высказываний. Раздел 2. Логика первого порядка. Раздел 3. Исчисление предикатов.	Текущий контроль – устный опрос, контрольная работа.	

		алгоритмизацию.			
--	--	-----------------	--	--	--

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ⁵

Таблица 3

№	Наименование оценочного средства	Характеристика оценочного средства	Представление оценочного средства в ФОС
1.	Устный опрос	Средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде собеседования преподавателя с обучающимися.	Вопросы по темам/разделам дисциплины
2.	Контрольная работа	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу	Комплект контрольных заданий по вариантам

3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Оценивание результатов обучения по дисциплине «Математика» осуществляется в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся.

Предусмотрены следующие виды контроля: текущий контроль (осуществление контроля всех видов аудиторной и внеаудиторной деятельности обучающегося с целью получения первичной информации о ходе усвоения отдельных элементов содержания дисциплины) и промежуточная аттестация (оценивается уровень и качество подготовки по дисциплине в целом).

Показатели и критерии оценивания компетенций, формируемых в процессе освоения данной дисциплины, описаны в табл. 4.

Таблица 4.

⁵ Указываются оценочные средства, применяемые в ходе реализации рабочей программы данной дисциплины.

Код компетенции	Уровень освоения компетенции	Индикаторы достижения компетенции	Критерии оценивания результатов обучения
ПК-10		Знает	
	Недостаточный уровень Оценка «незачтено».	ПК-10.1.	<i>Не знает значительной части материала курса, не способен самостоятельно выделять главные положения в изученном материале дисциплины.</i>
	Базовый уровень Оценка «зачтено».	ПК-10.1.	<i>Знает не менее 50 % основного материала курса, однако испытывает затруднения в его применении.</i>
	Средний уровень Оценка «зачтено».	ПК-10.1.	<i>Знает основную часть материала курса, способен применить изученный материал на практике, испытывает незначительные затруднения в решении профессиональных задач.</i>
	Высокий уровень Оценка «зачтено».	ПК-10.1.	<i>Показывает глубокое знание и понимание материала, способен применить изученный материал на практике.</i>
		Умеет	
	Базовый уровень	ПК-10.2.	<i>Умеет воспроизвести не менее 50 % основного материала курса, однако испытывает затруднения при решении практических задач.</i>
	Средний уровень	ПК-10.2.	<i>Умеет решать стандартные профессиональные задачи с применением полученных знаний, испытывает незначительные затруднения в решении задач.</i>
	Высокий уровень	ПК-10.2.	<i>Умеет решать стандартные профессиональные задачи с применением полученных знаний, показывает глубокое знание и понимание материала, способен решить задачу при изменении формулировки.</i>
		Владеет	
	Базовый уровень	ПК-10.3.	<i>Студент владеет основными навыками постановки задачи; навыками работы с библиографическими источниками информации; навыками решения поставленных задач в предметной области в рамках выбранного профиля; технологиями оценки эффективности алгоритмов; навыками доказательства корректности алгоритмов; навыками производить алгоритмизацию. Имеет несистематизированные знания основных разделов дисциплины.</i>

	Средний уровень	ПК-10.3.	Студент владеет знаниями всего изученного материала, владеет навыками постановки задачи; навыками работы с библиографическими источниками информации; навыками решения поставленных задач в предметной области в
			рамках выбранного профиля; технологиями оценки эффективности алгоритмов; навыками доказательства корректности алгоритмов; навыками производить алгоритмизацию. Испытывает незначительные затруднения в решении практических задач.
	Высокий уровень	ПК-10.3.	Свободно владеет навыками постановки задачи; навыками работы с библиографическими источниками информации; навыками решения поставленных задач в предметной области в рамках выбранного профиля; технологиями оценки эффективности алгоритмов; навыками доказательства корректности алгоритмов; навыками производить алгоритмизацию. Студент владеет концептуально-понятийным аппаратом, научным языком и терминологией профессиональной деятельности.

