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

УТВЕРЖДАЮ

Проректор по учебно-методической работе

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ОСВОЕНИЮ УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.ДВ.05.02 Теория алгоритмов

образовательная программа направления подготовки 09.03.03 «Прикладная информатика»
шифр, наименование

Направленность (профиль)
Цифровая трансформация

Квалификация (степень) выпускника: бакалавр

Форма обучения очная

Курс 3 семестр 5

Москва 2024

Содержание

АННОТАЦИЯ

Настоящие методические рекомендации разработаны для обучающихся очной формы обучения с учетом ФГОС ВО и рабочей программы дисциплины.

Цель освоения дисциплины «Теория алгоритмов» является формирование у студентов базовой основы знаний в области разработки и анализа алгоритмов, умений доказывать корректность алгоритмов, подготовка студентов к профессиональной деятельности в сфере разработки программных продуктов.

Задачи:

- изучение принципов построения поисковых, сортирующих и вычислительных алгоритмов;
- освоение некоторых стратегий разработки алгоритмов;
- формирование умения оценивать сложность алгоритмов, выделить легко и трудноразрешимые задачи, оценить классы задач P и NP;
- овладение базовыми методами и алгоритмами проверки логического следования, проверки корректности программ, способами определения сложности вычислений и организации эффективных алгоритмов;
- проведение оценки выбора технических и программных средств для создания программных продуктов.
- использование теории алгоритмов для алгоритмизации задач в предметной области;
- использование алгоритмического подхода для решения проблем и задач, возникающих в ходе управлении и принятия решений.

Процесс освоения учебной дисциплины направлен на формирование у обучающихся следующих компетенций:
Универсальные (УК), общепрофессиональные (ОПК), профессиональные (ПК) – в соответствии с ФГОС 3++.

Код компетенции	Содержание компетенции	Индикаторы достижения компетенции
ПК-10	Способен применять системный подход и математические методы в формализации решения прикладных задач	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. 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. Владеет навыками постановки задачи; навыками работы с библиографическими источниками информации; навыками решения поставленных задач в предметной области в рамках выбранного профиля.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ЛЕКЦИЯМ

Лекция 1 по теме «Роль алгоритмов в вычислениях».

Вопросы:

1. Что такое алгоритмы?
2. Какие задачи решаются с помощью алгоритмов?
3. Алгоритмы как технология.
4. Эффективность.
5. Алгоритмы и другие технологии.

Методические рекомендации

Лекция проводится как с применением традиционных технологий (обзорная лекция), так и интерактивных технологий (проблемная лекция).

В ходе лекционных занятий студентам рекомендовано вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Дорабатывать конспект лекции рекомендовано в соответствии рабочей программой дисциплины.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Лекция 2 по теме «Основы разработки и анализа алгоритмов»
Вопросы:

1. Алгоритм сортировки вставкой.
2. Анализ алгоритма сортировки вставкой.
3. Задача сортировки выбором.
4. Анализ алгоритма сортировки выбором.
5. Разработка алгоритма сортировки слиянием.
6. Анализ алгоритма сортировки слиянием.

Методические рекомендации

Лекция проводится как с применением традиционных технологий (обзорная лекция), так и интерактивных технологий (проблемная лекция).

В ходе лекционных занятий студентам рекомендовано вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Дорабатывать конспект лекции рекомендовано в соответствии рабочей программой дисциплины.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Лекция 3 по теме «Рост функций»

Вопросы:

1. Асимптотические обозначения.
2. Функции и время работы.
3. Асимптотические обозначения в уравнениях и задачах.
4. Стандартные обозначения и часто встречающиеся функции.

Методические рекомендации

Лекция проводится как с применением традиционных технологий (обзорная лекция), так и интерактивных технологий (проблемная лекция).

В ходе лекционных занятий студентам рекомендовано вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Дорабатывать конспект лекции рекомендовано в соответствии рабочей программой дисциплины.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). —

ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Лекция 4 по теме «Разделяй и властвуй»

Вопросы:

1. Задача поиска максимального подмассива.
2. Метод подстановки решения рекуррентных соотношений.
3. Анализ алгоритма поиска максимального подмассива.
4. Алгоритм Штрассена для умножения матриц.
5. Метод подстановки решения рекуррентных соотношений.
6. Как угадать решение и избежать ошибок.
7. Замена переменных.
8. Метод деревьев рекурсии.
9. Основной метод.
10. Основная теорема о рекуррентных соотношениях.
11. Использование основного метода.

Методические рекомендации

Лекция проводится как с применением традиционных технологий (обзорная лекция), так и интерактивных технологий (проблемная лекция).

В ходе лекционных занятий студентам рекомендовано вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Дорабатывать конспект лекции рекомендовано в соответствии рабочей программой дисциплины.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018

2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Лекция 5 по теме «Вероятностный анализ и рандомизированные алгоритмы»

Вопросы:

1. Задача о найме.
2. Анализ наихудшего случая в задаче о найме.
3. Вероятностный анализ.
4. Рандомизированные алгоритмы.
5. Индикаторная случайная величина.
6. Лемма о математическом ожидании индикаторной случайной величины.
7. Лемма о математическом ожидании количества наймов.
8. Анализ задачи о найме с помощью индикаторных случайных величин.
9. Задачи о гардеробщике и инверсии массива.
10. Изменения, которые требуется внести в алгоритм найма для рандомизации. Код случайной перестановки.
11. Лемма о математическом ожидании стоимости найма с кодом случайной перестановки.
12. Массивы после случайной перестановки. Лемма о равномерном распределении.

Методические рекомендации

Лекция проводится как с применением традиционных технологий (обзорная лекция), так и интерактивных технологий (проблемная лекция).

В ходе лекционных занятий студентам рекомендовано вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Дорабатывать конспект лекции рекомендовано в соответствии рабочей программой дисциплины.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Лекция 6 по теме «Алгоритмы сортировки»

Вопросы:

1. Парадокс дней рождения.
2. Анализ с применением индикаторной случайной величины.
3. Случайное наполнение корзин пронумерованными шарами.
4. Последовательность выпадения орлов.
5. Задача о найме в оперативном режиме.
6. Вероятностный подсчет.
7. Поиск в неотсортированном массиве.

Методические рекомендации

Лекция проводится как с применением традиционных технологий (обзорная лекция), так и интерактивных технологий (проблемная лекция).

В ходе лекционных занятий студентам рекомендовано вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации.

Рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Желательно оставить в рабочих конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

Дорабатывать конспект лекции рекомендовано в соответствии рабочей программой дисциплины.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

Практическое занятие 1 по теме «Роль алгоритмов в вычислениях».

Вопросы:

1. Что такое алгоритмы?
2. Какие задачи решаются с помощью алгоритмов?
3. Алгоритмы как технология.
4. Эффективность.
5. Алгоритмы и другие технологии.

Практические задания:

1. Приведите реальные примеры задач, в которых возникает потребность в сортировке или вычислении выпуклой оболочки.
2. Какими еще параметрами, кроме скорости, можно характеризовать алгоритм на практике?
3. Выберите одну из встречавшихся вам ранее структур данных н опишите ее 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.
6. Приведите пример приложения, для которого необходимо алгоритмическое наполнение на уровне приложений, и обсудите функции этих алгоритмов.
7. Предположим, на одной и той же машине проводится сравнительный анализ реализаций двух алгоритмов сортировки, работающих вставкой и слиянием. Для сортировки n элементов вставкой необходимо 8 n^2 шагов, а для сортировки слиянием — 64 n lgn шагов. При каком значении n время сортировки вставкой превысит время сортировки слиянием?
8. При каком минимальном значении n алгоритм, время работы которого определяется формулой 100 n^2, работает быстрее, чем алгоритм, время работы которого выражается как 2n, если оба алгоритма выполняются на одной и той же машине?
9. 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
lg n							
√n							
n							
n lgn							

n^2							
n^3							
2^n							
n!							

Методические рекомендации

При подготовке к практическим занятиям студент должен придерживаться следующих рекомендаций:

- внимательно изучить основные вопросы темы и план практического занятия,
- определить место темы занятия в общем содержании, ее связь с другими темами;
- найти и проработать соответствующие разделы в рекомендованных нормативных документах, учебниках и дополнительной литературе;
- после ознакомления с теоретическим материалом ответить на вопросы по теме курса;
- продумать пути и способы решения проблемных вопросов;
- продумать развернутые ответы на предложенные вопросы темы, опираясь на лекционные материалы, расширяя и дополняя их данными из учебников, дополнительной литературы.

В ходе практического занятия необходимо выполнить практическое задание, а затем объяснить методику его решения.

Источники и литература для подготовки:

Основная литература

1. Игошин, В.И. Сборник задач по математической логике и теории алгоритмов : учеб. пособие / В.И. Игошин. — Москва : КУРС ; ИНФРА-М, 2019. — 392 с. — (Бакалавриат). - ISBN 978-5-906818-08-9 (КУРС); ISBN 978-5-16-011429-3 (ИНФРА-М, print); ISBN 978-5-16-103684-6 (ИНФРА-М, online). - Текст : электронный. - URL: https://znanium.com/catalog/product/986940

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Практическое занятие 2 по теме «Основы разработки и анализа алгоритмов»
Вопросы:

1. Алгоритм сортировки вставкой.
2. Анализ алгоритма сортировки вставкой.
3. Задача сортировки выбором.
4. Анализ алгоритма сортировки выбором.
5. Разработка алгоритма сортировки слиянием.
6. Анализ алгоритма сортировки слиянием.

