

СТРУКТУРЫ ДАННЫХ В ЭВМ

**УЧЕБНАЯ ПРОГРАММА, МЕТОДИЧЕСКИЕ УКАЗАНИЯ
И КОНТРОЛЬНЫЕ ЗАДАНИЯ**

МИНСК 2011
МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«МИНСКИЙ ГОСУДАРСТВЕННЫЙ ВЫСШИЙ
РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

ПОДЛЕЖИТ ВОЗВРАТУ

УТВЕРЖДАЮ
Проректор по учебной работе
_____ П. И. Кирвель
«14» июня 2011 г.

СТРУКТУРЫ ДАННЫХ В ЭВМ

Учебная программа, методические указания
и контрольные задания
для студентов заочного отделения специальности 1-08 01 01-07
«Профессиональное обучение. (Информатика)»

Минск
МГВРК
2011

УДК 681.32(075)
ББК 32.973–018.2я7
С87

Рекомендовано к изданию кафедрой информатики (протокол № 8 от 30.03.2011 г.) и Научно-методическим советом учреждения образования «Минский государственный высший радиотехнический колледж» (протокол № 10 от 18.05.2011 г.)

С о с т а в и т е л ь

Т. М. Тарасова, ассистент кафедры информатики МГВРК

Р е ц е н з е н т

Л. Н. Громова, старший преподаватель
кафедры информатики МГВРК

Структуры данных в ЭВМ : учеб. программа, метод.
С83 указания и контрол. задания для студентов заочного отделения специальности 1-08 01 01-07 «Профессиональное обучение. (Информатика)» / сост. Т. М. Тарасова. – Минск : МГВРК, 2011. – 16 с.
ISBN 978-985-526-114-9

Пособие содержит учебную программу дисциплины, методические указания по выполнению, оформлению контрольной работы, ее варианты и список рекомендуемой литературы.

Предназначено для студентов заочного отделения и преподавателей колледжа.

УДК 681.32(075)
ББК 32.973-018.2я7

ISBN 978-985-526-114-9

© Тарасова Т.М., составление, 2011
© Учреждение образования «Минский государственный высший радиотехнический колледж», 2011

Предисловие

Главная цель дисциплины «Структуры данных в ЭВМ» – формирование у студентов базы знаний по структуре и организации данных ЭВМ.

Основными задачами дисциплины являются:

- изучение принципов логической и физической организации данных на ЭВМ и алгоритмов реализации основных операций над этими структурами;
- систематизация полученных знаний, умений и навыков;
- углубленное изучение таких структур, как списки, деревья, графы и алгоритмов их обработки;
- изучение методов внешней и внутренней сортировки и умение реализовать их на практике;
- изучение комбинаторных алгоритмов;
- формирование системного подхода к выбору наиболее подходящих структур данных при решении конкретных задач в последующих учебных курсах.

Дисциплина «Структуры данных в ЭВМ» опирается на знания и умения студентов, полученные при изучении таких дисциплин, как «Основы алгоритмизации и программирования», «Конструирование программ и языки программирования».

В свою очередь знания, умения и навыки, полученные при изучении дисциплины «Структуры данных в ЭВМ», найдут применение при освоении следующих дисциплин: «Функциональное и логическое программирование», «Технология разработки программного обеспечения», «Методы и алгоритмы принятия решений», а также при реализации задач курсового и дипломного проектирования.

1. Учебная программа дисциплины

1.1. Примерный тематический план

Т а б л и ц а 1

Наименование темы	Количество часов			
	Дневное отделение		Заочное отделение	
	Теория	Практика	Теория	Практика
1. Описание структур данных	4	-	1	-
2. Указатели. Рекурсия и итерация	6	2	1	1
3. Множества и массивы	6	4	1	1
4. Сортировка массивов	4	2	1	1
5. Записи	6	4	1	1
6. Файлы	6	4	1	1
7. Методы сортировки	6	4	1	1
8. Списковые структуры	8	4	1	1
9. Деревья	8	4	1	-
10. Графы. Сети	6	4	1	-
11. Комбинаторные алгоритмы	6	2	-	1
	66	34	10	8
Итого	100		18	

1.2. Структура курса на заочном отделении

В соответствии с действующим учебным планом специальности 1-08 01 01-07 «Профессиональное обучение. (Информатика)» на полный курс изучения дисциплины «Структуры в данных ЭВМ» отводится 18 часов обязательных аудиторных занятий: 10 – лекционных и 8 – лабораторно-практических.