4. Методические материалы, определяющие процедуры оценивания результатов обучения

Задания в форме устного опроса:

Устный опрос используется для текущего контроля успеваемости обучающихся по дисциплине в качестве проверки результатов освоения материала. Каждому студенту выдается свой собственный, узко сформулированный вопрос. Ответ должен быть четким и кратким, содержащим все основные характеристики описываемого понятия. В своем ответе студент должен показать умения прослеживать причинно-следственные связи и навыки рассуждений и доказательства.

Контрольная работа

Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу

5. Материалы для проведения текущего контроля и промежуточной аттестации

Задания в форме устного опроса Что такое алгоритмы.

- 1) Приведите реальные примеры задач, в которых возникает потребность в сортировке или вычислении выпуклой оболочки.
- 2) Какими ещё параметрами, кроме скорости, можно характеризовать алгоритм на практике?
- 3) Выберите одну из встречавшихся вам ранее структур данных и опишите её преимущества и ограничения.
- 4) Что общего между задачей об определении кратчайшего пути и задачей о коммивояжере? Чем они различаются?
- 5) Сформулируйте задачу, в которой необходимо только наилучшее решение. Сформулируйте также задачу, в которой может быть приемлемым решение, достаточно близкое к наилучшему.

Алгоритмы как технология.

- 1) Приведите пример приложения, для которого необходимо алгоритмическое наполнение на уровне приложений, и обсудите функции этих алгоритмов.
- 2) Предположим, на одной и той же машине проводится сравнительный анализ реализаций двух алгоритмов сортировки, работающих вставкой и слиянием. Для сортировки n элементов вставкой необходимо $8n^2$ шагов, а для сортировки слиянием – $64n \lg n$ шагов. При каком значении n время сортировки вставкой превысит время сортировки слиянием?

3) При каком минимальном значении n алгоритм, время работы которого определяется формулой $100n^2$, работает быстрее, чем алгоритм, время работы которого выражается как 2^n , если оба алгоритма выполняются на одной и той же машине?

Сортировка вставкой.

1) Используя рис.2.2 в качестве образца, проиллюстрируйте работу процедуры Insertion-Sort по сортировке массива $A = \langle 31, 41, 59, 26, 41, 58 \rangle$.

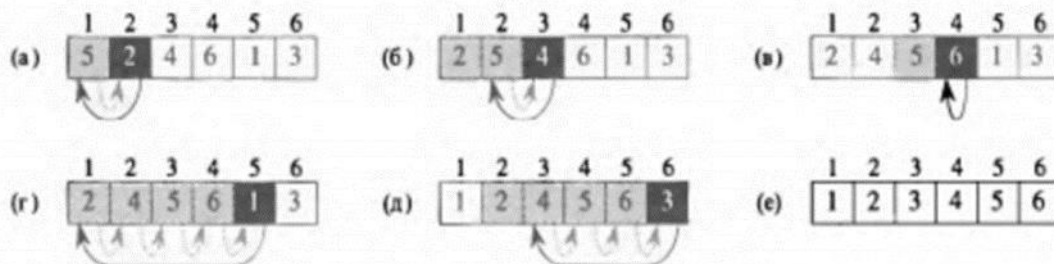


Рис. 2.2. Операции процедуры INSERTION-SORT над массивом $A = \langle 5, 2, 4, 6, 1, 3 \rangle$. Элементы массива обозначены квадратиками, над которыми находятся индексы, а внутри — значения соответствующих элементов. Части (а)–(д) этого рисунка соответствуют итерациям цикла **for** в строках 1–8 псевдокода. В каждой итерации черный квадратик содержит значение ключа из $A[j]$, которое сравнивается со значениями серых квадратиков, расположенных слева от него (строка псевдокода 5). Серыми стрелками указаны те значения массива, которые сдвигаются на одну позицию вправо (строка 6), а черной стрелкой — перемещение ключа (строка 8). В части (е) показано конечное состояние отсортированного массива.