Практические задания:

1. Проиллюстрируйте работу процедуры Insertion-Sort по сортировке массива $A = < 31, 41, 59, 26, 41, 58 >$.
2. Перепишите процедуру Insertion-Sort для сортировки в невозрастающем порядке вместо неубывающего.
3. Рассмотрим задачу поиска.
   Вход. Последовательность из n чисел $A = < a_1, a_2, \dots, a_n >$ и значение v.
   Выход. Индекс i, такой, что v, 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 инварианта 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.
5. Выразите функцию $n^3/1000 - 100\,n^2 - 100n + 3$ в Θ-обозначениях.
6. Рассмотрим сортировку 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 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 наилучшее и наихудшем случаях и запишите его в Θ-обозначениях.
7. При алгоритме линейного поиска 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.
8. Каким образом можно модифицировать почти каждый алгоритм, чтобы получить оптимальное время работы в наилучшем случае?
9. Проиллюстрируйте работу алгоритма сортировки слиянием для массива $A = < 3, 41, 52, 26, 38, 67, 9, 49 >$.
10. 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.

11. Воспользуйтесь методом математической индукции для доказательства того, что, когда n является точной степенью 2, решением рекуррентного соотношения

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

является $T(n) = nlgn$.

12. Сортировку вставкой можно представить в виде рекурсивной последовательности. Чтобы отсортировать массив A $[1..n]$, сначала нужно рекурсивно отсортировать массив A $[1..n-1]$, после чего в этот отсортированный массив помещается элемент A $[n]$. Запишите рекуррентное уравнение для времени работы этой рекурсивной версии сортировки вставкой.
13. Возвращаясь к задаче линейного поиска, 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 $\Theta = (\text{lgn})$.
14. Заметим, что в цикле while в строках 5—7 процедуры INSERTION-SORT для сканирования (в обратном порядке) отсортированного подмассива A $[1..\text{j}-1]$ используется линейный поиск. Можно ли использовать бинарный поиск (см. предыдущее упр.) вместо линейного, чтобы время работы этого алгоритма в наихудшем случае улучшилось и стало равным $\Theta = (nlgn)$?
15. Разработайте алгоритм со временем работы $\Theta = (lgn)$, который для заданного множества S из n целых чисел и другого целого числа x определяет, имеются ли в множестве S два элемента, сумма которые равна x.

Задачи

1. 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 $\Theta = (nlgn)$, 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 $\Theta = (nlg(\frac{n}{k})\ )$ в наихудшем случае.

c. Если такой модифицированный алгоритм выполняется за время $\Theta = (nk + nlg(\frac{n}{k})\ )$ 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?

2. 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 t = 1 to A.length — 1
2      for j = A.length downto i + 1
3          if A[j] < A[j—1]
4              поменять 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

$$A'[1] \leq A'[2] \leq \cdots \leq A'[n] \quad (2.3)$$

где $n = A.\text{length}$. Что еще 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.

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.

d) Определите время пузырьковой сортировки в наихудшем случае и сравните его со временем сортировки вставкой.

3. **Корректность правила Горнера**

Следующий фрагмент кода реализует правило Горнера для вычисления полинома

$$P(x) = \sum_{k=0}^{n} a_k x^k =$$

$$=a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + xa_n) \dots))$$

для заданных коэффициентов $a_0, a_1, \dots , a_n$.

1. $y = 0$
2. **for** *i*=*n* **downto** 0
3. $\quad y = a_i + x * y$

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.

c. Рассмотрим следующий инвариант цикла.

В начале каждой итерации цикла for в строках 2 и 3

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

Рассматривайте 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 $y = \sum_{k=0}^{n} a_k x^k$.

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

4. **Инверсии**

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

a. Перечислите пять инверсий массива <2, 3, 8, 6, 1>.

b. Какой массив из элементов множества {1, 2, ... , 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 $\Theta = (nlgn)$. (*Указание:* модифицируйте алгоритм сортировки слиянием.)

Источники и литература для подготовки:

Основная литература

1. Игошин, В.И. Сборник задач по математической логике и теории алгоритмов : учеб. пособие / В.И. Игошин. — Москва : КУРС ; ИНФРА-М, 2019. — 392 с. — (Бакалавриат). - ISBN 978-5-906818-08-9 (КУРС); ISBN 978-5-16-011429-3 (ИНФРА-М, print); ISBN 978-5-16-103684-6 (ИНФРА-М, online). - Текст : электронный. - URL: https://znanium.com/catalog/product/986940

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Практическое занятие 3 по теме «Рост функций»

Вопросы:

1. Асимптотические обозначения.
2. Функции и время работы.
3. Асимптотические обозначения в уравнениях и задачах.
4. Стандартные обозначения и часто встречающиеся функции.

Практические задания:

1. Пусть $f(n)$ и $g(n)$ — асимлтоткчески неотрицательные фунхиии. Докажите с по-мотцью быовою определения Θ-обозначениях, что $max\big(f(n), g(n)\big) = \Theta(f(n) + g(n))$.
2. Покажите, что для любых действительных констант a и b, где $b > 0$, вышолняется соотношение
3. $(n + a)^b = \Theta(n^b)$.
4. Поясните, почему утверждение «время работы алгоритма *A* равно как минимум $O(n^2)$» лишено смысла.

5. Справедливы ли соотношения $2^{n+1} = O(2^n)$ и $2^{2n} = O(2^n)$?
6. Докажите, что $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$, если $f(n) = o(g(n))$.
7. 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 — $\Omega(g(n))$.
8. Докажите, что множество $o(g(n)) \cap \omega(g(n))$ является пустым.
9. Можно обобщить наши обозначения на случай двух параметров n и m, которые могут возрастать до бесконечности по отдельности с разными скоростями. Для данной функции $g(n,m)$ обозначим как $O(g(n,m))$ множество функций $O(g(n,m)) = \{f(n,m)$: существуют положительные константы c, n_0 и m_0, такие, что $0 \le f(n,m) \le cg(n,m)$ для всех $n \ge n_0$ или $m \ge m_0\}$. Приведите соответствующие определения для $\Omega\big(g(\mathrm{n},\mathrm{m})\big)$ и $\Theta\big(g(\mathrm{n},\mathrm{m})\big)$.
10. 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 $f(n) \cdot g(n)$.
11. Докажите уравнение $a^{log_b c} = c^{log_b a}$.
12. Докажите уравнение $lg(n!) = \Theta(n * lg(n))$. Докажите также, что $n! = \omega(2^n)$ и $n! = o(n^n)$.
13. Является ли функция $\lceil lg(n) \rceil!$ поЛиномяально ограниченной? А функция $\lceil lglg(n) \rceil!$?
14. Какая из функций $lg(lg^*(n))$ и $lg^*(lg(n))$ является асимптотически большей?
15. Покажите, что золотое сечение ф и сопряженное с ним $\widehat{ф}$ удовлетворяют уравнению $x^2 = x + 1$.
16. Докажите по индукции, что i-е число Фибоначчи удовлетворяет уравнению $F_i = \frac{ф^i - \widehat{ф}^i}{\sqrt{5}}$, где ф — золотое сечение, а $\widehat{ф}$ — сопряженное с ним.
17. Покажите, что из $k \cdot ln(k) = \Theta(n)$ вытекает $k = \Theta(n/ln(n))$.

Задачи

1. Асимптотическое поведение полиномов

Пусть

$$p(n) = \sum_{i=0}^{d} a_i x^i$$

где $\mathrm{a_d} > 0$, представляет собой полином степени d от n, и пусть k является константой. Используя определения асимптотических обозначений, докажите следующие свойства.

a. Если $k \ge d$, то $p(n) = O(n^k)$.

b. Если $k \le d$, то $p(n) = \Omega(n^k)$.

c. Если $k = d$, то $p(n) = \Theta(n^k)$.

d. Если $k > d$, то $p(n) = o(n^k)$.

e. Если $k < d$, то $p(n) = \omega(n^k)$.

2. Относительный асимптотический рост

Для каждой пары приведённых в таблице выражений (A, B) укажете, каким отношением A связано с B: O, o, Ω, ш или О-. Предполагается, что $k \ge 1$, $\varepsilon > 0$ и $\mathrm{c} > 1$ — константы. Ваш ответ должен выражаться таблицей, в каждой ячейке которой указано значение “Да” или “Нет”.

	A	B	O	o	Ω	ω	Θ
а.	$lg^k n$	n^{ε}					
б.	n^k	c^n					
в.	$\sqrt{n}$	$sinn$					
г.	n^{lgc}	$2^{n/2}$					
д.	2^n	c^{lgn}					
е.	$\lg(n!)$	$lg(n^n)$					

3. Упорядочение по скорости асимптотического роста

а. Расположите приведенные ниже функции по скорости их асимптотического роста, т.е. постройте такую последовательность функций $g_1, g_2, \dots, g_{30}$, что $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \dots, g_{29} = \Omega(g_{30})$. Разбейте свой список на классы эквивалентности так, чтобы функции $f(n)$ и $g(n)$ находились в одном и том же классе тогда и только тогда, когда $f(n) \ = \Theta(g(n))$

$lg(lg * n)$	$2^{lg\, n}$	$(\sqrt{2})^{lg\, n}$	n^2	$n!$	$(\lg n)!$
$\left(\frac{3}{2}\right)^n$	n^3	$lg^2\, n$	$lg(n!)$	2^{2^n}	$n^{\frac{1}{\lg n}}$
$ln\, ln\, n$	$lg\, n$	$n \times 2^n$	$n^{lg\, lg\, n}$	$\ln n$	1
$2^{lg\, n}$	$lg(n)^{lg\, n}$	e^n	$4^{lg\, n}$	$(n+1)!$	$\sqrt{\lg n}$
$lg(lg\, n)$	$2^{\sqrt{2\, lg\, n}}$	n	2^n	$n \lg n$	$2^{2^{n+1}}$

б. Приведите пример неотрицательной функции $f(n)$, такой, что для всех фукций $g_i(n)$ из части (а) $f(n)$ не принадлежит ни множеству $O(g_i(n))$ ни множеству $\Omega(g_i(n))$

4. Свойства асимптотических обозначений

Пусть $f(n)$ и $g(n)$ — асимптотически положительные функции. Докажите или опровергните справедливость каждого из приведенных ниже утверждений.

