

Документ подписан простой электронной подписью
Информация о владельце:

ФИО: Богдалова Елена Вячеславовна

Должность: Проректор по образовательной деятельности

Дата подписания: 13.08.2025 14:01:37

Уникальный программный ключ:

ec85dd5a839619d48ea76b2d23dba88a9c82091a

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.О.26 Теория алгоритмов

наименование дисциплины

44.03.01 «Педагогическое образование»
шифр и наименование направления подготовки

Информатика
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
Ук 1	<p>Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач</p> <p>УК-1.1. Знает: методы критического анализа и оценки современных научных достижений; основные принципы критического анализа.</p> <p>УК-1.2. Умеет: получать новые знания на основе анализа, синтеза и других методов; собирать данные по сложным научным проблемам, относящимся к профессиональной области; осуществлять поиск информации и решений на основе экспериментальных действий.</p> <p>УК-1.3. Владеет: исследованием проблем профессиональной деятельности с применением анализа, синтеза и других методов интеллектуальной деятельности; выявлением научных проблем и использованием адекватных методов для их решения; демонстрированием оценочных суждений в решении проблемных профессиональных ситуаций.</p>
Опк 8	Способен осуществлять педагогическую деятельность на основе

	<p>специальных научных знаний</p> <p>ОПК-8.1. Знает: историю, теорию, закономерности и принципы построения и функционирования образовательного процесса, роль и место образования в жизни человека и общества в области гуманитарных знаний; историю, теорию, закономерности и принципы построения и функционирования образовательного процесса, роль и место образования в жизни человека и общества в области естественно- научных знаний; историю, теорию, закономерности и принципы построения и функционирования образовательного процесса, роль и место образования в жизни человека и общества в области нравственного воспитания.</p> <p>ОПК-8.2. Умеет: использовать современные, в том числе интерактивные, формы и методы воспитательной работы в урочной и внеурочной деятельности, дополнительном образовании детей.</p> <p>ОПК-8.3. Владеет: методами, формами и средствами обучения, в том числе выходящими за рамки учебных занятий, для осуществления проектной деятельности обучающихся, проведения лабораторных экспериментов, экскурсионной работы, полевой практики и т.п.; действиями организации различных видов внеурочной деятельности: игровой, учебно-исследовательской, художественно-продуктивной, культурно-досуговой с учетом возможностей образовательной организации, места жительства и историко-культурного своеобразия региона.</p>
--	---

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.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ¹

Таблица 2

№	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	Тест	Средство, позволяющее оценить уровень знаний обучающегося путем выбора им одного из нескольких вариантов ответов на поставленный вопрос. Возможно использование тестовых вопросов, предусматривающих ввод обучающимся короткого и однозначного ответа на поставленный вопрос.	Тестовые задания
3	Контрольная работа	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу	Комплект контрольных заданий по вариантам

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

4	экзамен	Вопросы к экзамену
---	---------	--------------------

Приведенный перечень оценочных средств при необходимости может быть дополнен.

3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Оценивание результатов обучения по дисциплине дискретная 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. 3.

Таблица 3.

Код компетенции	Уровень освоения компетенции	Индикаторы достижения компетенции	Критерии оценивания результатов обучения
УК-1	Знает		
	Недостаточный уровень Оценка «незачтено», «неудовлетворительно»	УК-1.1.	<i>Не знает значительной части материала курса, не способен самостоятельно выделять главные положения в изученном материале дисциплины</i>
	Базовый уровень Оценка, «зачтено», «удовлетворительно»	УК-1.1	<i>Знает не менее 50 % основного материала курса, однако испытывает затруднения в его применении</i>

Средний уровень Оценка «зачтено», «хорошо»	УК-1.1	<i>Знает основную часть материала курса, способен применить изученный материал на практике, испытывает незначительные затруднения в решении задач</i>
Высокий уровень Оценка «зачтено», «отлично»	УК-1.1	<i>Показывает глубокое знание и понимание материала, способен применить изученный материал на практике</i>
	Умеет	
Базовый уровень	УК-1.2	<i>Умеет воспроизвести не менее 50 % основного материала курса, однако испытывает затруднения при решении практических задач</i>
Средний уровень	УК-1.2	<i>Умеет решать стандартные профессиональные задачи с применением полученных знаний, испытывает незначительные затруднения в решении задач</i>
Высокий уровень	УК-1.2	<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	УК-1.3	<i>Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности, усвоил основное содержание материала дисциплины, но имеет пробелы в усвоении материала. Имеет несистематизированные знания основных разделов дисциплины.</i>