2) Перепишите процедуру Insertion-Sort для сортировки в невозрастающем порядке вместо неубывающего.

3) Рассмотрим задачу поиска.

Вход. Последовательность из n чисел $A = \langle a_1, a_2, \dots, a_n \rangle$ и значение v . Выход.

Индекс i , такой, что $v = A[i]$, или специальное значение NIL, если v в A отсутствует.

Составьте псевдокод линейного поиска, при работе которого выполняется сканирование последовательности в поисках значения v . Докажите корректность алгоритма с помощью инвариантности цикла. Убедитесь, что выбранный инвариант цикла удовлетворяет трем необходимым условиям.

4) Рассмотрим задачу сложения двух n -битовых двоичных целых чисел, хранящихся в n -элементных массивах A и B . Сумму этих двух чисел необходимо занести в двоичной форме в $(n+1)$ -элементный массив C . Приведите строгую формулировку задачи и составьте псевдокод для сложения этих двух чисел.

Анализ алгоритмов.

1) Выразите функцию $n^3/1000 - 100n^2 - 100n + 3$ Θ -обозначениях.

2) Рассмотрим сортировку элементов массива A , которая выполняется следующим образом. Сначала определяется наименьший элемент массива A , который становится на место элемента $A[1]$. Затем производится поиск второго

наименьшего элемента массива A , который становится на место элемента $A[2]$. Этот процесс продолжается для первых $n-1$ элементов массива A . Запишите псевдокод этого алгоритма, известного как сортировка выбором (selection sort). Какой инвариант цикла сохраняется для этого алгоритма? Почему его достаточно выполнить для первых $n-1$ элементов, а не для всех n элементов? Определите время работы алгоритма в наилучшем и наихудшем случаях и запишите его в Θ -обозначениях.

- 3) Вновь обратимся к алгоритму линейного поиска (см. сортировка вставкой 3). Для скольких элементов входной последовательности в среднем нужно произвести проверку, если предполагается, что все элементы массива с равной вероятностью могут иметь искомое значение? Что происходит в наихудшем случае? Чему равно время работы алгоритма линейного поиска в среднем и в наихудшем случаях в Θ -обозначениях?

Обоснуйте свой ответ.

- 4) Каким образом можно модифицировать почти каждый алгоритм, чтобы получить оптимальное время работы в наилучшем случае?

Разработка алгоритмов.

- 1) Используя в качестве образца рис 2.4, проиллюстрируйте работу алгоритма сортировки слиянием для массива $A = \langle 3, 41, 52, 26, 38, 57, 9, 49 \rangle$.



Рис. 2.4. Процесс сортировки слиянием массива $A = \langle 5, 2, 4, 7, 1, 3, 2, 6 \rangle$. Длины подлежащих слиянию отсортированных последовательностей возрастают в ходе работы алгоритма.

- 2) Перепишите процедуру Merge так, чтобы в ней не использовались сигнальные значения. Сигналом к остановке должен служить тот факт, что все элементы массива L

или массива R скопированы обратно в массив A, после чего в этот массив копируются элементы, оставшиеся в непустом массиве.

- 3) Воспользуйтесь методом математической индукции для доказательства того, что, когда n является точной степенью 2, решением рекуррентного соотношения

$$2, \quad \text{если } n = 2$$

$$T(n) = \begin{cases} 2T(n/2) + n, & \text{если } n = 2^k, \\ k > 1 \end{cases}$$

является $T(n) = n \lg n$.