а. Из $f(n) = O\big(g(n)\big)$ вытекает $g(n) = O\big(f(n)\big)$.
б. $f(n) + g(n) = \Theta(\min(f(n), g(n)))$.
в. Из $f(n) = O\big(g(n)\big)$ вытекает $\lg f(n) = O(\lg g(n))$, где $\lg(g(n)) \geq 1$ и $f(n) \geq 1$ для всех достаточно больших n .
г. Из $f(n) = O\big(g(n)\big)$ вытекает $2^{f(n)} = O\big(2^{g(n)}\big)$.
д. $f(n) = O\left(\big(f(n)\big)^2\right)$.
е. Из $f(n) = O\big(g(n)\big)$ вытекает $g(n) = \Omega(f(n))$.
ж. $f(n) = \Theta\left(f(\frac{n}{2})\right)$.
з. $f(n) + o(f(n)) = \Theta(f(n))$.

Методические рекомендации

При подготовке к практическим занятиям студент должен придерживаться следующих рекомендаций:

- внимательно изучить основные вопросы темы и план практического занятия,
- определить место темы занятия в общем содержании, ее связь с другими темами;

- найти и проработать соответствующие разделы в рекомендованных нормативных документах, учебниках и дополнительной литературе;
- после ознакомления с теоретическим материалом ответить на вопросы по теме курса;
- продумать пути и способы решения проблемных вопросов;
- продумать развернутые ответы на предложенные вопросы темы, опираясь на лекционные материалы, расширяя и дополняя их данными из учебников, дополнительной литературы.

В ходе практического занятия необходимо выполнить практическое задание, а затем объяснить методику его решения.

Источники и литература для подготовки:

Основная литература

1. Игошин, В.И. Сборник задач по математической логике и теории алгоритмов : учеб. пособие / В.И. Игошин. — Москва : КУРС ; ИНФРА-М, 2019. — 392 с. — (Бакалавриат). - ISBN 978-5-906818-08-9 (КУРС); ISBN 978-5-16-011429-3 (ИНФРА-М, print); ISBN 978-5-16-103684-6 (ИНФРА-М, online). - Текст : электронный. - URL: https://znanium.com/catalog/product/986940

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Практическое занятие 4 по теме «Разделяй и властвуй»

Вопросы:

1. Задача поиска максимального подмассива.
2. Метод подстановки решения рекуррентных соотношений.
3. Анализ алгоритма поиска максимального подмассива.
4. Алгоритм Штрассена для умножения матриц.
5. Метод подстановки решения рекуррентных соотношений.
6. Как угадать решение и избежать ошибок.
7. Замена переменных.
8. Метод деревьев рекурсии.
9. Основной метод.
10. Основная теорема о рекуррентных соотношениях.
11. Использование основного метода.

Практические задания:

1. Что возвращает процедура FIND-MAXIMUM-SUBARRAY, когда все элементы A отрицательны?
2. Напишите псевдокод для решения задачи поиска максимального подмассива методом грубой силы. Ваша процедура должна выполняться за время $\Theta(n^2)$.
3. Реализуйте и метод грубой силы, и рекурсивный алгоритм на своем компьютере. Каким оказывается размер задачи точки пересечения n_0, в которой рекурсивный алгоритм превосходит алгоритм грубой силы? Далее измените базовый случай рекуррентного алгоритма, применяя алгоритм грубой силы при размере задачи, не превосходящем n_0. Меняет ли это точку пересечения?
4. Предположим, что мы меняем определение задачи поиска максимального подмассива, позволяя конечному результату быть пустым массивом и полагая, что сумма значений пустого массива равна нулю. Как бы вы изменили любой алгоритм, не допускающий решения в виде пустого массива, чтобы такой результат в виде пустого подмассива стал возможным?
5. Воспользуйтесь приведенными далее идеями для разработки нерекурсивного алгоритма поиска максимального подмассива за линейное время. Начните с левого конца массива и двигайтесь вправо, отслеживая найденный к данному моменту максимальный подмассив. Зная максимальный подмассив массива $A[1\,..j]$, распространите ответ на поиск максимального подмассива, заканчивающегося индексом $j+1$, воспользовавшись следующим наблюдением: максимальный подмассив массива $A[1\,..j+1]$ представляет собой либо максимальный подмассив массива $A[1\,..j]$, либо подмассив $A[i\,..j+1]$ для некоторого $1 \leq i \leq j+1$. Определите максимальный подмассив вида A [$i\,..j+1$] за константное время, зная максимальный подмассив, заканчивающийся индексом j.
6. Воспользуйтесь алгоритмом Штрассена для вычисления произведения матриц $\begin{pmatrix} 1 & 3 \\ 7 & 5 \end{pmatrix}\begin{pmatrix} 6 & 8 \\ 4 & 2 \end{pmatrix}$. Покажите, как вы это делаете.
7. Запишите псевдокод алгоритма Штрассена.
8. Как модифицировать алгоритм Штрассена для перемножения матриц размером nxn, где п не является точной степенью 2? Покажите, что получающийся в результате алгоритм выполняется за время $o(n^{lg7})$.
9. Чему равно наибольшее k, такое, что если вы можете перемножить 3x3 - матрицы с помощью k умножений (не предполагая коммутативности умножения), то вы можете перемножить матрицы размером п x п за время $\theta(n^{lg7})$? Каким должно быть время работы такого алгоритма?
10. В. Пан (V. Pan) открыл способ перемножения матриц размером 68x68 с использованием только 132 464 умножений, способ перемножения матриц размером 70x70 с использованием 143 640 умножений и способ перемножения матриц размером 72x72 с использованием 155 424 умножений. Какой из методов дает нам лучшее асимптотическое время работы при его использовании в алгоритме "разделяй и властвуй" для перемножения матриц? Проведите сравнение с алгоритмом Штрассена.
11. Насколько быстро вы сумеете умножить матрицу размером knxn на матрицу размером nxkn, применяя алгоритм Штрассена в качестве подпрограммы? Ответьте на тот же вопрос для ситуации, когда мы меняем входные матрицы местами.
12. Покажите, как перемножить комплексные числа $a+bi$ и c $+\ di$, используя только три умножения действительных чисел. Алгоритм должен получать a, b, c и d в качестве входных данных и возвращать действительную $(ac - bd)$ и мнимую $(ad + bc)$ части произведения по отдельности.

13. Воспользуйтесь деревом рекурсии для определения точной асимптотической верхней границы рекуррентного соотношения $T(n) = 3T(\lfloor n/2 \rfloor) + n$. Для проверки своего ответа используйте метод подстановок.
14. Воспользуйтесь деревом рекурсии для определения точной асимптотической верхней границы рекуррентного соотношения $T(n) = T(n/2) + n^2$. Для проверки своего ответа используйте метод подстановок.
15. Воспользуйтесь деревом рекурсии для определения точной асимптотической верхней границы рекуррентного соотношения $T(n) = 4T(n/2 + 2) + n$. Для проверки своего ответа используйте метод подстановок.
16. Воспользуйтесь деревом рекурсии для определения точной асимптотической верхней границы рекуррентного соотношения $T(n) = 2T(n-1) + 1$. Для проверки своего ответа используйте метод подстановок.
17. Воспользуйтесь деревом рекурсии для определения точной асимптотической верхней границы рекуррентного соотношения $T(n) = T(n-1) + T(n/2) + n$. Для проверки своего ответа используйте метод подстановок.
18. Обратившись к соответствующему дереву рекурсии, докажите, что решением рекуррентного соотношения $T(n) = T(n/3) + T(2n/3) + cn$, где с представляет собой константу, является $\Omega(n \cdot lg\, n)$.
19. Постройте дерево рекурсии для рекуррентного соотношения $\mathrm{T}(n) = 4T(\lfloor n/2 \rfloor) + cn$, где с - константа, и найдите точную асимптотическую границу его решения. Проверьте её с помощью метода подстановок.
20. Найдите с помощью дерева рекурсии точную асимптотическую оценку решения рекуррентного соотношения $T(n) = T(n-a) + T(a) + cn$, где $a \geq 1$ и $c > 0$ являются константами.
21. Найдите с помощью дерева рекурсии точную асимптотическую оценку решения рекуррентного соотношения $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$, где α - константа из диапазона $0 < \alpha < 1$; константой является и с > 0.
22. С помощью основной теоремы найдите точные асимптотические границы следующих рекуррентных соотношений.

   а. $T(n) = 2T(n/4) + 1$.

   б. $T(n) = 2T(n/4) + \sqrt{n}$.

   в. $T(n) = 2T(n/4) + n$.

   г. $T(n) = 2T\ (n/4) + n^2$.

23. Профессор хочет разработать алгоритм матричного умножения, асимптотически более быстрый, чем алгоритм Штрассена. Его алгоритм будет использовать метод "разделяй и властвуй", разбивая каждую матрицу на части размером $n/4 \times n/4$, причем шаги разделения и комбинирования выполняются за время (n^2). Профессору требуется определить, сколько подзадач должен создавать его алгоритм. чтобы опередить алгоритм Штрассена. Если алгоритм профессора создает a подзадач, то рекуррентное соотношение для времени работы $T(n)$ принимает вид $T(n) = aT(n/4) + \Theta(n^2)$. Каково наибольшее целочисленное

значение a, для которого алгоритм профессора оказывается асимптотически быстрее алгоритма Штрассена?

