

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Томский государственный университет систем управления и радиоэлектроники»
(ТУСУР)

Кафедра автоматизированных систем управления
(АСУ)

Структуры и алгоритмы обработки данных в ЭВМ

Методические указания по выполнению лабораторных работ студентов
всех форм обучения для направления подготовки 230100.62
"Информатика и вычислительная техника"

2011 г.

Горитов А.Н.

Структуры и алгоритмы обработки данных в ЭВМ: методические указания по выполнению лабораторных работ студентов всех форм обучения для направления подготовки: 230100.62 – "Информатика и вычислительная техника" / А.Н. Горитов. – Томск: ТУСУР, 2011. – 15 с.

Методические указания разработаны в соответствии с решением кафедры автоматизированных систем управления

Составитель: д.т.н., профессор каф. АСУ А.Н. Горитов

Методические указания утверждены на заседании кафедры автоматизированных систем управления 30 августа 2011 г., протокол № 1

Содержание

1. Цели и задачи дисциплины	4
2. Методы и формы организации обучения	4
3. Место дисциплины в структуре ООП.....	5
4. Лабораторный практикум	5
4.1 Перечислимые и интервальные типы	6
4.2 Множества	7
4.3 Стеки, очереди.....	7
4.4 Связанные списки	8
4.5 Бинарные деревья.....	9
4.6 Сортировка.....	9
4.7 Внешняя сортировка.....	10
4.8 Динамическое программирование	11
4.9 Поиск подстрок	11
4.10 Фундаментальные алгоритмы на графах.....	12
4.11 Кратчайшие пути в графе.....	13
5. Учебно-методическое и информационное обеспечение дисциплины....	14
5.1 Основная литература	14
5.2 Дополнительная литература	14

1. Цели и задачи дисциплины

Целью изучения дисциплины «Структуры и алгоритмы обработки данных в ЭВМ» является изучение применяемых в программировании (и информатике) структур данных, их спецификации и реализации, а также алгоритмов обработки данных и анализ этих алгоритмов, взаимосвязь алгоритмов и структур.

В результате изучения дисциплины студент должен:

а) иметь представление об основных тенденциях в создании структур данных, методах оптимального использования памяти и времени для обработки структур данных и управления процессами обработки данных;

б) знать и использовать различные (динамические и статистические) структуры данных в соответствии с запросами алгоритмов;

в) создавать списковые и древообразные структуры и управлять организацией этих структур (изменение списков и деревьев посредством включения исключения, замены элементов структур) знать, использовать оптимальные методы поиска и сортировки данных;

г) знать и использовать основные алгоритмы решения классических задач информатики;

д) иметь представление о математических методах анализа алгоритмов; классификации алгоритмических задач по сложности, сводимости алгоритмических задач к известным задачам определенного класса сложности;

е) иметь опыт работы с алгоритмическими языками программирования.

2. Методы и формы организации обучения

Процесс изучения дисциплины «Структуры и алгоритмы обработки данных в ЭВМ» направлен на формирование следующих компетенций:

общекультурные компетенции (ОК):

- Владеет культурой мышления, способен к обобщению, анализу, восприятию информации, постановке цели и выбору путей её достижения (ОК-1);
- Имеет навыки работы с компьютером как средством управления информацией (ОК-12);

профессиональные компетенции (ПК):

- Способностью осваивать методики использования программных средств для решения практических задач (ПК-2);
- Способностью разрабатывать модели компонентов информационных систем, включая модели баз данных (ПК-4);

- Способностью разрабатывать компоненты программных комплексов и баз данных, использовать современные инструментальные средства и технологии программирования (ПК-5).

профессионально-специализированные компетенции (ПСК):

- Работать с различными видами исходных данных о предметной области (ПСК-1);
- Осуществлять разработку программного обеспечения на современных языках программирования (ПСК-10);
- Осуществлять отладку программ (ПСК-11).