- 4) Сортировку вставкой можно представить в виде рекурсивной последовательности. Чтобы отсортировать массив $A[1 \dots n]$, сначала нужно рекурсивно отсортировать массив $A[1 \dots n - 1]$, после чего в этот отсортированный массив помещается элемент $A[n]$. Запишите рекуррентное уравнение для времени работы этой рекурсивной версии сортировки вставкой.
- 5) Возвращаясь к задаче поиска (см. сортировка вставкой 3), нетрудно заметить, что если последовательность A отсортирована, то можно сравнить значение среднего элемента этой последовательности с искомым значением v и сразу исключить половину последовательности из дальнейшего рассмотрения. Бинарный поиск (binary search) – это алгоритм, в котором такая процедура повторяется неоднократно, что всякий раз приводит к уменьшению оставшейся части последовательности в два раза. Запишите псевдокод алгоритма бинарного поиска (либо итеративный, либо рекурсивный). Докажите, что время работы этого алгоритма в наихудшем случае составляет $\Theta(\lg n)$.
- 6) Заметим, что в цикле while в строках 5-7 процедуры Insertion-Sort в разделе 2.1 для сканирования (в обратном порядке) отсортированного подмассива $A[1 \dots j - 1]$ используется линейный поиск. Можно ли использовать бинарный поиск (см. упр. 2.3.5) вместо линейного, чтобы время работы этого алгоритма в наихудшем случае улучшилось и стало равным $\Theta(n \lg n)$.
- 7) Разработайте алгоритм со временем работы $\Theta(n \lg n)$, который для заданного множества S из n целых чисел и другого целого числа x определяет, имеются ли во множестве S два элемента, сумма которых равна x .

Асимптотические обозначения.

- 1) Пусть $f(n)$ и $g(n)$ – асимптотически неотрицательные функции. Докажите с помощью базового определения Θ -обозначений, что $\max(f(n), g(n)) = \Theta(f(n) + g(n))$.
- 2) Покажите, что для любых действительных констант a и b , где $b > 0$, выполняется соотношение $(n + a)^b = \Theta(n^b)$.
- 3) Поясните, почему утверждение «время работы алгоритма A равно как минимум $O(n^2)$ » лишено смысла.
- 4) Справедливы ли соотношения $2^{n+1} = O(2^n)$ и $2^{2n} = O(2^n)$?
- 5) Докажите, что для любых двух функций $f(n)$ и $g(n)$ мы имеем $f(n) = \Theta(g(n))$ тогда и только тогда, когда $f(n) = O(g(n))$ и $f(n) = \Omega(g(n))$.
- 6) Докажите, что время работы алгоритма равно $\Theta(g(n))$ тогда и только тогда, когда его время работы в наихудшем случае равно $O(g(n))$, а в наилучшем – $\Omega(g(n))$.

7) Докажите, что множество $o(g(n)) \cap \omega(g(n))$ является пустым.

Стандартные обозначения и часто встречающиеся функции.

- 1) Покажите, что если функции $f(n)$ и $g(n)$ монотонно неубывающие, то таковыми же являются и функции $f(n) + g(n)$ и $f(g(n))$, а если вдобавок $f(n)$ и $g(n)$ неотрицательны, то монотонно неубывающей является и функция $f(n) \cdot g(n)$.
- 2) Докажите уравнение $a^{\log bc} = c^{\log ba}$.
- 3) Докажите уравнение $\lg(n!) = \Theta(n \lg n)$. Докажите также, что $n! = \omega(2^n)$ и $n! = o(n^n)$
- 4) Докажите, что из $k \ln k = \Theta(n)$ вытекает $k = \Theta(n / \ln n)$

Контролируемые компетенции: ПК-10

Оценка компетенций осуществляется в соответствии с таблицей 4.

Контрольная работа

Задача 1

Ниже приведена таблица, строки которой соответствуют различным функциям $f(n)$, а столбцы – значениям времени t . Заполните таблицу максимальными значениями n , для которых задача может быть решена за время t , если предполагается, что время работы алгоритма, необходимое для решения задачи, равно $f(n)$ микросекунд.