24. Покажите с помощью основного метода, что $T(n) = \Theta(lg\, n)$ является решением рекуррентного соотношения $T(n) = (n/2) + \Theta(1)$, возникающего в ходе анализа алгоритма бинарного поиска. (Алгоритм бинарного поиска описан в упр. 2.3.5.)
25. Можно ли применить основной метод к рекуррентному соотношению $T(n) = 4T(n/2) + n^2 lg\, n$? Обоснуйте свой ответ. Найдите асимптотическую верхнюю границу решения этого рекуррентного соотношения.
26. Приведите простое и точное выражение для n_j в уравнении (4.27) для случая, когда b - положительное целое число (а не произвольное действительное).
27. Покажите, что если выполняется соотношение $f(n) = \Theta(n^{log_b a}\, lg^k\, n)$, где $k \geq 0$, то основное рекуррентное соотношение имеет решение $\mathrm{T}(n) = \Theta(n^{log_b a}\, lg^{k+1}\, n)$. Для простоты рассмотрите только случай точных степеней b.
28. Покажите, что в случае 3 основной теоремы одно из условий излишнее в том смысле, что из условия регулярности $af(n/b) \leq cf(n)$ для некоторой константы с < 1 следует, что существует константа ε > 0, такая, что $f(n) = \Omega(n^{log_b a+\varepsilon})$.

Задачи

1. **Примеры рекуррентных соотношений**

Определите верхнюю и нижнюю асимптотические границы функции $T(n)$ для каждого из представленных ниже рекуррентных соотношений. Считаем, что $T(n)$ при п 5 2 является константой. Попытайтесь сделать эту оценку как можно более точной и обоснуйте свой ответ.

$a.T(n) = 2T(n/2) + n^4$.

б. $T(n) = T(7n/10) + n$.

в. $T(n)\ 16T(n/4) + n^2$.

г. $T(n) = 7T(n/3) + n^2$.

д. $T(n) = 7T(n/2) + n^2$.

е. $T(n) = 2T(n/4) + \sqrt{n}$.

ж. $\mathrm{T}(n) = T(n - 2) + n^2$

2. **Стоимости передачи параметров**

Будем предполагать, что передача параметров при вызове процедуры занимает фиксированное время, даже если передается N-элементный массив. Для большинства вычислительных систем это предположение справедливо, поскольку передается не сам массив, а указатель на него. В данной задаче исследуются три стратегии передачи параметров.

1. Массив передается посредством указателя. Время равно $\Theta(1)$.

2. Массив передается посредством копирования. Время равно $\Theta(N)$, где N - размер массива.

3. Массив передается путем копирования только некоторого поддиапазона, к которому обращается вызываемая процедура. Время равно $\Theta\ (q - \mathrm{p} + 1)$ при передаче подмассива $\mathrm{A}[p..q]$

а. Рассмотрите рекурсивный алгоритм бинарного поиска, предназначенный для нахождения числа в отсортированном массиве (см. упр. 2.3.5). Приведите рекуррентные соотношения, описывающие время бинарного поиска в наихудшем случае, если массивы передаются с помощью каждого из описанных выше методов, и дайте точные верхние границы решений этих рекуррентных соотношений. Пусть размер исходной задачи равен N, а размер подзадачи - n.

б. Выполните задание части (а) для алгоритма MERGE-SORT из раздела 2.3.1.

3. **Другие примеры рекуррентных соотношений**

Дайте верхнюю и нижнюю асимптотические оценки функции $T(n)$ в каждом из приведенных ниже рекуррентных соотношений. Предполагается, что $\mathrm{T}(n)$ для достаточно малых значений п является постоянной величиной. Постарайтесь, чтобы оценки были как можно более точными и обоснуйте ответы.

а. $T(n) = 4T(n/3) + n\,lg\,n.$

б. $T(n) = 3T(n/3) + n/\,lg\,n.$

в. $T(n) = 4T(n/2) + n^2\sqrt{n}.$

г $T(n) = 3T\ (n/3 - 2) + n/2.$

д. $T(n) = 2T(n/2) + n/lg\,n.$

е. $T(n) = T(n/2) + T(n/4) + T(n/8) + n.$

ж. $T(n) = T(n-1) + 1/n$

з. $T(n) = T(n-1) + \lg\ n\ .$

и. $\mathrm{T}(n) = T(n-2) + 1/lg\,n.$

к. $T(n) = \sqrt{n}T(\sqrt{n}) + n\ .$

4. **Числа Фибоначчи**

В этой задаче раскрываются свойства чисел Фибоначчи, определенных с помощью рекуррентного соотношения (3.22). Воспользуемся для решения рекуррентного соотношения Фибоначчи методом производящих функций. Определим **производящую функцию** (или **формальный степенной ряд**) $\boldsymbol{F}$ как

$$F(z) = \sum_{i=0}^{\infty} F_i \, z^i$$

$$= 0 + z + z^2 + 2z^3 + 3z^4 + 5z^5 + 8z^6 + 13z^7 + 12z^8 \ldots$$

где Fi, представляет собой i-е число Фибоначчи.

а. Покажите, что $F(z) \; = \; 2 \; + \; zF(z) \; + \; z^2F(z)$,

б. Покажите, что

$$F(z) \; = \frac{z}{1 - z - z^2}$$

$$= \frac{z}{(1 - \emptyset)(1 - \widehat{\emptyset})}$$

$$= \frac{1}{\sqrt{5}} \left(\frac{1}{1 - \emptyset z} - \frac{1}{1 - \widehat{\emptyset} z} \right)$$

Где

$$\emptyset \; = \frac{1 + \sqrt{5}}{2} = 1.61803 \ldots$$

и

$$\widehat{\emptyset} \; = \frac{1 - \sqrt{5}}{2} = 0.61803 \ldots$$

в. Покажите, что

$$F(z) = \sum_{i=0}^{\infty} \frac{1}{\sqrt{5}} (\emptyset^i - \widehat{\emptyset}^i) z^i$$

г. Воспользуйтесь п. (в) для доказательства того, что $F_i = \frac{\emptyset^i}{\sqrt{5}}$ (округленное до ближайшего целого) для всех $i \; > \; 0$. (Указание: обратите внимание, что $|\emptyset\rceil < 1$.)

5. **Тестирование микросхем**

В распоряжении профессора есть n предположительно идентичных микросхем ("чипов"), которые в принципе способны тестировать друг друга. В тестирующее приспособление за один раз можно поместить две микросхемы. При этом каждая микросхема тестирует соседнюю и выдает отчет о результатах тестирования. Исправная микросхема всегда выдает правильные результаты тестирования другой микросхемы, а результатам неисправной микросхемы доверять нельзя. Таким образом, возможны четыре варианта результатов тестирования, приведенные в таблице ниже.

Отчет А	Отчёт В	Заключение
В исправна	А исправна	Либо обе исправны, либо

		обе неисправны
В исправна	А неисправна	Неисправна как минимум одна микросхема
В неисправна	А исправна	Неисправна как минимум одна микросхема
В неисправна	А неисправна	Неисправна как минимум одна микросхема

а. Покажите, что, если как минимум $n/2$ микросхем неисправны, профессор не сможет точно определить исправные микросхемы, какой бы стратегией попарных испытаний он не пользовался. (Предполагается, что неисправные микросхемы не договариваются между собой, чтобы обмануть профессора.)

б. Рассмотрим задачу о поиске одной исправной микросхемы среди n микросхем, если предполагается, что исправно более половины всех микросхем. Покажите, что $[n/2]$ попарных тестирований достаточно для сведения этой задачи к подзадаче, размер которой приблизительно в два раза меньше.

в. Покажите, что можно найти все исправные микросхемы с помощью $\Theta(n)$ попарных тестирований в предположении, что исправны более половины микросхем. Сформулируйте и решите рекуррентное соотношение, описывающее количество тестирований.

6. **Массивы Монжа**

Массив А размером $m \times n$, состоящий из действительных чисел, является массивом Монжа (Monge array), если для всех i, j, k и l, таких, что $1 \le i < k \le m$ и $1 \le j < l \le n$, мы имеем

$$A[i,j] + A[k,l] \le A[i,l] + A[k,j]$$

Другими словами, при любом выборе двух строк и двух столбцов из массива Монжа (при этом элементы массива на пересечении выбранных строк и столбцов образуют прямоугольник) сумма элементов в левом верхнем и правом нижнем углах полученного прямоугольника не превышает сумму элементов в левом нижнем и правом верхнем его углах. Приведем пример массива Монжа.

10	17	13	28	23
17	22	16	29	23
24	28	22	34	24
11	13	6	17	7
45	44	32	37	23
36	33	19	21	6
75	66	51	53	34

а. Докажите, что массив является массивом Монжа тогда и только тогда, когда для всех $i = 1,2,\dots,m-1$ и $j = 1,2,\dots.n-1$ выполняется условие

$$A[i,j] + A[i+1,j+1] \le A[i,j+1] + A[i+1,j].$$

(Указание: для прямого доказательства воспользуйтесь методом математической индукции отдельно для строк и для столбцов.)

б. Приведенный ниже массив не является массивом Монжа. Измените в нем один элемент таким образом, чтобы он стал массивом Монжа. (Указание: воспользуйтесь п. (а).)

$$\begin{matrix} 37 & 23 & 22 & 32 \\ 21 & 6 & 7 & 10 \\ 53 & 34 & 30 & 31 \\ 32 & 13 & 9 & 6 \\ 43 & 21 & 15 & 8 \end{matrix}$$

в. Пусть $f(i)$ представляет собой индекс столбца, содержащего крайний слева минимальный элемент в строке i. Докажите, что $f(1) \leq f(2) \leq \ldots f(m)$ для любого массива Монжа размером $m \times n$

г. Далее приведено описание алгоритма "разделяй и властвуй", предназначенного для вычисления крайнего слева минимального элемента в каждой строке, входящей в состав массива Монжа А размером $m \times n$

Постройте подматрицу A' матрицы А. состоящую из четных строк матрицы А.