1.3. Содержание дисциплины

Тема 1. Описание структур данных

Введение. Уровни описания структур данных. Функциональная спецификация. Логическое описание. Физическое представление. Примеры описания структур.

Литература [10, с. 12 –28]

Тема 2. Указатели. Рекурсия и итерация

Использование механизма указателей. Работа с указателями в языках программирования. Использование рекурсии и итерации. Алгоритмы, для которых использование рекурсии неэффективно. Алгоритмы с возвратом.

Литература [2, с. 23–32], [4, с. 87–95]

Тема 3. Множества и массивы

Множества. Основные определения. Способы представления. Основные операции с множествами. Использование множеств в языках программирования. Массивы. Основные определения. Операции с элементами массивов. Работа с массивами в языках программирования.

Литература [1, с. 44–53], [10, с. 39–59]

Тема 4. Сортировка массивов

Понятие сортировки. Эффективность сортировок. Сортировка включением, выбором, обменом. Шейкерная сортировка. Улучшенные методы сортировки. Сортировка методом Шелла. Анализ методов.

Литература [3, с. 46–71], [5, с. 19–37]

Тема 5. Записи

Записи. Поля записи. Основные определения. Обработка записей. Записи с вариантами. Использование записей в языках программирования.

Литература [2, с. 70–83], [10, с. 60–74]

Тема 6. Файлы

Файлы. Операции над файлами: установочные, завершающие, специальные, ввод-вывод, перемещение по файлу. Доступ к файлам. Типы файлов: текстовые, типизированные, нетипизированные. Алгоритмы обработки файлов: сортировка слиянием, распределяющая сортировка.

Литература [2, с. 113–136], [10, с. 89–93]

Тема 7. Методы сортировки

Метод сбалансированного слияния. Многофазовое слияние. Двухфазная процедура. Многофазовое слияние в случае чисел Фибоначчи. Каскадное слияние и условия его выполнимости.

Чтение ленты в обратном порядке. Обратное чтение в многофазовом слиянии и в каскадном слиянии. Внешняя поразрядная сортировка.

Литература [3, с. 85–92], [5, с. 113–144]

Тема 8. Списковые структуры

Понятие линейного списка. Операции с линейными списками. Понятие стека, очереди, дека и их использование. Приоритетные очереди. Их представление в виде связанных бинарных деревьев. Понятие отношения порядка. Топологическая сортировка. Двусвязные списки и операции с ними. Кольцевые списки.

Литература [2, с. 140–160], [10, с. 103–119]

Тема 9. Деревья

Основные определения. Способы представления деревьев: вложенные множества, вложенные скобки, уступчатый список, десятичная система Дьюи. Особенности бинарных деревьев. Идеальное сбалансированное дерево, его построение. Обход бинарных деревьев: прямой, обратный, концевой. Поиск, включение, исключение элементов. Сортировка с помощью дерева.

Литература [7, с. 83–95], [10, с. 167–194]

Тема 10. Графы. Сети

Основные определения. Алгоритмы на графах. Машинное представление графов. Матрица смежности. Нахождение кратчайшего пути в графе. Алгоритмы Дейкстры и Флойда. Алгоритм нахождения максимального потока в сети. «Задача коммивояжера».

Литература [6, с. 81–103], [7, с. 54–61]

Тема 11. Комбинаторные алгоритмы

Основные понятия. Комбинаторные алгоритмы: нахождения перестановок, разбиений, подмножеств данного множества.

Литература [6, с. 103–132]

1.4. Примерный перечень практических занятий

1. Работа с указателями в C++.
2. Обработка массивов.
3. Сортировка массивов.
4. Работа с записями.

5. Работа с файлами.
6. Сортировка файлов.
7. Работа со списковыми структурами.
8. Алгоритмы обработки деревьев.
9. Алгоритмы на графах.
10. Комбинаторные алгоритмы.