	Секунда	Минута	Час	День	Месяц	Год	Век
$\log n$							
$\sqrt{n}$							
n							
$n \log n$							
n^2							
n^3							
2^n							
$n!$							

Задача 2 Рассмотрите сортировку n элементов

массива A , которая выполняется следующим образом. Сначала определяется наименьший элемент массива A , который ставится на место элемента $A[1]$. Затем производится поиск второго наименьшего элемента массива A , который ставится на место элемента $A[2]$. Этот процесс продолжится для первых $n-1$ элементов массива A . Запишите псевдокод этого алгоритма, известного как *сортировка выбором (selection sort)*. Какой инвариант цикла сохраняется для этого алгоритма? Почему его достаточно выполнить для первых $n-1$ элементов, а не для всех n элементов? Определите время работы алгоритма в наилучшем и в наихудшем случаях и запишите его в Θ обозначениях.

Задача 3 Рассмотрите сортировку n элементов массива A , которая называется *сортировка вставкой* (*insertion sort*). Она напоминает способ к которому прибегают игроки для сортировки имеющихся на руках карт. Сначала в левой руке нет ни одной карты и все они лежат на столе рубашкой вверх. Далее со стола берется по одной карте, каждая из которых помещается в нужное место среди карт, которые находятся в левой руке. Чтобы определить, куда поместить очередную карту, ее масть и достоинство сравниваются с мастью и достоинством карт в руке. После каждого шага карты в левой руке будут отсортированы. Пусть сравнение проводится в направлении слева направо. Запишите псевдокод алгоритма сортировки вставкой. Какой инвариант цикла сохраняется для этого алгоритма? Определите время работы алгоритма в наилучшем и в наихудшем случаях и запишите его в Θ обозначениях.

Задача 4

Сортировка вставкой малых массивов в процессе сортировки слиянием

Несмотря на то, что с увеличением количества сортируемых элементов время сортировки методом слияний в наихудшем случае растет как $\Theta(n \lg n)$, а время сортировки вставкой – как $\Theta(n^2)$, благодаря постоянным множителям на практике для малых размеров задач на большинстве машин сортировка вставкой выполняется быстрее. Таким образом, есть смысл использовать сортировку вставок в процессе сортировке методом слияний, когда подзадачи становятся достаточно маленькими. Рассмотрите модификацию алгоритма сортировки слиянием, в котором n/k подмассивов длиной k сортируются вставкой, после чего они объединяются с помощью обычного механизма слияния. Величина k должна быть найдена в процессе решения задачи.

- a. Покажите, что сортировка вставкой позволяет отсортировать n/k подпоследовательностей длиной k каждая за время $\Theta(nk)$ в худшем случае.
- b. Покажите, как выполнить слияние этих подпоследовательностей за время $\Theta(n \lg(n/k))$ в наихудшем случае.
- c. Если такой модифицированный алгоритм выполняется за время $\Theta(nk + n \lg(n/k))$ в наихудшем случае, то чему равно наибольшее значение k как функции от n , для которого модифицированный алгоритм в Θ -обозначениях имеет то же время работы, что и стандартная сортировка слиянием?
- d. Как следует выбирать k на практике?

Задача 5

Корректность пузырьковой сортировки

Пузырьковая сортировка представляет собой популярный, но не эффективный алгоритм сортировки. В его основе лежит многократная перестановка соседних элементов, нарушающих порядок сортировки.

Bubblesort(A)

```

1          for i = 1 to A.length - 1
2          for j = A.length downto i + 1
3          If A[j] < A[j - 1]
4          Поменяем A[j] и A[j - 1] местами

```

а. Пусть $\bar{A}$ обозначает выход процедуры Bubblesort(A). Для доказательства корректности процедуры Bubblesort необходимо доказать, что она завершается и что

$$\bar{A}[1] \leq \bar{A}[2] \leq \dots \leq \bar{A}[n]$$

где $n = A.length$. Что ещё необходимо доказать для того, чтобы показать, что процедура Bubblesort действительно выполняет сортировку?

В следующих двух частях доказываются неравенства (2.3).