С помощью рекурсивной процедуры найдите в каждой строке матрицы А' крайний слева минимальный элемент. Затем найдите крайний слева минимальный элемент среди нечетных строк матрицы А.

Объясните, как найти крайний слева минимальный элемент среди нечетных

строк матрицы А (предполагается, что крайний слева минимальный элемент среди четных столбцов этой матрицы известен) за время $O(m+n)$.

д. Запишите рекуррентное соотношение, описывающее время работы алгоритма из п. (г). Покажите, что его решение имеет вид $O(m + n \log m)$.

Методические рекомендации

При подготовке к практическим занятиям студент должен придерживаться следующих рекомендаций:

- внимательно изучить основные вопросы темы и план практического занятия,
- определить место темы занятия в общем содержании, ее связь с другими темами;
- найти и проработать соответствующие разделы в рекомендованных нормативных документах, учебниках и дополнительной литературе;
- после ознакомления с теоретическим материалом ответить на вопросы по теме курса;
- продумать пути и способы решения проблемных вопросов;
- продумать развернутые ответы на предложенные вопросы темы, опираясь на лекционные материалы, расширяя и дополняя их данными из учебников, дополнительной литературы.

В ходе практического занятия необходимо выполнить практическое задание, а затем объяснить методику его решения.

Источники и литература для подготовки:

Основная литература

1. Игошин, В.И. Сборник задач по математической логике и теории алгоритмов : учеб. пособие / В.И. Игошин. — Москва : КУРС ; ИНФРА-М, 2019. — 392 с. — (Бакалавриат). - ISBN 978-5-906818-08-9 (КУРС); ISBN 978-5-16-011429-3 (ИНФРА-М, print); ISBN 978-5-16-103684-6 (ИНФРА-М, online). - Текст : электронный. - URL: https://znanium.com/catalog/product/986940

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Практическое занятие 5 по теме «Вероятностный анализ и рандомизированные алгоритмы»

Вопросы:

1. Задача о найме.
2. Анализ наихудшего случая в задаче о найме.
3. Вероятностный анализ.
4. Рандомизированные алгоритмы.
5. Индикаторная случайная величина.
6. Лемма о математическом ожидании индикаторной случайной величины.
7. Лемма о математическом ожидании количества наймов.
8. Анализ задачи о найме с помощью индикаторных случайных величин.
9. Задачи о гардеробщике и инверсии массива.
10. Изменения, которые требуется внести в алгоритм найма для рандомизации. Код случайной перестановки.
11. Лемма о математическом ожидании стоимости найма с кодом случайной перестановки.
12. Массивы после случайной перестановки. Лемма о равномерном распределении.

Практические задания:

1. Покажите, что из предположения о том, что в строке 4 процедуры Hire-Assistant всегда можно определить, какой кандидат наилучший, следует, что мы знаем общий порядок рангов кандидатов.
2. Опишите реализацию процедуры $Random(a, b)$, которая может использовать только один вызов — процедуры $Random(0,1)$. Чему равно математическое ожидание времени работы вашей реализации как функции от a и b?
3. Предположим, что нужно выводить 0 и 1 с вероятностью 1/2. В нашем распоряжении имеется процедура Biased-Random, которая с вероятностью p выдает 0 и с вероятностью 1 — р — число 1; значение р нам неизвестно. Сформулируйте алгоритм, использующий в качестве подпрограммы процедуру Biased-Random +и возвращающий равномерно распределенные числа 0 н 1, те. вероятность вывода каждого из них равна 1/2. Чему равно математическое ожидание времени работы такой процедуры как функции от р?
4. Чему равна вероятность того, что в процедуре Hire-Assistant будет нанят ровно один кандидат, если предполагается, что собеседование с кандидатами проводится в случайном порядке? А вероятность того, что будут наняты п кандидатов?
5. Чему равна вероятность того, что в процедуре Hire-Assistant будет нанято ровно два кандидата, если предполагается, что собеседование с кандидатами проводится в случайном порядке?
6. Вычислите математическое ожидание суммы очков на n игральных костях с помощью индикаторных случайных величин
7. Решите с помощью индикаторных случайных величин задачу, известную как **задача о гардеробщике**. Каждый из п посетителей ресторана сдает свою шляпу в гардероб. Гардеробщик возвращает шляпы случайным образом. Чему равно математическое ожидание количества посетителей, получивших обратно свои собственные шляпы?
8. Пусть $A[1 \dots n]$ — массив, состоящий из n различных чисел. Бели $i < j$, а $A[i] > A[j]$, то пара (i, j) называется ***инверсией*** массива А (более детальную информацию об инверсиях можно найти в задаче 2.4), Предположим, что элементы массива А образуют равномерную случайную перестановку чисел (1,2,...,n). Воспользуйтесь индикаторными случайными величинами для поиска математического ожидания количества инверсий в массиве.
9. У профессора возникли возражения против инварианта цикла, использующегося при доказательстве леммы 5.5. Он сомневается, что этот инвариант выполняется перед первой итерацией. Согласно его доводам, пустой подмассив не содержит никаких размещений из 0 элементов, поэтому вероятность того, что в таком подмассиве находится то или иное размещение, должна быть равна 0. Из этих рассуждений следует, что инвариант цикла перед первой итерацией не выполняется. Перепишите процедуру Randomize In Place таким образом, чтобы связанный с ней инвариант цикла перед первой итерацией применялся к непустому подмассиву, и соответствующим образом модифицируйте доказательство леммы 5.5 для своей процедуры.
10. Профессор решил разработать алгоритм, в результате выполнения которого получались бы все случайные перестановки, кроме тождественной. Он предложил такую процедуру.

Permute-Without-Identity(A)

1 $n = A.length$

2 $\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ n - 1$

Обменять $A[i]$ с $A[Random(i+1,n)]$

Добьется ли профессор поставленной цели с помощью этого кода?

11. Предположим, что вместо того, чтобы менять местами элемент $A[i]$ со случайно выбранным элементом из подмассива $A[i..n]$, мы меняем его местами с любым случайно выбранным элементом массива A.

Pernute-With-All(A)

$n = A.length$

$\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ n$

Обменять $A[i]$ с $A[Random(1,n)]$

Получится ли в результате выполнения этого кода равномерная случайная перестановка? Обоснуйте свой ответ.

12. Профессор предложил для генерации случайных перестановок с однородным распределением такую процедуру.

Pernute-By-Cyclic(A)

$n = A.length$

Пусть $B[1..n]$ — новый массив

$Offset = Random(1,n)$

$\boldsymbol{for}\ i = 1\ \boldsymbol{to}\ n$

$\quad dest = i + offset$

$\quad \boldsymbol{if}\ dest > n$

$\quad\quad dest = dest - n$

$\quad B[dest] = A[i]$

return B

13. Покажите, что каждый элемент $A[i]$ оказывается в определенной позиции массива B с вероятностью $1/n$. Затем покажите, что алгоритм профессора ошибочен в том смысле, что полученные в результате его выполнения перестановки не будут распределены равномерно.

14. Докажите, что в результате выполнения процедуры Permute By Sorting вероятность отсутствия одинаковых элементов в массиве P не меньше величины $1 - \frac{1}{n}$.

15. Объясните, как следует реализовать алгоритм Permute By Sorting в случае, когда два или более приоритетов идентичны. Другими словами, алгоритм должен выдавать случайные равномерно распределенные перестановки даже в случае, если два или более приоритетов имеют одинаковые значения

16. Предположим, что мы хотим создать **случайную выборку** из множества $\{1,2,3,\dots,n\}$, те. m-элементное подмножество S, где $0 \le \mathrm{m} \le \mathrm{n}$, такое, что каждое m-подмножество создается с одинаковой вероятностью. Один из способов состоит в присваивании $A[i] = i$ для $i = 1,2,3,\dots,n$ и вызове Randomize-In-Place(A), после чего следует просто взять первые m элементов массива. Этот метод выполняет п вызовов процедуры Random. Если n гораздо меньше m, можно создать случайную выборку с помощью меньшего количества вызовов Random. Покажите, что приведенная далее рекурсивная процедура возвращает случайное m-подмножество множества S, состоящего из элементов $\{1,2,3,\dots,n\}$, причем каждое m-подмножество равновероятно, и при этом процедура Random вызывается только m раз.

Random-Sample(m,n)

```
if m == 0
    return ∅
else S = Random Sample(m − 1, n − 1)
    i = random(1, n)
    if i ∈ S
        S = S ∪ {n}
    else S = S ∪ {i}
    return S
```

17. Сколько человек должно собраться в комнате, чтобы вероятность того, что день рождения у кого-нибудь из них совпадет с вашим, была не меньшей $1/2$? Сколько человек необходимо, чтобы вероятность того, что хотя бы двое из них родились 7 ноября, превысила величину $1/2$?

18. Предположим, что мы наполняем b корзин шарами до тех пор, пока в какой-то из корзин не окажется два шара. Все опускания шаров выполняются независимо, и шар с равной вероятностью может оказаться в любой корзине. Чему равно математическое ожидание количества опущенных в корзины шаров?

19. В ходе анализа парадокса дней рождения было принято предположение о взаимной независимости всех дней рождения. Является ли это предположение существенным, или достаточно попарной независимости? Обоснуйте свой ответ.

20. Сколько человек нужно пригласить на вечеринку, чтобы вероятность того, что трое из них родились в один и тот же день, достигла заметной величины?

21. Какова вероятность того, что строка длиной k, составленная из символов -элементного множества, является размещением k элементов этого множества? Как этот вопрос связан с парадоксом дней рождения?

22. Предположим, что n шаров распределяются по n корзинам. Каждый шар опускается независимо от других и с равной вероятностью может оказаться в любой из корзин. Чему равно математическое ожидание количества пустых корзин? Чему равно математическое ожидание количества корзин с одним шаром?