Для успешного освоения дисциплины применяются различные образовательные технологии, которые обеспечивают достижение планируемых результатов обучения согласно основной образовательной программе, с учетом требований к объему занятий в интерактивной форме.

Интерактивные формы обучения, которые используются в данном курсе, включают: «Работа в команде» и «Поисковый метод».

Для контроля освоения компетенций используются следующие формы контроля: защита лабораторных работ, опрос по изучаемым разделам дисциплины, тесты.

3. Место дисциплины в структуре ООП

Дисциплина «Структуры и алгоритмы обработки данных в ЭВМ» относится к вариативной части профессионального цикла ООП. Успешное овладение дисциплиной предполагает предварительные знания таких дисциплин как «Математический анализ», «Дискретная математика», «Программирование», «Математическая логика и теория алгоритмов» в объеме, предусмотренном направлением 230100.62 – «Информатика и вычислительная техника».

Знания и навыки, полученные при изучении этой дисциплины, используются при изучении дисциплин «Основы разработки программного обеспечения», «Теория языков программирования и методы трансляции», «Теория вычислительных процессов», «Базы данных».

4. Лабораторный практикум

Лабораторный практикум дисциплины "Структуры и алгоритмы обработки данных в ЭВМ" позволяет получить практические навыки использования изучаемых структур данных и эффективных алгоритмов решения различных задач.

Результаты выполнения лабораторных работ представляются в виде отчета, который состоит из следующих пунктов:

1. Тема
2. Цель работы
3. Задание
4. Алгоритм решения задачи
5. Текст программы
6. Результаты работы программы
7. Выводы

Порядок выполнения лабораторных работ

- 1) изучить теоретический материал по теме лабораторной работы;
- 2) составить программу на одном из алгоритмических языков программирования для заданного варианта задания;
- 3) выполнить отладку составленной программы и показать преподавателю;
- 4) составить и защитить отчет по лабораторной работе.

4.1 Перечислимые и интервальные типы

Целью лабораторной работы – изучение основных способов работы с перечислимыми и интервальными типами данных.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами дисциплины:

Типы данных, определяемые программистом. Интервальные типы данных

Методические рекомендации

При выполнении лабораторной работы рекомендуется следующая последовательность:

1. Изучить теорию по теме лабораторной работы.
2. Получить задание на выполнение лабораторной работы.
3. Используя один из алгоритмических языков выполнить задание на лабораторную работу.
4. Подготовить отчет по лабораторной работе.
5. Защитить выполненную лабораторную работу.

Контрольные вопросы

1. Перечислимый тип. Представление в памяти.
2. Операции, выполняемые над перечислимым типом.
3. Отличия перечислимого типа данных от стандартного типа данных
4. Задание интервального типа данных.

5. Что дает для программиста использование перечислимых и интервальных типов данных?

4.2 Множества

Цель лабораторной работы – изучение основных способов использования множеств.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами дисциплины:

Множества. Операции над множествами. Представление в памяти.

Методические рекомендации

При выполнении лабораторной работы рекомендуется следующая последовательность:

1. Изучить теорию по теме «Множества».
2. Получить задание на выполнение лабораторной работы.
3. Используя один из алгоритмических языков выполнить задание на лабораторную работу.
4. Подготовить отчет по лабораторной работе.
5. Защитить выполненную лабораторную работу.

Контрольные вопросы

1. Как задается тип данных «Множество»?
2. Как множества представляются в памяти ЭВМ.
3. Операции, выполняемые над множествами.

4.3 Стеки, очереди

Целью лабораторной работы является изучение способов представления стеков и очередей в оперативной памяти ЭВМ и реализация основных операций работы с ними.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами дисциплины:

Стек, очередь и дек. Стек, очередь и дек как абстрактные типы данных. Представление и реализация.

Примеры алгоритмов, использующих стек, очередь, дек.

Методические рекомендации

При выполнении лабораторной работы необходимо дать описание абстрактного типа данных «Стек» или «Очередь». Реализация основных

операции АДД выполняется в виде процедур или функций. Все данные в процедуры или функции передаются через формальные параметры.

