

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

АЛГОРИТМЫ И АНАЛИЗ ИХ СЛОЖНОСТИ

Методические указания к практическим занятиям
студентов всех форм обучения для направления 01.04.02 (010400)
"Прикладная математика и информатика"
(квалификация (степень) "магистр")

Томск-2015

Горитов А.Н.

Алгоритмы и анализ их сложности: методические указания к практическим занятиям студентов всех форм обучения по направлению подготовки "010400 Прикладная математика и информатика" (квалификация (степень) "магистр") / А.Н. Горитов. – Томск: ТУСУР, 2015. – 8 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 28 августа 2015 г., протокол № 1

© ТУСУР, каф. АСУ, 2015

© Горитов А.Н. 2015

СОДЕРЖАНИЕ

1. Цели и задачи дисциплины.....	4
2. Методы и формы организации обучения	4
3. Место дисциплины в структуре ООП	5
4. Практические занятия	5
4.1 Вычислительная сложность алгоритмов.....	5
4.2 NP-полные и труднорешаемые задачи	5
4.3 Метод ветвей и границ.....	5
4.4 Поиск подстрок.....	6
4.5 Фундаментальные алгоритмы на графах	6
4.6 Кратчайшие пути во взвешенном графе	6
4.7 Задачи о потоках	7
4.8 Задача о паросочетании	7
4.9 Решение задач с помощью алгоритма с возвратом	7
5. Список литературы	8
5.1 Основная	8
5.2 Дополнительная	8

1. Цели и задачи дисциплины

Целью изучения дисциплины «Алгоритмы и анализ их сложности» 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 деревообразные 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 «Алгоритмы и анализ их сложности» направлен на формирование следующих компетенций:

общекультурные компетенции (ОК):

- способностью к абстрактному мышлению, анализу, синтезу (ОК-1);

общепрофессиональные компетенции (ПК):

- способностью использовать и применять углубленные знания в области прикладной математики и информатики (ОПК-4);

профессиональные компетенции (ПК):

- способностью проводить научные исследования и получать новые научные и прикладные результаты самостоятельно и в составе научного коллектива (ПК-1);
- способностью разрабатывать и анализировать концептуальные и теоретические модели решаемых научных проблем и задач (ПК-2).

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: контроль в ходе выполнения практических заданий, опрос по изучаемым разделам дисциплины, тесты. По всем практическим занятиям подготовлены списки контрольных вопросов.

3. Место дисциплины в структуре ООП

Дисциплина «Алгоритмы и анализ их сложности» относится к вариативной части Блока 1 «Дисциплины (модули)». Успешное овладение дисциплиной предполагает предварительные знания математического анализа, вычислительных методов, дискретной математики в объеме, предусмотренном специальностью «Прикладная математика и информатика», а также навыки программирования на языках высокого уровня.

4. Практические занятия

4.1 Вычислительная сложность алгоритмов

Алгоритмы и их классификация. Вычислительная сложность алгоритмов.
Литература 3, 5.

Контрольные вопросы

1. Какие критерии используются при выборе алгоритмов?
2. Как оценивается трудоемкость алгоритма?
3. Что такое O-нотация и для чего она используется?
4. Какие группы функций можно выделить с помощью O-нотации?
5. Какие рекомендации следует использовать при выборе алгоритмов с помощью O-нотации?

4.2 NP-полные и труднорешаемые задачи

Массовая и индивидуальная задачи. Сложность алгоритма. Полиномиальные алгоритмы и класс P. Недетерминированные алгоритмы и класс NP.
Полиномиальная преобразуемость задач. NP-трудные и NP-полные задачи.
Литература 3, 4, 5.

Контрольные вопросы

1. Как оценивается сложность алгоритма?
2. Чем отличаются алгоритмы класса P от алгоритмов класса NP?
3. Что такое «полиномиальная приводимость задач»?
4. Что такое «Задача принятия решения»?
5. Какая задача является NP-трудной?

4.3 Метод ветвей и границ

Метод декомпозиции. Пример.

Поиск с возвратом. Общий алгоритм. Пример.

Метод ветвей и границ. Применение метода при решении задачи о рюкзаке, задаче о назначениях и задаче коммивояжера.

Динамическое программирование. Пример. Нисходящее и восходящее динамическое программирование. Задача определения наиболее длинной общей подпоследовательности. Задача построения оптимального бинарного дерева поиска.

Литература 1, 3, 4.

Контрольные вопросы

1. В чем заключается метод ветвей и границ.

2. Как строятся оценочные функции состояний? Приведите примеры.
3. Что такое динамическое программирование?
4. В каких случаях применяется динамическое программирование?
5. В чем заключается восходящий и нисходящий методы динамического программирования?