23. Уточните нижнюю оценку длины последовательности выпадений орлов. Для этого покажите, что при n подбрасываниях симметричной монеты вероятность того, что такая последовательность будет не длиннее $\lg n - 2\lg\lg n$, меньше $1/n$.

Задачи

1. Вероятностный подсчет

С помощью -битового счетчика можно вести подсчет до $2^b - 1$ элементов. Предложенный Р. Моррисом (R. Morris) ***вероятностный подсчет*** (probabilistic counting) позволяет проводить нумерацию намного большего количества элементов ценой потери точности.

Пусть значение переменной-счетчика $i = 0, 1, \dots, 2^b - 1$ означает номер элемента n_i возрастающей последовательности неотрицательных чисел. Будем считать, что начальное значение счетчика равно нулю, т.е.$n_0 = 0$. Операция $Increment$ увеличивает значение счетчика i случайным образом. Если $i = 2^b - 1$, то в результате этого действия выдается сообщение о переполнении. В противном случае значение счетчика с вероятностью $1/(n_{i+1} — n_i)$ возрастает на единицу и остается неизменным с вероятностью $1 - 1/(n_{i+1} — n_i)$

Если для всех $i \geq 0$ выбрать $n_i = i$, то мы получим обычный счетчик. Более интересная ситуация возникает, если выбрать, скажем, = $2^{1\,1}$ для i > 0 или n_i= Fi (i-е число Фибоначчи; см. раздел 3.2).

В задаче предполагается, что число п2ь_1 достаточно большое, чтобы вероятностью переполнения можно было пренебречь.

а. Покажите, что математическое ожидание значения счетчика после применения к нему п операций $Increment$ равно n.

б. Анализ дисперсии значения счетчика зависит от выбора последовательности n_i•. Рассмотрим простой случай, когда $n_i = 100i$ для всех $i \geq 0$. Оцените дисперсию значения счетчика после выполнения операции $Increment$ n раз.

2. Поиск в неотсортированном массиве

В этой задаче исследуются три алгоритма поиска значения x в неотсортированном -элементном массиве A.

Рассмотрим следующую рандомизированную стратегию: выбираем элемент массива A с произвольным индексом i и проверяем справедливость равенства $A[i] = x$. Если оно выполняется, то алгоритм завершается. В противном случае продолжаем поиск, случайным образом выбирая новые элементы массива A. Перебор индексов продолжается до тех пор, пока не будет найден такой индекс j, что $A[i] = x$, или пока не будут проверены все элементы массива. Заметим, что выбор каждый раз производится среди всех индексов массива, поэтому круг поиска не сужается и один и тот же элемент может проверяться неоднократно.

а. Напишите псевдокод процедуры `Random-Search`, реализующей описанную стратегию. Позаботьтесь, чтобы алгоритм прекращал работу после того, как будут проверены все индексы массива.

б. Предположим, что имеется ровно один индекс i, такой, что $A[i] = x$. Чему равно математическое ожидание количества индексов в массиве A, которые будут проверены до того, как будет найден элемент x и процедура `Random-Search` завершит работу?

в. Обобщите решение сформулированной в п. (б) задачи для ситуации, когда имеются $k \geq 1$ индексов i, таких, что $A[i] = x$. Чему равно математическое ожидание количества индексов в массиве A, которые будут проверены до того, как будет найден элемент с и процедура `Random-Search` завершит работу? Ответ должен представлять собой функцию от величин n и k.

г. Предположим, что условие $A[i] = x$ не выполняется ни для какого индекса i. Чему равно математическое ожидание количества индексов в массиве A, которые придется проверить до того, как будут проверены все элементы массива и процедура `Random-Search` завершит работу?

Теперь рассмотрим детерминированный алгоритм линейного поиска `Deterministic-Search`. В этом алгоритме поиск элемента с производится путем последовательной проверки элементов $A[1]$, $A[2]$, $A[3]$, …, $A[n]$ до тех пор, пока не произойдет одно из двух событий: либо будет найден элемент $A[i] = x$, либо будет достигнут конец массива. Предполагается, что все возможные перестановки элементов входного массива встречаются с одинаковой вероятностью.

д. Предположим, что имеется всего один индекс i, такой, что $A[i]$ с. Чему равно время работы процедуры `Deterministic-Search` в среднем случае? Как ведет себя эта величина в наихудшем случае?

е. Обобщите решение сформулированной в п. (д) задачи для ситуации, когда имеются $k \geq 1$ индексов i, для которых $A[i] = x$. Чему равно время работы процедуры `Deterministic-Search` в среднем случае? Как ведет себя эта величина в наихудшем случае? Ответ должен быть функцией от n и k.

ж. Предположим, что условие $A[i] = x$ не выполняется ни для какого элемента массива A. Чему равно время работы процедуры `Deterministic-Search` в среднем случае? Как ведет себя эта величина в наихудшем случае?

Наконец рассмотрим рандомизированную алгоритм `Scramble-Search`, в котором сначала выполняется случайная перестановка элементов входного массива, а затем в полученном массиве выполняется описанный выше линейный детерминированный поиск.

з. Пусть k — количество индексов i, таких, что $A[i] = x$. Определите математическое ожидание времени работы процедуры `Scramble-Search` и время ее работы в наихудшем случае для значений $k = 0$ и $k = 1$. Обобщите решение для случая $k \geq 1$.

и. Какой из трех представленных алгоритмов вы бы предпочли? Поясните свой ответ.

Методические рекомендации

При подготовке к практическим занятиям студент должен придерживаться следующих рекомендаций:

- внимательно изучить основные вопросы темы и план практического занятия,
- определить место темы занятия в общем содержании, ее связь с другими темами;
- найти и проработать соответствующие разделы в рекомендованных нормативных документах, учебниках и дополнительной литературе;
- после ознакомления с теоретическим материалом ответить на вопросы по теме курса;
- продумать пути и способы решения проблемных вопросов;

- продумать развернутые ответы на предложенные вопросы темы, опираясь на лекционные материалы, расширяя и дополняя их данными из учебников, дополнительной литературы.

В ходе практического занятия необходимо выполнить практическое задание, а затем объяснить методику его решения.

Источники и литература для подготовки:

Основная литература

1. Игошин, В.И. Сборник задач по математической логике и теории алгоритмов : учеб. пособие / В.И. Игошин. — Москва : КУРС ; ИНФРА-М, 2019. — 392 с. — (Бакалавриат). - ISBN 978-5-906818-08-9 (КУРС); ISBN 978-5-16-011429-3 (ИНФРА-М, print); ISBN 978-5-16-103684-6 (ИНФРА-М, online). - Текст : электронный. - URL: https://znanium.com/catalog/product/986940

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Практическое занятие 6 по теме «Алгоритмы сортировки»

Вопросы:

1. Парадокс дней рождения.
2. Анализ с применением индикаторной случайной величины.
3. Случайное наполнение корзин пронумерованными шарами.
4. Последовательность выпадения орлов.
5. Задача о найме в оперативном режиме.
6. Вероятностный подсчет.
7. Поиск в неотсортированном массиве.

Практические задания:

1. Чему равно минимальное и максимальное количества элементов в пирамиде высотой h?

2. Покажите, что n-элементная пирамида имеет высоту $\lfloor \lg n \rfloor$.
3. Покажите, что в любом поддереве невозрастающей пирамиды корень этого поддерева содержит наибольшее значение среди узлов поддерева.
4. Где в невозрастающей пирамиде может находиться наименьший ее элемент, если все элементы различаются по величине?
5. Является ли массив с отсортированными элементами неубывающей пирамидой?
6. Является ли последовательность значений (23, 17, 14, 6, 13, 10, 1, 5, 7, 12) невозрастающей пирамидой?
7. Покажите, что если n-элементную пирамиду представить в виде массива, то ее листьями будут элементы с индексами $\lfloor n/2 \rfloor + 1$, $\lfloor n/2 \rfloor + 2$, …, n.
8. Воспользовавшись рис. 6.2 в качестве образца, проиллюстрируйте работу процедуры Max-Heapify(A, i) с массивом A = (27, 17, 3, 16, 13, 10, 1, 5, 7, 12, 4, 8, 9, 0).
9. Используя в качестве отправной точки процедуру Max-Heapify, напишите псевдокод процедуры Min-Heapify (A, i), которая выполняет соответствующие действия над неубывающей пирамидой. Каково время работы процедуры Min-Heapify по сравнению со временем работы процедуры Max-Heapify?
10. Как влияет на вызов процедуры Max-Heapify (A, i) ситуация, когда элемент $A[i]$ больше, чем его дочерние элементы?
11. К чему приведет вызов процедуры Max-Heapify (A, i) в случае $i > A.heap\text{-}size/2$?
12. Код процедуры Max-Heapify достаточно рационален, если не считать рекурсивного вызова в строке 10, из-за которого некоторые компиляторы могут сгенерировать неэффективный код. Напишите эффективную процедуру Max-Heapify, в которой вместо рекурсивного вызова использовалась бы итеративная управляющая конструкция (цикл).
13. Покажите, что в наихудшем случае время работы процедуры Max-Heapify на пирамиде размером n равно $\Omega(lg\, n)$. (Указание: в пирамиде с n узлами присвойте узлам такие значения, чтобы процедура Max-Heapify рекурсивно вызывалась в каждом узле, расположенном на простом пути от корня до листа.)
14. Воспользовавшись в качестве образца рис. 6.3, проиллюстрируйте работу процедуры Build-Max-Heap над входным массивом A = (5, 3, 17, 10, 84, 19, 6, 22, 9).
15. Почему индекс цикла i в строке 2 процедуры Build-Max-Heap убывает от $\lfloor A.\ length/2 \rfloor$ до 1, а не возрастает от 1 до $\lfloor A.\ length/2 \rfloor$?
16. Покажите, что в любой n-элементной пирамиде на высоте h находится не более $\lceil n/2^{h+1} \rceil$ узлов.