Перед завершением выполнения программы необходимо убедиться, что вся динамическая память, использованная для создания динамических структур данных, освобождена. В противном случае обеспечить ее полное освобождение.

Контрольные вопросы

1. Понятие стека. Операции, выполняемые над стеком.
2. Понятие очереди. Операции, выполняемые над очередью.
3. Понятие дека. Операции, выполняемые над деком.
4. Представление стека с помощью массива. Выполнение основных операций.
5. Представление очереди с помощью массива. Выполнение основных операций.

4.4 Связанные списки

Целью лабораторной работы является изучение основных способов представления связанных списков в оперативной памяти ЭВМ и реализация основных операций работы со списками.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами дисциплины:

Односвязные, двусвязные, кольцевые списки и операции над ними. Представление и реализация (непрерывная, ссылочная в связанной памяти).

Методические рекомендации

При выполнении лабораторной работы необходимо дать описание абстрактного типа данных «Список». Реализация основных операции АДД выполняется в виде процедур или функций. Все данные в процедуры или функции передаются через формальные параметры.

Перед завершением выполнения программы необходимо убедиться, что вся динамическая память, использованная для создания динамических структур данных, освобождена. В противном случае обеспечить ее полное освобождение.

Контрольные вопросы

1. Отличие динамических структур от статических и полустатических.
2. Основные операции, выполняемые над списками.
3. Что такое кольцевой список.

4.5 Бинарные деревья

Цель лабораторной работы – изучение основных способов представления бинарных деревьев в оперативной памяти ЭВМ и практическая реализация алгоритма работы с деревьями.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами дисциплины:

Обходы бинарных деревьев: рекурсивные и не рекурсивные алгоритмы. Обходы дерева или леса.

Представления и реализации бинарных деревьев: ссылочная реализация в связанной памяти, ссылочная реализация ограниченного бинарного дерева на базе вектора.

Методические рекомендации

Необходимо дать описание «дерева» как абстрактного типа данных. Затем реализовать основные операции АТД «Дерево» в виде процедур или функций, при этом данные в процедуры или функции должны передаваться через формальные параметры.

Во всех заданиях предполагается вывод сформированного дерева на экран.

Перед завершением выполнения программы необходимо убедиться, что вся динамическая память, использованная для создания динамических структур данных, освобождена. В противном случае обеспечить ее полное освобождение.

Контрольные вопросы

1. Что такое «полное бинарное дерево»?
2. Что такое «почти полное бинарное дерево»?
3. Сколько узлов в полном бинарном дереве высоты h ?
4. Какова минимальная высота бинарного дерева с n узлами?
5. Какие операции определены на АТД «Бинарное дерево поиска»?

4.6 Сортировка

Цель лабораторной работы – изучение основных способов сортировки данных.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами дисциплины:

Стратегии внутренней сортировки. Сортировка выборкой: метод простого выбора, турнирная сортировка, пирамидальная сортировка, анализ сложности алгоритмов. Сортировка включением: метод простых вста-

вок, метод вставки с убывающим шагом, анализ сложности алгоритмов. Обменные сортировки: сортировка пузырьком, быстрая сортировка, анализ сложности алгоритмов. Сортировка распределением: двоичная быстрая сортировка, цифровая распределяющая сортировка, блочная сортировка, сортировка подсчетом, анализ сложности алгоритмов. Сортировка слиянием.

Нижняя граница сложности задачи сортировки.

Методические рекомендации

При выполнении лабораторной работы рекомендуется следующая последовательность:

1. Изучить теорию по теме «Сортировка».
2. Получить задание на выполнение лабораторной работы.
3. Используя один из алгоритмических языков выполнить задание на лабораторную работу.
4. Подготовить отчет по лабораторной работе.
5. Защитить выполненную лабораторную работу.

Контрольные вопросы

1. В чем заключается сортировка методом вставки с убывающим шагом?
2. В чем заключается быстрая сортировка?
3. В чем заключается двоичная быстрая сортировка?
4. В чем заключается цифровая распределяющая сортировка?
5. В чем заключается сортировка слиянием?