4.4 Поиск подстрок

Задача поиска подстрок. Простейший алгоритм. Алгоритм Кнута-Морриса-Пратта. Алгоритм Бойера-Мура. Алгоритм Рабина-Карпа.
Литература 3, 4

Контрольные вопросы

1. Что такое префикс–функция, ассоциированная с образцом?
2. Что такое эвристика стоп–символа?
3. В чем состоит эвристика безопасного суффикса?
4. Что такое собственный префикс?
5. Что такое собственный суффикс?

4.5 Фундаментальные алгоритмы на графах

Графы: определения и примеры. Представления графов: матрица инциденций, матрица смежности, список пар, списки смежности.

Фундаментальные алгоритмы на графах: Поиск в графе. Поиск в ширину. Поиск в глубину.

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

Эйлеров путь в графе. Алгоритм построения Эйлерова пути.

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

Минимальное остовное дерево. Алгоритм Прима. Алгоритм Крускала.

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

Литература 1, 2, 3, 4, 5, 6, 8, 9, 10.

Контрольные вопросы

1. Какие способы можно использовать для представления графов как структур данных?
2. Какие структуры данных необходимы для реализации списков смежности?
3. Как реализуются операции добавления и удаления ребер в графе?
4. Как реализуются операции добавления и удаления вершин в графе?
5. Какие шаги включает в себя алгоритм поиска в глубину?

4.6 Кратчайшие пути во взвешенном графе

Кратчайшие пути в графе. Кратчайшие пути от фиксированной вершины. Алгоритм Форда-Беллмана. Случай неотрицательных весов: алгоритм Дейкстры. Кратчайшие пути в бесконтурном графе. Топологическая сортировка. 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.

Литература 2, 3, 4, 5, 6, 8, 9, 10.

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 бесконечном графе?
4. Что такое «транзитивное замыкание».
5. Поясните, в чем состоит алгоритм Уоршалла.

4.7 Задачи о потоках

Задача о максимальном потоке. Введение. Транспортные сети и потоки. Алгоритм Форда-Фалкерсона. Алгоритм Эдмондса-Карпа. Алгоритм Диница.

Задача о потоке минимальной стоимости.

Литература 2, 9, 10.

Контрольные вопросы

1. Что такое «Транспортная сеть»?
2. Что такое «Остаточная сеть»?
3. Что такое «Увеличивающий путь»?
4. В чем заключается алгоритм Форда-Фалкерсона?
5. В чем заключается алгоритм Эдмондса-Карпа?

4.8 Задача о паросочетании

Паросочетание. Алгоритм определение максимального паросочетания. Задача о полном паросочетании.

Литература 2, 9, 10.

Контрольные вопросы

1. Что такое «Двудольный граф»?
2. Что такое «Паросочетание»?
3. Что такое «Чередующаяся цепь»?
4. В чем заключается задача о полном паросочетании?
5. Опишите алгоритм решения задачи о полном паросочетании.

4.9 Решение задач с помощью алгоритма с возвратом

Гамильтонов путь. Клики графа.

Литература 6, 8.

Контрольные вопросы

1. В чем заключается алгоритм с возвратом?
2. Как применяется алгоритм с возвратом при нахождении гамильтонового пути в графе?
3. Как применяется алгоритм поиска с возвратом при решении задачи нахождения всех клик графа?

5. Список литературы

5.1 Основная

1. Юрьева А.А. Математическое программирование [Электронный ресурс]: учебное пособие. — СПб. : Лань, 2014. — 432 с. — Режим доступа: http://e.lanbook.com/books/element.php?p11_id=68470.
2. Горлач Б.А. Исследование операций [Электронный ресурс]: учебное пособие. — СПб. : Лань, 2013. — 442 с. — Режим доступа: http://e.lanbook.com/books/element.php?p11_id=4865. Глава 5. Теория графов. Стр. 270 – 323.

5.2 Дополнительная

3. Гагарина Л.Г., Колдаев В.Д. Алгоритмы и структуры данных. – М: Финансы и статистика; ИНФРА-М, 2009. – 304 с.
4. Горитов А.Н. Основы структур и алгоритмов обработки данных: Учебное пособие. – Томск: ТУСУР, 2007. – 229 с. (50 экз.)
5. Макконелл Дж. Основы современных алгоритмов. 2-е дополненное издание. – Москва: Техносфера, 2004. – 368 с. (13 экз.)
6. Иванов Б.Н. Дискретная математика. Алгоритмы и программы: Учебное пособие. – М.: Лаборатория базовых знаний, 2003. – 288 с. (50 экз.)
7. Окулов С. М. Программирование в алгоритмах. – 2-е изд., доп. – М.: БИНОМ. Лаборатория знаний, 2006. – 384 с. (30 экз.)
8. Новиков Ф.А. Дискретная математика для программистов: Учебник для вузов. – СПб: Питер, 2002. – 302 с. (19 экз.)
9. Андерсон Д.А. Дискретная математика и комбинаторика. – М.: Издательский дом "Вильямс", 2004. – 960 с. (10 экз.)
10. Липский В. Комбинаторика для программистов. – М.: Мир, 1989. – 213 с. (9 экз.)