17. Воспользовавшись в качестве образца рис. 6.4, проиллюстрируйте работу процедуры HEAPSORT над входным массивом A = (5, 13, 2, 25, 7, 17, 20, 8, 4).
18. Докажите корректность процедуры Heapsort с помощью следующего инварианта цикла: «В начале каждой итерации цикла **for** в строках 2-5 подмассив A[1 ... i] представляет собой невозрастающую пирамиду, содержащую i наименьших элементов массива A[1 ... n], а в подмассиве A[i+1 ... n] содержатся n-i наибольших элементов массива A[1 … n] в отсортированном состоянии».
19. Чему равно время работы процедуры Heapsort с массивом A длиной n, в котором элементы отсортированы и расположены в порядке возрастания? В порядке убывания?
20. Покажите, что время работы процедуры Heapsort в наихудшем случае составляет $\Omega(n \lg n)$
21. Покажите, что, когда все элементы различны, время работы процедуры Heapsort в наилучшем случае равно $\Omega(n \lg n)$
22. Проиллюстрируйте работу процедуры Heap-Extract-Max над пирамидой A = (15, 13, 9, 5, 12, 8, 7, 4,0, 6, 2, 1).
23. Проиллюстрируйте операцию Max-Heap-Insert(A, *10*) над пирамидой A = (15, 13, 9, 5, 12, 8, 7, 4, 0, 6, 2, 1).
24. Напишите псевдокоды процедур Heap-Minimum, Heap-Extract-Min, Heap-Decrease-Key и Min-Heap-Insert, реализующих неубывающую очередь с приоритетами на базе неубывающей пирамиды.
25. Зачем нужна такая мера предосторожности, как присвоение ключу добавляемого в строке 2 процедуры Max-Heap-Insert в пирамиду узла значения -∞, если уже на следующем шаге значение этого ключа увеличивается до требуемой величины?
26. Докажите корректность процедуры Heap-Increase-Key с помощью следующего инварианта цикла: «В начале каждой итерации цикла **while** в строках 4-6 A[Parent(i) ≥ A[Left(i)] и A[Parent(i) ≥ A[Right(i)], если эти узлы существуют, а подмассив A[1 . . *A.heap-size*] удовлетворяет свойству невозрастающей пирамиды, за исключением, возможно, одного нарушения: A[i] может быть больше, чем A[Parent(i)]». Можно считать, что в момент вызова процедуры Heap-Increase-Key подмассив A[1 . . *A.heap-size*] удовлетворяет свойству невозрастающей пирамиды.
27. Каждая операция обмена в строке 5 процедуры Heap-Increase-Key обычно требует трех присваиваний. Покажите, как воспользоваться идеей внутреннего цикла процедуры Insertion-Sort, чтобы вместо трех присваиваний обойтись только одним.

28. Покажите, как с помощью очереди с приоритетами реализовать очередь ”первым вошел — первым вышел'! Продемонстрируйте, как с помощью очереди с приоритетами реализовать стек. (Очереди и стеки определены в разделе 10.1)
29. Процедура Heap-Delete (*A, i*) удаляет из пирамиды *A* узел *i*. Разработайте реализацию этой процедуры, которой требуется время $O(\lg n)$ для удаления узла из *n*-элементной невозрастающей пирамиды.
30. Разработайте алгоритм, объединяющий *k* отсортированных списков в один список за время $O(n \lg k)$, где *n* — общее количество элементов во всех входных списках. (Указание: для слияния списков воспользуйтесь неубывающей пирамидой.)

6. 195

Задачи

1. Построение пирамиды вставками

Пирамиду можно построить с помощью многократного вызова процедуры Max-Heap-Insert для вставки элементов в пирамиду. Рассмотрим следующий вариант процедуры Build-Max-Heap.

```
Build-Max-Heap' (A)
    A. heap-size = 1
    for i = 2 to A. length
        Max-Heap-Insert (A, A[i])
```

а. Всегда ли процедуры Build-Max-Heap и Build-Max-Heap' для одного и того же входного массива создают одну и ту же пирамиду? Докажите, что это так, или приведите контрпример.

б. Покажите, что в наихудшем случае для создания п-элементной пирамиды процедуре Build-Max-Heap' потребуется время $\theta(n \lg n)$

2. Анализ d-арных пирамид

d-арные пирамиды подобны бинарным, но отличаются тем, что все внутренние узлы (с одним возможным исключением) имеют вместо двух *d* дочерних узлов.

а. Как бы вы представили *d*-арную пирамиду в виде массива?

б. Как выражается высота *d*-арной *n*-элементной пирамиды через *n* и *d*?

в. Разработайте эффективную реализацию процедуры Extract-Max, предназначенную для работы с *d*-арной невозрастающей пирамидой. Проанализируйте время работы этой процедуры и выразите его через *d* и *n*.

г. Разработайте эффективную реализацию процедуры Insert, предназначенную для работы с *d*-арной невозрастающей пирамидой. Проанализируйте время работы этой процедуры и выразите его через *d* и *n*.

д. Разработайте эффективную реализацию процедуры Increase-Key (*A, i, k*), которая при $k < A[i]$ сообщает об ошибке, а в противном случае выполняег присваивание $A[i] = k$ и соответствующим образом обновляет структуру *d*-арной невозрастающей пирамиды. Проанализируйте время работы этой процедуры и выразите его через *d* и *n*.

3. Таблицы Юнга

Таблица Юнга (Young tableau) $m \times n$ представляет собой матрицу размером $m \times n$, элементы которой в каждой строке отсортированы слева направо, а в каждом столбце — сверху вниз. Некоторые элементы таблицы Юнга могут быть равны ∞, что трактуется как отсутствие элемента. Таким образом, таблицу Юнга можно использовать для хранения $r \leq mn$ конечных чисел.

а. Начертите таблицу Юнга 4 x 4, в которой содержатся элементы {9, 16, 3, 2, 4, 8. 5, 14, 12}.

б. Докажите, что таблица Юнга У размером $m \times n$ пуста, если $Y[1, 1] = \infty$Докажите, что таблица Y полностью заполнена (т.е. содержит mn элементов), если $Y[m,n] < \infty$.

в. Разработайте алгоритм, реализующий процедуру Extract-Min для непустой таблицы Юнга $m \times n$ за время $O(m + n)$. В алгоритме следует использовать рекурсивную подпрограмму, которая решает задачу размером $m \times n$ путем рекурсивного сведения к задачам $(m\text{-}1) \times n$ или $m \times (n\text{-}1)$. (Указание: вспомните о процедуре Max-Heapify.) Обозначим максимальное время обработки произвольной таблицы Юнга $m \times n$ с помощью процедуры Extract-Min как $T(p)$, где $p = m + n$. Запишите и решите рекуррентное соотношение, которое дает для $T(p)$ границу $O(m + n)$.

г. Покажите, как вставить новый элемент в незаполненную таблицу Юнга размером $m \times n$ за время $O(m + n)$.

д. Покажите, как с помощью таблицы Юнга $m \times n$ выполнить сортировку n^2чисел за время $O(n^3)$ не используя при этом никаких других подпрограмм сортировки.

е. Разработайте алгоритм, позволяющий за время $O(m + n)$ определить, содержится ли в таблице Юнга размером $m \times n$ заданное число.

Методические рекомендации

При подготовке к практическим занятиям студент должен придерживаться следующих рекомендаций:

- внимательно изучить основные вопросы темы и план практического занятия,
- определить место темы занятия в общем содержании, ее связь с другими темами;
- найти и проработать соответствующие разделы в рекомендованных нормативных документах, учебниках и дополнительной литературе;
- после ознакомления с теоретическим материалом ответить на вопросы по теме курса;
- продумать пути и способы решения проблемных вопросов;
- продумать развернутые ответы на предложенные вопросы темы, опираясь на лекционные материалы, расширяя и дополняя их данными из учебников, дополнительной литературы.

В ходе практического занятия необходимо выполнить практическое задание, а затем объяснить методику его решения.

Источники и литература для подготовки:

Основная литература

1. Игошин, В.И. Сборник задач по математической логике и теории алгоритмов : учеб. пособие / В.И. Игошин. — Москва : КУРС ; ИНФРА-М, 2019. — 392 с. — (Бакалавриат). - ISBN 978-5-906818-08-9 (КУРС); ISBN 978-5-16-011429-3 (ИНФРА-М, print); ISBN 978-5-16-103684-6 (ИНФРА-М, online). - Текст : электронный. - URL: https://znanium.com/catalog/product/986940

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ К САМОСТОЯТЕЛЬНОЙ РАБОТЕ

Самостоятельная работа по теме «Роль алгоритмов в вычислениях»

Вопросы:

1. Что такое алгоритмы?
2. Какие задачи решаются с помощью алгоритмов?
3. Алгоритмы как технология.
4. Эффективность.
5. Алгоритмы и другие технологии.

Методические рекомендации

Аудиторная самостоятельная работа по учебной дисциплине осуществляется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется по заданию преподавателя без его непосредственного участия.

Самостоятельная работа студентов при подготовке к лекции заключается в рассмотрении общих научных основ и анализе конкретных процессов и факторов, определяющих содержание темы.

Самостоятельная работа студентов при подготовке к практическому занятию включает подбор материала, данных и специальных источников, с которыми предстоит учебная работа, а также решение ситуационных и практических заданий. В связи с этим студентам рекомендуется детально разобрать теоретические вопросы лекционного курса, а затем закрепить материал в процессе решения проблемных ситуаций, задач.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы, то нужно сравнить их и выбрать самый рациональный. Решение проблемных задач следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями и схемами. Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом.

Самостоятельная подготовка к зачету должна осуществляться в течение всего семестра, а не за несколько дней до его проведения. При подготовке к зачету студентам рекомендуется:

- перечитать все лекции, а также материалы, которые готовились к практическим занятиям в течение семестра.