2. Вопросы для самоконтроля

1. Что такое структура данных?
2. На каких трех уровнях можно описать структуру?
3. Что такое функциональная спецификация?
4. Что такое логическое описание?
5. Что такое физическое представление?
6. Что такое множество?
7. Как можно определить одномерный и многомерный массивы?
8. Что мы понимаем под сортировкой?
9. На какие три группы можно условно разбить все методы сортировки?
10. Что такое сортировка выбором, включением, обменом?
11. Чем отличается сортировка методом прямого включения от сортировки методом двоичного включения?
12. Что такое запись или структура?
13. Частные случаи структур в языке программирования C++.
14. Как задаются константы в перечислениях?
15. Что такое рекурсия?
16. Что такое итерация?
17. Для чего нужны указатели?
18. Какие статические структуры вы знаете?
19. Какие динамические структуры вы знаете?
20. Что такое стек, дек, очередь?
21. Какие бывают списки?
22. Что такое дерево?
23. Чем отличается дерево от двоичного дерева?
24. Как преобразовать дерево в двоичное дерево?
25. Перечислить методы обхода двоичных деревьев.
26. Что такое граф?
27. Чем ориентированный граф отличается от неориентированного?

28. Какие существуют способы машинного представления графов?
29. Что такое файл?
30. Чем отличается последовательный файл от файла прямого доступа?
31. Перечислить основные комбинаторные алгоритмы.

3. Требования, предъявляемые к выполнению контрольной работы

В соответствии с учебным планом студенты должны выполнить одну домашнюю контрольную работу, включающую ответы на два теоретических вопроса и три практических задания.

Перед выполнением работы необходимо изучить теоретический материал по каждой теме и ответить на вопросы для самоконтроля.

Цель контрольной работы – проверка умений студентов пользоваться законспектированным на лекционных занятиях материалом и дополнительной литературой. Ответ на вопрос должен в сжатой конспективной форме содержать информацию по заданной теме.

Излагая ответ на теоретический вопрос, необходимо давать точные формулировки всех определений, полностью раскрывать суть вопроса, конкретно отвечая на него. Объем контрольной работы зависит от объема излагаемого материала.

Вариант контрольной работы определяется последней цифрой шифра студента.

Контрольная работа сдается в срок, установленный в учебном графике.

Контрольная работа выполняется машинописным способом. Работа должна быть подписана с указанием даты ее выполнения, должна иметь поля для замечаний рецензента, страницы необходимо пронумеровать. В конце работы необходимо указать список использованной литературы или названия сайтов Интернета, из которых бралась информация. В конце работы необходимо оставить место для рецензии преподавателя.

Цель выполнения практических заданий контрольной работы – приобретение навыков разработки программ с использова-

нием структурированных типов данных. Результат должен быть представлен в виде листинга программы или на электронном носителе.

Проверенная контрольная работа представляется на экзамен. На экзамене преподаватель может проверить знания не только по билету, но и по вопросам контрольной работы.

Порядок выполнения работы:

1) теоретическая часть: ответить на вопросы 1, 2 со ссылкой на источники информации;

2) практическая часть: разработать программы, соответствующие заданиями 3–5.

4. Контрольная работа

Вариант 0

1. Множества. Множества в Pascal.

2. Граф. Машинное представление графов. Матрица смежности. Нахождение кратчайшего пути в графе.

Написать программы на языке программирования C++ или Pascal.

3. В файле находится текст программы на языке C++. Используя стек, проверить правильность вложенности циклов.

4. Используя двунаправленный список, написать программу, реализующую отдельные действия в игре «Новое домино», где, кроме традиционных правил, можно на каждом ходу заменять конечную кость на свою. Необходимо выполнение следующих функций:

- проверить, можно ли сделать следующий ход;
- сделать ход;
- проверить, можно ли сделать замену;
- заменить кость;
- проверить, закончена ли игра.

5. Реализовать на языке C++ алгоритм нахождения максимального потока в сети.

Вариант 1

1. Уровни описания структур данных.

2. Массивы. Одномерные и двумерные массивы в языке программирования C++.

Написать программы на языке программирования C++ или Pascal.

3. Просматривая текстовый файл, составить программу, формирующую перекрестные ссылки, то есть, печатающую список заданных слов и номера строк, в которых они встречаются. Сформировать из этой информации запись и поместить в список.

4. Разработать программу генерации всех перестановок n -элементного множества в лексикографическом порядке.

Множество – {1.2.3}, перестановки в лексикографическом порядке: (1 2 3), (1 3 2), (2 1 3), (2 3 1), (3 1 2), (3 2 1).

5. Реализовать на языке C++ «Задачу коммивояжера».

Вариант 2

1. Сортировка массивов при помощи вставки, выбора и обмена.

2. Файлы. Работа с файлами в языке программирования C++.

Написать программы на языке программирования C++ или Pascal.

3. В текстовом файле применяются скобки трех типов: (), [], {}. Используя стек, проверить баланс скобок.