- б. Точно сформулируйте инвариант цикла for в строках 2-4 и докажите, что он выполняется. Доказательство должно иметь ту же структуру доказательства инварианта цикла, которая ранее использовалась в аналогичных доказательствах в данной главе.
- с. С помощью условия завершения инварианта цикла, доказанного в части (б), сформулируйте инвариант цикла for в строках 1-4, который позволил бы доказать неравенства (2.3). Доказательство должно иметь ту же структуру доказательства инварианта цикла, которая использовалась ранее в аналогичных доказательствах в данной главе.
- д. Определите время пузырьковой сортировки в наихудшем случае и сравните его со временем сортировки вставкой.

Задача 6

Корректность правила Горнера

Следующий фрагмент кода реализует правило Горнера для вычисления полинома

$$p(x) = \sum_{i=0}^n a_i x^i = a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + x a_n) \dots))$$

для заданных коэффициентов $a_0, a_1, a_2, \dots, a_n$ и значения x .

```

1          y = 0
2          for I = n downto 0
3          y = a_i + x * y

```

- а. Чему равно время работы этого фрагмента кода правила Горнера в Θ -обозначениях?

б. Напишите псевдокод, реализующий алгоритм обычного вычисления полинома, когда каждое слагаемое полинома вычисляется отдельно. Определите асимптотическое время работы этого алгоритма и сравните его со временем работы алгоритма, основанного на правиле Горнера. с. Рассмотрим следующий инвариант цикла.

В начале каждой итерации цикла for в строках 2 и 3

$$y = \sum_{k=0}^{n-(i+1)} a_{k+i+1}x^k$$

Рассмотрим сумму без членов как равную нулю. Следуя структуре доказательства инварианта цикла, которая использовалась ранее в данной главе, воспользуйтесь указанным инвариантом цикла, чтобы показать, что по завершении работы $y = \sum_{k=0}^n a_{k+1}x^k$.

д. Сделайте заключение, что в приведенном фрагменте кода правильно вычисляется значение полинома, который задается коэффициентами $a_0, a_1, a_2, \dots, a_n$.

Задача 7

Инверсии

Пусть $A[1 \dots n]$ представляет собой массив из n различных чисел. Если $i < j$ и $A[i] > A[j]$, то пара (i, j) называется инверсией A .

- Перечислите пять инверсий массива $\langle 2, 3, 8, 6, 1 \rangle$.
- Какой массив из элементов множества $\{1, 2, \dots, n\}$ содержит максимальное количество инверсий? Сколько инверсий в этом массиве?
- Какая существует взаимосвязь между временем сортировки методом вставок и количеством инверсий во входном массиве? Обоснуйте свой ответ.
- Разработайте алгоритм, определяющий количество инверсий, содержащихся в произвольной перестановке n элементов, время работы которого в наихудшем случае равно $\Theta(n \lg n)$. (Указание: модифицируйте алгоритм сортировки слиянием.)

Контролируемые компетенции: ПК-10

Оценка компетенций осуществляется в соответствии с таблицей 4.

Вопросы к зачету

- Что такое алгоритмы?
- Алгоритмы как технология.
- Алгоритм сортировки вставкой.
- Анализ алгоритма сортировки вставкой.

5. Задача сортировки выбором.
6. Анализ алгоритма сортировки выбором.
7. Разработка алгоритма сортировки слиянием.
8. Анализ алгоритма сортировки слиянием.
9. Асимптотические обозначения.
10. Сравнение функций.
11. Задача поиска максимального подмассива.
12. Метод подстановки решения рекуррентных соотношений.
13. Анализ алгоритма поиска максимального подмассива.
14. Алгоритм Штрассена для умножения матриц.
15. Задача о найме.
16. Анализ наихудшего случая в задаче о найме.
17. Лемма о математическом ожидании индикаторной случайной величины.
18. Лемма о математическом ожидании количества наймов.
19. Анализ задачи о найме с помощью индикаторных случайных величин.
20. Задачи о гардеробщике и инверсии массива.
21. Массивы после случайной перестановки. Лемма о равномерном распределении.
22. Вероятностный подсчет.
23. Поиск в неотсортированном массиве.

Контролируемые компетенции: ПК-10

Оценка компетенций осуществляется в соответствии с таблицей 4.