

Программа учебной дисциплины «Алгоритмы и структуры данных»
для образовательной программы Программная инженерия
направления подготовки 09.03.04 – Программная инженерия
Уровень: бакалавр

Утверждена
Академическим советом ООП
Протокол №__ от «_»____2016г.

Автор	Пономаренко Александр Александрович
Число кредитов	8
Контактная работа (час.)	60
Самостоятельная работа (час.)	244
Курс	2
Формат изучения дисциплины	Без использования онлайн курса

I. ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ

Целями освоения дисциплины «Алгоритмы и структуры данных» является изучение студентами основных структур данных, базовых алгоритмов и развитие алгоритмического мышления. В результате освоения дисциплины студент должен:

- знать основные структуры данных (список, дерево, хеш-таблицы, графы), базовые алгоритмы (поиск в глубину, поиск в ширину, принцип разделяй и властвуй, динамическое программирование, поиск с отсечением, генерирование комбинаторных объектов).
- развить алгоритмическое мышление,
- овладеть навыками программирования на языке C++ в плане реализации основных вычислительных алгоритмов и организации структур данных

Изучение данной дисциплины базируется на дисциплинах:

- Дискретная математика
- Основы и методология программирования

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

- Операционные системы
- Конструирование программного обеспечения
- Многопоточное программирование
- Сложность алгоритмов
- Анализ и разработка данных

II. СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Тема 1. Понятие и свойство алгоритма. Способы записи алгоритма. Различные формализации понятия алгоритма.

Определение понятия «алгоритм». Основные свойства алгоритма. Способы записи алгоритмов, их «плюсы» и «минусы». Математическое определение алгоритма (по Колмого-

рову). Тезис Тьюринга-Чёрча. Нормальные алгоритма Маркова. Частично рекурсивные функции.

Тема 2. Сложность алгоритмов. Классы алгоритмов.

Понятие сложности алгоритма. Базовые инструменты оценки сложности. Нотация большого «O». Классификация алгоритмов по сложности. Понятие Машины Тьюринга.

Тема 3. Алгоритмы сортировки.

Алгоритм сортировки пузырьком. Сортировка вставками. Сортировка слиянием. Быстра сортировка. Теоретическая сложность быстрой сортировки. Теорема о сложности оптимального алгоритма, основанного на сравнениях.

Тема 4. Поиск в ширину. Поиск в глубину. Рекурсия.

Алгоритм Ли. Игра пятнашки. Оценка вычислительной сложности рекурсивных алгоритмов.

Тема 5. Основные структуры данных

Списки. Двоичные деревья. Случайные деревья. Терминология и классификация деревьев. Сбалансированные деревья. Примеры использования деревьев.

Тема 6 Хеширование.

Хеширования с цепочками. Хеширования с открытой адресаций. Линейное хеширования, квадратичное, двойное. Консистенное хеширование.

Тема 7. Жадные алгоритмы.

Принцип жадности. Оптимальность подзадач. Матройды – теоретическое основание жадных алгоритмов. Сжатие без потерь. Алгоритм Хафмана.

Тема 8. Построение различных комбинаторных объектов.

Генерация множества всех подмножеств. Генерация сочетаний. Генерация перестановок. Генерация размещений. Задача о рюкзаке. Задача о сумме подмножества.

Тема 9. Принцип Белмана.

Задача о поиске кратчайшего пути в графе. Алгоритм Дейкстры. Псевдополиномиальное решение задачи о рюкзаке.

Тема 10. Динамическое программирование.

Расстояние редактирования. Задача о расстановки скобок. Задача «Взлом сети».

Тема 11. Графы - 1.

Представление графов в памяти компьютера. Матрица смежности, инцидентности, список рёбер. Поиск всех путей длиной k. Поиск кратчайшего пути. Поиск максимального пути.

Тема 12. Графы - 2.

Поиск максимального независимого множества. Вычисление характеристик графа. Задача поиска минимального остова. Проверка графов на изоморфность.

Тема 13. Построение выпуклой оболочки. Вращение геометрических фигур. Элементы компьютерной графики.

Тема 14. Классы NP-complete и NP-hard.

Понятие NP полноты. Классы NP-complete и NP-hard. Задача SAT, 3SAT. Задача о максимальном независимом множестве. Сведение задачи SAT → 3SAT. SAT → Max Clique

Тема 15. Методы эффективного решения NP-полных и NP-трудных задач. Перебор с отсечениями. Метод Branch-and-Price. Метод Branch-and-Cut.