4.7 Внешняя сортировка

Цель лабораторной работы – изучение основных способов сортировки данных.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами дисциплины:

Внешняя сортировка. Простое слияние. Естественное слияние. Многофазная сортировка.

Методические рекомендации

При выполнении лабораторной работы рекомендуется следующая последовательность:

1. Изучить теорию по теме «Сортировка».
2. Получить задание на выполнение лабораторной работы.
3. Используя один из алгоритмических языков выполнить задание на лабораторную работу.
4. Подготовить отчет по лабораторной работе.

5. Защитить выполненную лабораторную работу.

Контрольные вопросы

1. Что такое проход?
2. Из каких этапов состоит внешняя сортировка?
3. Что такое сбалансированное слияние?
4. В чем заключается однофазное слияние?
5. В чем заключается многофазная сортировка слиянием?

4.8 Динамическое программирование

Цель лабораторной работы – изучение основных способов применения динамического программирования.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами дисциплины:

Динамическое программирование. Пример и общая идея. Вычисление чисел Фибоначчи. Восходящее и нисходящее динамическое программирование. Задача определения наиболее длинной общей подпоследовательности. Задача определения порядка умножения цепочки матриц.

Методические рекомендации

При выполнении лабораторной работы рекомендуется следующая последовательность:

1. Изучить теорию по теме «Динамическое программирование».
2. Получить задание на выполнение лабораторной работы.
3. Используя один из алгоритмических языков выполнить задание на лабораторную работу.
4. Подготовить отчет по лабораторной работе.
5. Защитить выполненную лабораторную работу.

Контрольные вопросы

1. В чем заключается метод динамического программирования?
2. Для каких задач эффективен метод динамического программирования?
3. За счет чего этот метод позволяет получить более быстрые алгоритмы?
4. В чем отличия восходящего и нисходящего динамического программирования?

4.9 Поиск подстрок

Цель лабораторной работы – изучение эффективных алгоритмов поиска подстрок.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами дисциплины:

Задача поиска подстроки. Алгоритм Кнута-Морриса-Пратта. Алгоритм Боуера-Мура. Алгоритм Рабина-Карпа.

Методические рекомендации

При выполнении лабораторной работы рекомендуется следующая последовательность:

1. Изучить теоретический материал по алгоритмам поиска подстрок.
2. Разобрать работу этих алгоритмов на примерах.
3. Получить задание на выполнение лабораторной работы.
4. Используя один из алгоритмических языков выполнить задание на лабораторную работу.
5. Подготовить отчет по лабораторной работе.
6. Защитить выполненную лабораторную работу.

Контрольные вопросы

1. Что такое собственный суффикс и собственный префикс образца?
2. Как строится префикс-функция в алгоритме Кнута-Морриса-Пратта?
3. Что такое «эвристика безопасного суффикса»?
4. Что такое «эвристика стоп-символа»?
5. За счет чего в алгоритме Рабина-Карпа удается получить высокое быстроедействие?

4.10 Фундаментальные алгоритмы на графах

Цель лабораторной работы – изучение основных способов представления графов в оперативной памяти ЭВМ и практическая реализация алгоритма работы с графами.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами теории графов:

Графы: определения и примеры. Упорядоченный граф. Представления графов: матрица инцидентностей, матрица смежности, список пар, списки смежности.

Поиск в графе: Основные методы обработки графов. Поиск в ширину. Поиск в глубину.

Связные компоненты: Определение компонент связности. Топологическая сортировка.

Двусвязность: Точки сочленения и их свойства. Алгоритм выделения компонент двусвязности графа.

Эйлеров путь в графе. Алгоритм построения Эйлерова пути.

Гамильтонов путь в графе. Нахождение Гамильтонова пути в графе с помощью алгоритма с возвратом.

Циклы: Фундаментальное множество циклов графа. Алгоритм отыскания фундаментального множества циклов в графе.