4. Разработать программу генерации всех перестановок n -элементного множества в антилексикографическом порядке.

5. Разработать алгоритм обхода двоичных деревьев: нисходящий, восходящий и смешанный.

Вариант 3

1. Комбинаторные алгоритмы. Алгоритмы генерирования перестановок n -элементного множества (в лексикографическом и антилексикографическом порядке).

2. Записи или структуры. Структуры в C++.

Написать программы на языке программирования C++ или Pascal.

3. Используя очередь, проверить, какие строки текстового файла являются симметричными.

4. По графу А построить граф В с тем же множеством вершин, но степень каждой вершины в графе В должна быть на 1 больше чем в графе А.

5. Разработать алгоритм генерации всех перестановок 8-элементного множества за минимальное количество транспозиций. Поместить в кольцо перестановки, начинающиеся с цифры 3.

Вариант 4

1. Сортировка методом сбалансированного слияния, методом многофазового слияния и методом многофазового слияния в случае чисел Фибоначчи.

2. Стеки, очереди. Работа с ними в C++.

Написать программы на языке программирования C++ или Pascal.

3. Решить задачу, используя очередь и стек. Текстовый файл содержит текст, сбалансированный по (). Для каждой пары скобок напечатать номера их позиций в тексте, упорядочив пары по возрастанию номеров столбцов открывающих скобок.

4. Написать программу, которая каждый элемент дерева возводит в квадрат.

5. Разработать алгоритм генерации всех k -элементных подмножеств n -элементного множества.

Вариант 5

1. Распределяющая сортировка, внешняя поразрядная сортировка. Сортировка при помощи каскадного слияния.

2. Деревья. Двоичные деревья. Алгоритмы обхода деревьев. Преобразование дерева в двоичное дерево.

Написать программы на языке программирования C++ или Pascal.

3. Написать функции включения, удаления и чтения элемента стека объемом n -элементов. В случае переполнения стека или попытки удаления из пустого стека вставить флаг, выдать сообщение.

4. Создать числовое дерево. Используя рекурсивную функцию, подсчитать сумму элементов.

5. Разработать алгоритм генерации всех k -элементных подмножеств n -элементного множества в антилексикографическом порядке.

Вариант 6

1. Способы представления деревьев. Вложенные множества, вложенные скобки, уступчатый список, десятичная система Дьюи.

2. Комбинаторные алгоритмы. Генерирование всех подмножеств n -элементного множества. Генерирование всех k -элемент-

ных подмножеств n -элементного множества в лексикографическом и антилексикографическом порядке.

Написать программы на языке программирования C++ или Pascal.

3. Просматривая текстовый файл, составить программу, формирующую перекрестные ссылки, то есть, формирующую список заданных слов и номера строк, в которых они встречаются. Сформировать из этой информации записи и поместить в список. Затем, изменяя набор обрабатываемых слов, скорректировать исходный текстовый файл.

4. Найти кратчайшее расстояние между двумя вершинами в графе. Найти все возможные пути между этими вершинами, не пересекающиеся:

- по ребрам,
- по вершинам.

5. Разработать алгоритм всех разбиений n -элементного множества.

Вариант 7

1. Сети. Поиск в сети. Алгоритмы нахождения минимального и максимального потока в сети.

2. Комбинаторные алгоритмы. Генерирование всех разбиений n -элементного множества.

Написать программы на языке программирования C++ или Pascal.

3. В текстовом файле верно записаны формулы следующего вида:

$\langle \text{формула} \rangle ::= \langle \text{цифра} \rangle ! M(\langle \text{формула} \rangle, \langle \text{формула} \rangle) ! m(\langle \text{формула} \rangle, \langle \text{формула} \rangle) \langle \text{цифра} \rangle = 0!1!2!3!4!5!6!7!8!9$

M – это функция \max , а m – функция \min . Используя стек, вычислить значение данной формулы.

Например: $M(5, m(6,8))=6$.

4. Используя граф, решить следующую задачу. Группа состоит из N человек. Некоторые из них знакомы между собой. Можно ли перезнакомить их опосредованно, то есть только через общих знакомых?

5. Разработать алгоритм генерирования перестановок n -элементного множества за минимальное число транспозиций соседних элементов.

Вариант 8

1. Комбинаторные алгоритмы. Алгоритмы генерирования перестановок n -элементного множества (за минимальное число транспозиций, за минимальное число транспозиций соседних элементов).