III. ОЦЕНИВАНИЕ

3.1 Оценочные средства для промежуточного/итогового контроля студента

По дисциплине предусмотрен промежуточный и итоговый контроль в форме экзамена. На экзамене студенты получают 3-4 практических задания и должны выбрать из них одно (в соответствии с желаемой оценкой, задания разделены по сложности) и предоставить решение в виде готового алгоритма на одном из языков программирования.

3.2 Порядок формирования оценок по дисциплине

Преподаватель оценивает работу студентов исходя из выполнения ими лабораторных работ, а также по результатам промежуточного и итогового контроля.

Оценка за курс формируется следующим образом:

$$O_{\text{накопленная}_1} = (O_{\text{лаб1}} + O_{\text{лаб2}} + O_{\text{лаб3}} + O_{\text{лаб4}} + O_{\text{лаб5}}) / (\text{максимальный_возможный_балл}) * 10$$

$$O_{\text{промежуточная}_1} = O_{\text{накопленная}_1} * 0.5 + O_{\text{экзамен}_1} * 0.5$$

$$O_{\text{накопленная}_2} = ((O_{\text{лаб1}} + O_{\text{лаб2}} + O_{\text{лаб3}} + O_{\text{лаб4}} + O_{\text{лаб5}}) / (\text{максимальный_возможный_балл}) * 10) * 0,5 + O_{\text{промежуточная}_1} * 0,5$$

$$O_{\text{итоговая}} = O_{\text{накопленная}_2} * 0.5 + O_{\text{экзамен}_2} * 0.5$$

Где $O_{\text{накопленная}_1}$ - накопленная оценка за первую часть курса,

$O_{\text{экзамен}_1}$ - оценка за промежуточный экзамен,

$O_{\text{накопленная}_2}$ - суммарная накопленная оценка за вторую часть курса, учитывающая промежуточный контроль,

$O_{\text{экзамен}_2}$ - оценка за итоговый экзамен,

IV. ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ

4.1 Примеры заданий текущего контроля

Тематика практических заданий

1. Исследование простых алгоритмов и рекурсии.
2. Исследование методов сортировки
3. Задачи на поиск в ширину.
4. Задачи на полный перебор.
5. Исследования различных способов хэширование.
6. Жадные алгоритмы. Алгоритм Хафмана.
7. Динамическое программирование.
8. Алгоритмы на графах
9. Поиск с отсечениями.
10. Программирование алгоритма обратного распространения ошибки.

11. Геометрические задачи. Построение выпуклой оболочки. Вращение объектов.

Задача 1. Хэширование

В программе задаётся размер хэш-таблицы, путь к файлу словаря (русский или английский слов). Она должна считать указанный словарь и добавить каждое слово в словарь. При этом она должна оценивать число возникающих коллизий во время процесса добавления. Это число она должна вывести по завершению процесса добавления. Так же программа должна уметь выполнять поиск слова вводимого через терминал. В результате поиска она должна вывести информацию о том, есть ли такое слово, сколько раз вычислялась хэш-функция и время работы.

Задача 2. Двоичное дерево, работающее с файлом

Программа должна иметь следующие функции: добавление информации по заданному ключу в двоичный файл, вывод информации связанной с заданным ключом, вывод информации связанной с ключами лежащих в заданном диапазоне. Под информацией мы понимаем множество целых чисел. Для упрощения множество целых чисел можно ограничить 10-ю элементами.

Главное требование – программа должна уметь сохранять и считывать информацию о двоичном дереве из файла. Информация в файле может быть представлена в любом формате (можно в двоичном формате, можно в текстовом).

Идея этого задания в том, чтобы получить представление о работе индекса. Так данными связанные с ключами могут быть номера строк в некоторой внешней таблице, значения которой мы индексируем. То есть теперь у нас появляется возможность быстро определить в каких строках это внешней таблице присутствует то или иное значение.

Задача 3. Dream Team.

В файле в первой строчке заданы два числа $n \leq 100000$ и $k \leq 1000$. Во второй строчке идёт последовательность из n чисел $a_i \leq 1000$. Требуется выбрать подмножество A из этих n чисел, чтобы любой элемент из A не был больше суммы двух других элементов из A плюс число k и при этом сумма элементов входящих в A была максимально. Программа должна вывести сумму чисел входящих в A .

Примерный перечень вопросов к промежуточному контролю для самопроверки студентов

1. Какие свойства алгоритма относятся к важнейшим?
2. Принцип Белмана
3. Принцип поиска в ширину.
4. Как можно классифицировать алгоритмы?
5. Что такое сложность алгоритма?
6. Что такое хеширования с открытой адресацией.
7. Что такое хеширование цепочками
8. Как свести задачу SAT к задаче о клики.
9. Связь задачи о независимом множестве и максимальной клики
10. Какой худший случай для алгоритма Quicksort?