Остовные деревья графа: Связные компоненты. Построение и свойства остовных деревьев при поиске в глубину и в ширину.

Остовные деревья минимального веса: Минимальное остовное дерево. Алгоритм Прима. Алгоритм Крускала.

Методические рекомендации

Необходимо дать описание «Графа» как абстрактного типа данных. Затем реализовать основные операции АТД «Граф» в виде процедур или функций, при этом данные в процедуры или функции должны передаваться через формальные параметры.

Перед завершением выполнения программы необходимо убедиться, что вся динамическая память, использованная для создания динамических структур данных, освобождена. В противном случае обеспечить ее полное освобождение.

Контрольные вопросы

1. Опишите, распространенные способы представления графов в памяти ЭВМ.
2. Что такое эйлеров путь.
3. Необходимое и достаточное условие существования эйлерова пути.
4. Что такое гамильтонов путь.
5. Что такое стягивающее дерево.

4.11 Кратчайшие пути в графе

Цель лабораторной работы – получение навыком при решении задач нахождения кратчайшего пути в графе.

Теория

Для выполнения лабораторной работы необходимо ознакомиться со следующими разделами теории графов:

Кратчайшие пути в графе от фиксированной вершины. Алгоритм Форда-Беллмана. Алгоритм Дейкстры. Кратчайшие пути в бесконтурном графе.

Кратчайшие пути между всеми парами вершин: Матрица смежности, матрица достижимости и транзитивное замыкание отношения, алгоритм Уоршалла. Алгоритм Флойда-Уоршалла вычисления расстояний между всеми парами вершин, одновременное построение путей.

Методические рекомендации

Необходимо ввести абстрактный тип данных «Граф». Изучить методы нахождения кратчайших путей в графе. Реализовать основные операции АТД «Граф» в виде процедур или функций, при этом данные в процедуры или функции должны передаваться через формальные параметры.

Перед завершением выполнения программы необходимо убедиться, что вся динамическая память, использованная для создания динамических структур данных, освобождена. В противном случае обеспечить ее полное освобождение.

Контрольные вопросы

1. Поясните, в чем состоит алгоритм Форда-Беллмана.
2. Поясните, в чем состоит алгоритм Дейкстры.
3. На каком принципе строится алгоритм нахождения кратчайших путей в бесконтурном графе?
4. Что такое «транзитивное замыкание».
5. Поясните, в чем состоит алгоритм Уоршалла.

5. Учебно-методическое и информационное обеспечение дисциплины

5.1 Основная литература

1. Горитов А.Н. Основы структур и алгоритмов обработки данных: Учебное пособие. – Томск: ТУСУР, 2007. – 229 с. (50 экз.)

5.2 Дополнительная литература

2. Макконелл Дж. Основы современных алгоритмов. 2-е дополненное издание. – Москва: Техносфера, 2004. – 368 с. (13 экз.)
3. Иванов Б.Н. Дискретная математика. Алгоритмы и программы: Учебное пособие. – М.: Лаборатория базовых знаний, 2003. – 288 с. (50 экз.)
4. Окулов С. М. Программирование в алгоритмах. – 2-е изд., доп. – М.: БИНОМ. Лаборатория знаний, 2006. – 384 с. (30 экз.)
5. Новиков Ф.А. Дискретная математика для программистов: Учебник для вузов. – СПб: Питер, 2002. – 302 с. (19 экз.)
6. Андерсон Д.А. Дискретная математика и комбинаторика. – М.: Издательский дом "Вильямс", 2004. – 960 с. (10 экз.)
7. Липский В. Комбинаторика для программистов. – М.: Мир, 1989. – 213 с. (9 экз.)

8. Вирт Н. Алгоритмы и структуры данных. – М.: Мир, 1989. – 360 с. (50 экз.)
9. Бежанова М.М. Практическое программирование. Структуры данных и алгоритмы. – М.: Логос, 2001. – 224 с. (9 экз.)