	Средний уровень	УК-1.3	<i>Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности, способен самостоятельно выделять главные положения в изученном материале. Испытывает незначительные затруднения в решении задач.</i>
	Высокий уровень	УК-1.3	<i>Свободно владеет навыками теоретического и экспериментального исследования, показывает глубокое знание и понимание изученного материала</i>
<i>ОПК-8</i>		Знает	
	Недостаточный уровень Оценка «незачтено», «неудовлетворительно»	ОПК-8.1	<i>Не знает значительной части материала курса, не способен самостоятельно выделять главные положения в изученном материале дисциплины</i>
	Базовый уровень Оценка, «зачтено», «удовлетворительно»	ОПК-8.1	<i>Знает не менее 50 % основного материала курса, однако испытывает затруднения в его применении</i>
	Средний уровень Оценка «зачтено», «хорошо»	ОПК-8.1	<i>Знает основную часть материала курса, способен применить изученный материал на практике, испытывает незначительные затруднения в решении задач</i>
	Высокий уровень Оценка «зачтено», «отлично»	ОПК-8.1	<i>Показывает глубокое знание и понимание материала, способен применить изученный материал на практике</i>
		Умеет	

	Базовый уровень	<i>ОПК-8.2</i>	<i>Умеет воспроизвести не менее 50 % основного материала курса, однако испытывает затруднения при решении практических задач</i>
	Средний уровень	<i>ОПК-8.2</i>	<i>Умеет решать стандартные профессиональные задачи с применением полученных знаний, испытывает незначительные затруднения в решении задач</i>
	Высокий уровень	<i>ОПК-8.2</i>	<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	<i>ОПК-8.3</i>	<i>Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности, усвоил основное содержание материала дисциплины, но имеет пробелы в усвоении материала. Имеет несистематизированные знания основных разделов дисциплины.</i>
	Средний уровень	<i>ОПК-8.3</i>	<i>Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности, способен самостоятельно выделять главные положения в изученном материале. Испытывает незначительные затруднения в решении задач.</i>
	Высокий уровень	<i>ОПК-8.3</i>	<i>Свободно владеет навыками теоретического и экспериментального исследования, показывает глубокое знание и понимание изученного материала</i>

4. Методические материалы, определяющие процедуры оценивания результатов обучения

Устный опрос

Тест

Контрольная работа

экзамен

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 $A = <31,41,59,26,41,58>$.

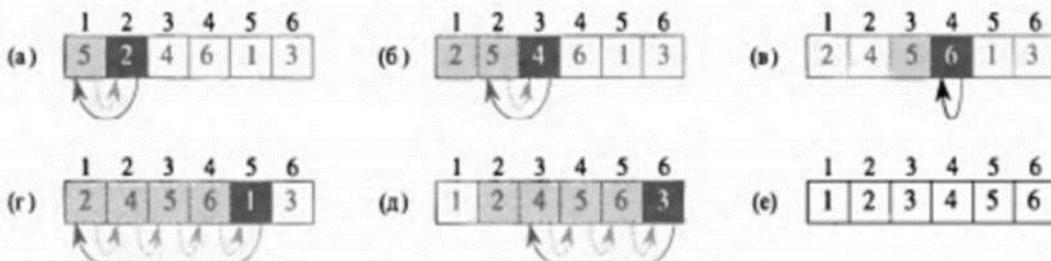


Рис. 2.2. Операции процедуры `INSERTION-SORT` над массивом $A = \{5, 2, 4, 6, 1, 3\}$. Элементы массива обозначены квадратиками, над которыми находятся индексы, а внутри — значения соответствующих элементов. Части (а)–(д) этого рисунка соответствуют итерациям цикла `for` в строках 1–8 псевдокода. В каждой 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 $A = <3,41,52,26,38,57,9,49>$.