2. Линейные односвязные и двусвязные списки. Кольца.

Написать программы на языке программирования C++ или Pascal.

3. Написать функции для включения и исключения из кольца.

4. В заданном графе необходимо определить, существует ли цикл, проходящий по каждому ребру графа только один раз.

5. Разработать алгоритм генерации всех k -элементных подмножеств n -элементного множества в лексикографическом порядке.

Множество – $\{1.2.3\}$, перестановки в лексикографическом порядке: (1 2 3), (1 3 2), (2 1 3), (2 3 1), (3 1 2), (3 2 1).

Вариант 9

1. Граф. Поиск в графе в глубину и в ширину.

2. «Задача коммивояжера».

Написать программы на языке программирования C++ или Pascal.

3. Разработать программу формирования стека, куда помещаются целые числа, вводимые с клавиатуры. Процесс ввода прекращается после ввода отрицательного числа. Затем программа выводит сообщение о том, что введено отрицательное число и отображает содержимое стека на экран в порядке ввода.

4. Группа состоит из N человек. В ней каждый имеет $N/2$ друзей и не более K врагов. У одного из них есть книга, которую все хотят прочитать. Написать программу, реализующую способ передачи книги таким образом, чтобы она, переходя от друга к другу, побывала в руках у каждого не более одного раза и вернулась к владельцу.

5. Разработать алгоритм генерирования перестановок n -элементного множества за минимальное число транспозиций.

Рекомендуемая литература

1. Бородич, Ю. С. Паскаль для персональных компьютеров / Ю. С. Бородич, А. Н. Вальвачев, А. И. Кузьмич. – Минск : Выш. шк., 1991. – 365 с.
2. Касаткин, А. И. От Turbo C к Borland C++ / А. И. Касаткин, А. Н. Вальвачев. – Минск : Выш. шк., 1992. – 239 с.
3. Вирт, Н. Алгоритмы и структуры данных / Н. Вирт. – М. : Мир, 1989. – 360 с.
4. Кнут, Д. Искусство программирования для ЭВМ : в 3 т. / Д. Кнут. – М. : Мир, 1976. – Т 1 : Основные алгоритмы. – 735 с.
5. Кнут, Д. Искусство программирования для ЭВМ : в 3 т. / Д. Кнут. – М. : Мир, 1976. – Т3 : Сортировка и поиск. – 844 с.
6. Липский, В. Комбинаторика для программистов / В. Липский. – М. : Мир, 1988. – 213 с.
7. Лэнгсам, Й. А. Структуры данных для персональных ЭВМ / Й. Лэнгсам, М. Огенстан, А. Тененбаум. – М. : Мир, 1989. – 568 с.
8. Майника, Э. Алгоритмы оптимизации на сетях и графах / Э. Майника. – М. : Мир, 1988. – 335 с.
9. Мейер, Б. Методы программирования / Б. Мейер, К. Бодуэн. – М. : Мир, 1982.
10. Флорес, И. Структуры и управление данными / И. Флорес. – М. : Финансы и статистика, 1982. – 319 с.

Оглавление

Предисловие	3
1. Учебная программа дисциплины	4
1.1. Примерный тематический план	4
1.2. Структура курса на заочном отделении	4
1.3. Содержание дисциплины	4
1.4. Примерный перечень практических занятий	6
2. Вопросы для самоконтроля	7
3. Требования, предъявляемые к выполнению контрольной работы	8
4. Контрольная работа	9
Рекомендуемая литература	14

Учебное издание

СТРУКТУРЫ ДАННЫХ В ЭВМ

Учебная программа, методические указания
и контрольные задания
для студентов заочного отделения
специальности 1-08 01 01-07
«Профессиональное обучение. (Информатика)»

Составитель
Тарасова Татьяна Михайловна

Ответственный за выпуск О. П. Козельская
Редактор Г. Л. Говор
Корректор О. А. Артемчик
Компьютерная верстка В. С. Понтус

Подписано в печать 14.06.2011. Формат 60×84¹/₁₆. Бумага писчая.
Ризография. Усл. печ. л. 0,93. Уч.-изд. л. 0,51. Тираж 60 экз. Заказ 103.

Издатель и полиграфическое исполнение:
учреждение образования «Минский государственный высший
радиотехнический колледж»
ЛИ № 02330/0494033 от 08.01.2009.
Пр. Независимости, 62, 220005, Минск.