4.2 Примеры заданий промежуточной/итоговой аттестации

Задача 1

В файле input.txt задана строка – последовательность символов. Нужно проверить, симметрична ли строка относительно символа X (икс). При этом нужно учитывать только символы латинского алфавита без учёта регистра. Длина строки $\leq 1,000,000$.

Пример входа: bbba8X7aBVb

Задача 2

В файле input.txt есть $n+1$ строка. в каждой пара чисел $[a_i, b_i]$. Это координаты начала и концов интервалов. Нужно проверить если такая точка, которая лежит в $[a_0, b_0]$ и при этом не принадлежит любому другому интервалу. Если нет, вывести "нет". Если есть хотя бы одна точка, то вывести её координаты. Считается, что $n \leq 1,000,000$

Пример входа:

```
0 100
0 49
51 100
```

Задача 3

В файле input.txt в первой строке целое число m

Далее идут n строк из m чисел $\{0,1\}$. 0 - вода. 1 - земля. Пройти можно с земли на землю на 4 смежные клетки. По диагонали ходить нельзя. Посчитать и вывести на экран число островов. $M \leq 1000$, $n \leq 10,000$

Пример входа:

```
8
1 0 1 0 1 0 1 0
1 0 1 0 1 0 1 0
1 0 1 0 1 0 1 0
1 1 1 0 1 1 1 0
1 0 0 0 0 0 1 1
1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 1
1 0 0 0 0 0 0 1
1 1 1 0 0 0 0 1
0 0 1 1 0 0 0 1
0 0 0 1 1 0 0 1
0 0 0 0 1 1 0 1
0 0 0 0 0 1 1 1
```

Задача 4. Написать программу, находящую путь в лабиринте

Задача 5. Написать программу, находящее максимальное число Фибоначчи в диапазоне от миллиона до двух.

Задача 6. Написать программу, расставляющую знаки между числами, так чтобы получить заданное число.

5. РЕСУРСЫ

1. Основная литература

1. Алгоритмы: построение и анализ / Т. Кормен [и др.]. - 2-е изд. - М.: Вильямс, 2012. - 1290 с.
2. [Белов, В.В.](#) Алгоритмы и структуры данных [электронный ресурс]: учебник / В.В.Белов, В.И.Чистякова; Эбс Znanium. - М.:КУРС, НИЦ ИНФРА-М, 2016. - 240 с. – Режим доступа: <http://znanium.com/bookread2.php?book=551224>. – Загл. с экрана.

2. Дополнительная литература

1. [Пруцков, А.В.](#) Математическая логика и теория алгоритмов [Электронный ресурс]: учебник / А.В.Пруцков, Л.Л.Волкова; ЭБС Znanium. - М.:КУРС, НИЦ ИНФРА-М, 2016. - 152 с. – Режим доступа: <http://znanium.com/bookread2.php?book=558694>. – Загл. с экрана.
2. [Колдаев, В.Д.](#) Структуры и алгоритмы обработки данных [Электронный ресурс]: учебное пособие / В.Д.Колдаев; ЭБС Znanium. - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2014. - 296 с. - (Высшее образование: Бакалавриат). – Режим доступа: <http://znanium.com/bookread2.php?book=418290>. – Загл. с экрана. Гриф УМО
3. Алексеев, В.Е. Графы и алгоритмы. Структуры данных. Модели вычислений: учебник / В.Е.Алексеев, В.А.Таланов. - М.: БИНОМ, 2012. - 319 с. - (Основы информационных технологий). Гриф МО РФ

3. Программное обеспечение

№ п/п	Наименование	Условия доступа
1.	Microsoft Windows 7 Professional RUS Microsoft Windows 10 Microsoft Windows 8.1 Professional RUS	<i>Из внутренней сети университета (договор)</i>
2.	Microsoft Office Professional Plus 2010	<i>Из внутренней сети университета (договор)</i>
3.	Microsoft Visual Studio Community Edition	<i>Свободно распространяемое ПО</i>

4. Профессиональные базы данных, информационные справочные системы, интернет-ресурсы (электронные образовательные ресурсы)

№ п/п	Наименование	Условия доступа
	<i>Профессиональные базы данных, информационно-справочные системы</i>	
1.	Электронные образовательные ресурсы	<i>Договор на использование электронных баз данных/по подключению и обеспечению доступа к базам данных</i>

5. Материально-техническое обеспечение дисциплины

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

- ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);
- мультимедийный проектор с дистанционным управлением.

Также аудитория оборудована маркерной доской.