- соотнести эту информацию с вопросами, которые даны к зачету. Если информации недостаточно, ответы находят в предложенной преподавателем литературе.

При подготовке к зачету рекомендуется делать краткие записи для формирования четкой логической схемы ответа на вопрос.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Самостоятельная работа по теме «Основы разработки и анализа алгоритмов»

Вопросы:

1. Алгоритм сортировки вставкой.
2. Анализ алгоритма сортировки вставкой.
3. Задача сортировки выбором.
4. Анализ алгоритма сортировки выбором.
5. Разработка алгоритма сортировки слиянием.
6. Анализ алгоритма сортировки слиянием.

Методические рекомендации

Аудиторная самостоятельная работа по учебной дисциплине осуществляется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется по заданию преподавателя без его непосредственного участия.

Самостоятельная работа студентов при подготовке к лекции заключается в рассмотрении общих научных основ и анализе конкретных процессов и факторов, определяющих содержание темы.

Самостоятельная работа студентов при подготовке к практическому занятию включает подбор материала, данных и специальных источников, с которыми предстоит учебная работа, а также решение ситуационных и практических заданий. В связи с этим студентам рекомендуется детально разобрать теоретические вопросы лекционного курса, а затем закрепить материал в процессе решения проблемных ситуаций, задач.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы, то нужно сравнить их и выбрать самый рациональный. Решение проблемных задач следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями и схемами. Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом.

Самостоятельная подготовка к зачету должна осуществляться в течение всего семестра, а не за несколько дней до его проведения. При подготовке к зачету студентам рекомендуется:

- перечитать все лекции, а также материалы, которые готовились к практическим занятиям в течение семестра.

- соотнести эту информацию с вопросами, которые даны к зачету. Если информации недостаточно, ответы находят в предложенной преподавателем литературе.

При подготовке к зачету рекомендуется делать краткие записи для формирования четкой логической схемы ответа на вопрос.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Самостоятельная работа по теме «Рост функций»

Вопросы:

1. Асимптотические обозначения.

2. Функции и время работы.
3. Асимптотические обозначения в уравнениях и задачах.
4. Стандартные обозначения и часто встречающиеся функции.

Методические рекомендации

Аудиторная самостоятельная работа по учебной дисциплине осуществляется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется по заданию преподавателя без его непосредственного участия.

Самостоятельная работа студентов при подготовке к лекции заключается в рассмотрении общих научных основ и анализе конкретных процессов и факторов, определяющих содержание темы.

Самостоятельная работа студентов при подготовке к практическому занятию включает подбор материала, данных и специальных источников, с которыми предстоит учебная работа, а также решение ситуационных и практических заданий. В связи с этим студентам рекомендуется детально разобрать теоретические вопросы лекционного курса, а затем закрепить материал в процессе решения проблемных ситуаций, задач.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы, то нужно сравнить их и выбрать самый рациональный. Решение проблемных задач следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями и схемами. Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом.

Самостоятельная подготовка к зачету должна осуществляться в течение всего семестра, а не за несколько дней до его проведения. При подготовке к зачету студентам рекомендуется:

- перечитать все лекции, а также материалы, которые готовились к практическим занятиям в течение семестра.

- соотнести эту информацию с вопросами, которые даны к зачету. Если информации недостаточно, ответы находят в предложенной преподавателем литературе.

При подготовке к зачету рекомендуется делать краткие записи для формирования четкой логической схемы ответа на вопрос.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с.

— (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018

2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Самостоятельная работа по теме «Разделяй и властвуй»

Вопросы:

1. Задача поиска максимального подмассива.
2. Метод подстановки решения рекуррентных соотношений.
3. Анализ алгоритма поиска максимального подмассива.
4. Алгоритм Штрассена для умножения матриц.
5. Метод подстановки решения рекуррентных соотношений.
6. Как угадать решение и избежать ошибок.
7. Замена переменных.
8. Метод деревьев рекурсии.
9. Основной метод.
10. Основная теорема о рекуррентных соотношениях.
11. Использование основного метода.

Методические рекомендации

Аудиторная самостоятельная работа по учебной дисциплине осуществляется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется по заданию преподавателя без его непосредственного участия.

Самостоятельная работа студентов при подготовке к лекции заключается в рассмотрении общих научных основ и анализе конкретных процессов и факторов, определяющих содержание темы.

Самостоятельная работа студентов при подготовке к практическому занятию включает подбор материала, данных и специальных источников, с которыми предстоит учебная работа, а также решение ситуационных и практических заданий. В связи с этим студентам рекомендуется детально разобрать теоретические вопросы лекционного курса, а затем закрепить материал в процессе решения проблемных ситуаций, задач.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы, то нужно сравнить их и выбрать самый рациональный. Решение проблемных задач следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями и схемами. Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом.

Самостоятельная подготовка к зачету должна осуществляться в течение всего семестра, а не за несколько дней до его проведения. При подготовке к зачету студентам рекомендуется:

- перечитать все лекции, а также материалы, которые готовились к практическим занятиям в течение семестра.
- соотнести эту информацию с вопросами, которые даны к зачету. Если информации недостаточно, ответы находят в предложенной преподавателем литературе.

При подготовке к зачету рекомендуется делать краткие записи для формирования четкой логической схемы ответа на вопрос.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Самостоятельная работа по теме «Вероятностный анализ и рандомизированные алгоритмы»

Вопросы:

1. Задача о найме.
2. Анализ наихудшего случая в задаче о найме.
3. Вероятностный анализ.
4. Рандомизированные алгоритмы.
5. Индикаторная случайная величина.
6. Лемма о математическом ожидании индикаторной случайной величины.
7. Лемма о математическом ожидании количества наймов.
8. Анализ задачи о найме с помощью индикаторных случайных величин.
9. Задачи о гардеробщике и инверсии массива.

10. Изменения, которые требуется внести в алгоритм найма для рандомизации. Код случайной перестановки.
11. Лемма о математическом ожидании стоимости найма с кодом случайной перестановки.
12. Массивы после случайной перестановки. Лемма о равномерном распределении.

Методические рекомендации

Аудиторная самостоятельная работа по учебной дисциплине осуществляется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется по заданию преподавателя без его непосредственного участия.

Самостоятельная работа студентов при подготовке к лекции заключается в рассмотрении общих научных основ и анализе конкретных процессов и факторов, определяющих содержание темы.

Самостоятельная работа студентов при подготовке к практическому занятию включает подбор материала, данных и специальных источников, с которыми предстоит учебная работа, а также решение ситуационных и практических заданий. В связи с этим студентам рекомендуется детально разобрать теоретические вопросы лекционного курса, а затем закрепить материал в процессе решения проблемных ситуаций, задач.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы, то нужно сравнить их и выбрать самый рациональный. Решение проблемных задач следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями и схемами. Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом.

Самостоятельная подготовка к зачету должна осуществляться в течение всего семестра, а не за несколько дней до его проведения. При подготовке к зачету студентам рекомендуется:

- перечитать все лекции, а также материалы, которые готовились к практическим занятиям в течение семестра.

- соотнести эту информацию с вопросами, которые даны к зачету. Если информации недостаточно, ответы находят в предложенной преподавателем литературе.

При подготовке к зачету рекомендуется делать краткие записи для формирования четкой логической схемы ответа на вопрос.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

Самостоятельная работа по теме «Алгоритмы сортировки»

Вопросы:

1. Парадокс дней рождения.
2. Анализ с применением индикаторной случайной величины.
3. Случайное наполнение корзин пронумерованными шарами.
4. Последовательность выпадения орлов.
5. Задача о найме в оперативном режиме.
6. Вероятностный подсчет.
7. Поиск в неотсортированном массиве.

Методические рекомендации

Аудиторная самостоятельная работа по учебной дисциплине осуществляется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется по заданию преподавателя без его непосредственного участия.

Самостоятельная работа студентов при подготовке к лекции заключается в рассмотрении общих научных основ и анализе конкретных процессов и факторов, определяющих содержание темы.

Самостоятельная работа студентов при подготовке к практическому занятию включает подбор материала, данных и специальных источников, с которыми предстоит учебная работа, а также решение ситуационных и практических заданий. В связи с этим студентам рекомендуется детально разобрать теоретические вопросы лекционного курса, а затем закрепить материал в процессе решения проблемных ситуаций, задач.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы, то нужно сравнить их и выбрать самый рациональный. Решение проблемных задач следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями и схемами. Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом.

Самостоятельная подготовка к зачету должна осуществляться в течение всего семестра, а не за несколько дней до его проведения. При подготовке к зачету студентам рекомендуется:

- перечитать все лекции, а также материалы, которые готовились к практическим занятиям в течение семестра.

- соотнести эту информацию с вопросами, которые даны к зачету. Если информации недостаточно, ответы находят в предложенной преподавателем литературе.

При подготовке к зачету рекомендуется делать краткие записи для формирования четкой логической схемы ответа на вопрос.

Источники и литература для подготовки:

Основная литература

1. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат) - Текст : электронный. - URL: https://znanium.com/catalog/product/956763

Дополнительная литература

1. Судоплатов, С. В. Математическая логика и теория алгоритмов : учебник и практикум для академического бакалавриата / С. В. Судоплатов, Е. В. Овчинникова. — 5-е изд., стер. — Москва : Издательство Юрайт, 2019. — 255 с. — (Высшее образование). — ISBN 978-5-534-00767-1. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/432018
2. Крупский, В. Н. Теория алгоритмов. Введение в сложность вычислений : учебное пособие для бакалавриата и магистратуры / В. Н. Крупский. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 117 с. — (Авторский учебник). — ISBN 978-5-534-04817-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/444131

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Содержание изменения	Измененные пункты	Решение Учебно-методического совета