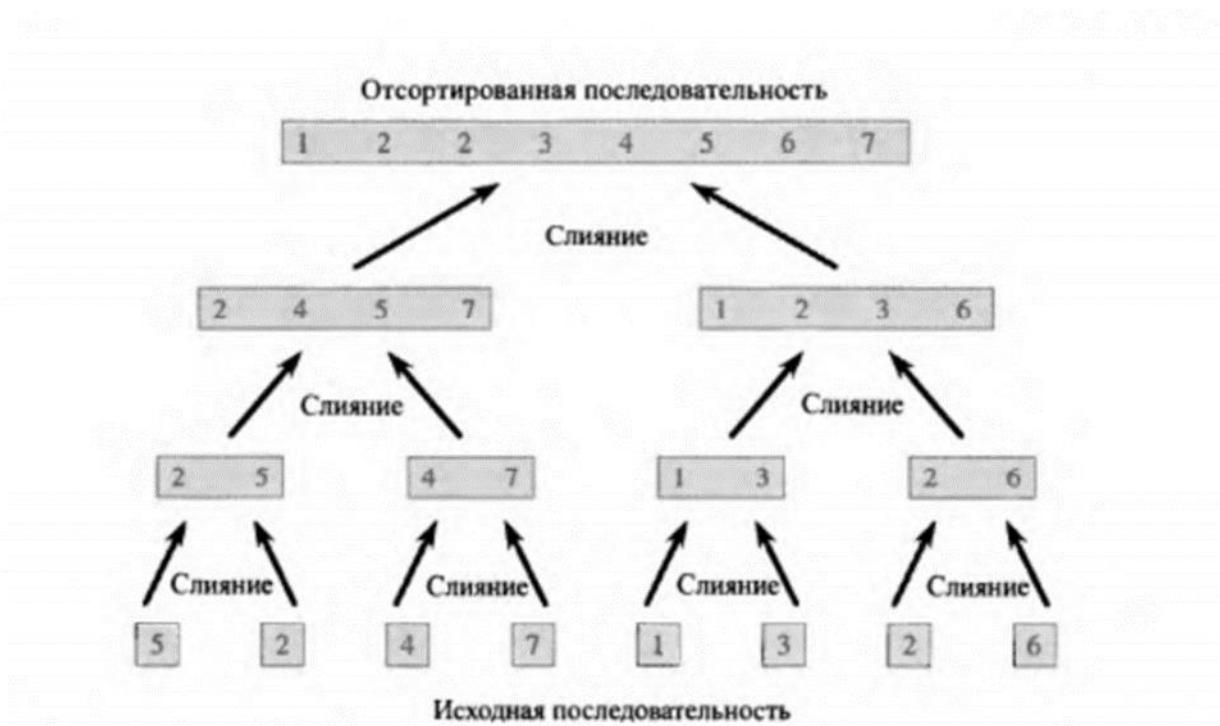


Рис. 2.4. Процесс сортировки слиянием массива $A = \{5, 2, 4, 7, 1, 3, 2, 6\}$. Длины подлежащих слиянию отсортированных последовательностей возрастают в ходе работы алгоритма.

- 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
$$T(n) = \begin{cases} 2, & \text{если } n = 2 \\ 2T(n/2) + n, & \text{если } n = 2^k, \quad k > 1 \end{cases}$$
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 А 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 в добавок $f(n)$ и $g(n)$ неотрицательны, то монотонно неубывающей является и функция $f(n) \cdot g(n)$.
- 2) Докажите уравнение $a^{\log_{bc} c} = c^{\log_{ba} a}$.
- 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							
$2n$							
$n!$							

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 сортировки 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

```

- a. Пусть A обозначает выход процедуры Bubblesort(A). Для доказательства корректности процедуры Bubblesort необходимо доказать, что она завершается и что

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

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).

- b. 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.
- c. С помощью условия завершения инварианта цикла, доказанного в части (b), 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.
- d. 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

```

- a. Чему равно время работы этого фрагмента кода правила Горнера в θ -обозначениях?
- b. 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. Следующей структуре доказательства инварианта цикла, которая

$\underset{=0}{}$ использовалась ранее в данной главе, воспользуйтесь указанным инвариантом цикла, чтобы показать, что по завершении работы $y = \sum n k - a_k x^k$.

d. 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 .

- a. Перечислите пять инверсий массива $\langle 2, 3, 8, 6, 1 \rangle$.
- b. Какой массив из элементов множества $\{1, 2, \dots, n\}$ содержит максимальное количество инверсий? Сколько инверсий в этом массиве?
- c. Какая существует взаимосвязь между временем сортировки методом вставок и количеством инверсий во входном массиве? Обоснуйте свой ответ.
- d. 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.)

Вопросы к экзамену

1. Что такое алгоритмы?
2. Алгоритмы как технология.

3. Алгоритм сортировки вставкой.
4. 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.

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

--	--	--	--

