

## Программа учебной дисциплины «Алгоритмы и структуры данных»

Утверждена

Академическим советом ООП

Протокол № 8.1.2.1-54/02 от «27 » июня 2016 г.

Автор	Шутов А.А. , ст. преподаватель
Число кредитов	6
Контактная работа (час.)	80
Самостоятельная работа (час.)	148
Курс	2
Формат изучения дисциплины	без использования онлайн курса

### I. ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ

Целями освоения дисциплины «Алгоритмы и структуры данных» являются углубленное изучение основ алгоритмизации и структур данных, овладение методами разработки и описания различных алгоритмов, связанных с управлением данными и применение полученных знаний для работы в избранной сфере деятельности, обладать универсальными и предметно-специализированными компетенциями, способствующими его социальной мобильности и устойчивости на рынке труда.

В результате освоения дисциплины студент должен:

**знать:**

- основные алгоритмы поиска, сортировки и сжатия, знать основные структуры данных, применяемые при разработке программного обеспечения и операции над ними;

**уметь:**

- применять полученные знания к решению вопросов проектирования и разработки программного обеспечения связанного с обработкой данных;

**владеть:**

- навыками использования изученных методов при решении научно-технических задач.

Изучение дисциплины «Алгоритмы и структуры данных» базируется на следующих дисциплинах:

- Программирование;
- Теоретические основы информатики.

Для освоения дисциплины студент должен владеть современными методами и средствами программирования.

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

- 1 Управление качеством программного обеспечения;
- 2 Информационные процессы, системы и сети.

### II. СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

#### Раздел 1. Основы структуры данных.

##### Тема 1.1 Введение, базовые структуры данных

Разложение действия на элементарные части (структура действия). Порождение структуры операндов структурой действия.

Рекурсия как средство повышения эффективности программирования и определяемая ею собственная структура операндов (векторы, матрицы и др., примеры структур).

Структура алгоритмов и структура данных. Связь с математическим понятием структуры. Графический образ структуры.

Переменные величины и схемы структур. Значения переменных структур и экземпляры схем.

Элементы структуры, имена, значения. Основные и вспомогательные базисные множества и отношения в структуре.

Структура машинной памяти. Примеры структур хранения данных. Вектор памяти. Массивы. Адресная арифметика как средство задания отношений в структуре хранения. Структуры хранения, операции над структурами и типы.

### **Тема 1.2 Динамические структуры данных**

Переработка информации как преобразование структур данных. Преобразования, приводящие к рекурсивным отношениям исходных и результирующих структур.

Динамические структуры - класс структур с частичным упорядочением (по включению) структур данных, примеры динамических структур (стеки, очереди, списки).

Массивы.

Динамические структуры и распределение памяти; средства поддержания динамической структуры. Выражение отношений программными средствами. Пример: структура типа стека и ее структура хранения.

Сравнение структур хранения и хранения динамических структур.

Статическое и динамическое распределение памяти. Управление памятью.

### **Тема 2.1 Внутренние сортировки**

Алгоритмы сортировок данных для внутренней памяти.

Типы сортировок, анализ скорости работы сортировок, используемая память.

Сравнение различных сортировок.

### **Раздел 2. Сортировки.**

#### **Тема 2.2 Внешние сортировки**

Алгоритмы сортировок данных для внешней памяти.

Типы сортировок, анализ скорости работы сортировок, используемая память.

Сравнение различных сортировок.

#### **Тема 3.1 Алгоритмы поиска во внутренней памяти**

Понятие дерева поиска. Выполнение операций поиска, включения и исключения записей. Возможность выполнения операций без перепакетки памяти.

Оценка эффективности выполнения операций. Анализ балансировки деревьев (возможность вырождения дерева поиска в линейный упорядоченный список). Сбалансированные и идеально сбалансированные деревья.

Алгоритм вставки с сохранением балансировки дерева поиска.

Красно-черные деревья, цифровые деревья поиска.

Хэш-таблицы.

### **Раздел 3. Алгоритмы поиска.**

#### **Тема 3.2 Алгоритмы поиска во внешней памяти**

Понятие таблицы (ключ, тело, запись). Таблицы имен и адресов.

Операции над таблицей: поиск, включение и исключение записей. Таблица как динамическая структура данных.

Применение хэш-таблиц для поиска во внешней памяти.

Применение деревьев для поиска во внешней памяти.

### **Раздел 4. Алгоритмы сжатия.**

#### **Тема 4.1 Алгоритмы сжатия без потерь**

Применение сжатия данных, классификация алгоритмов.

Алгоритмы Хаффмана и LZW.

### **Раздел 5. Примеры конкурсных задач.**

#### **Тема 5.1 Примеры конкурсных задач**

Анализ задач связанных со структурами данных и алгоритмами, применяемых на различных курсах и экзаменах.

### III. ОЦЕНИВАНИЕ

Преподаватель оценивает работу студентов на практических занятиях: активность в дискуссиях, качество и креативность предлагаемых решений, полноту ответов. Оценки за работу на практических занятиях преподаватель выставляет в рабочую ведомость. Накопленная оценка по 10-ти балльной шкале за работу на практических занятиях определяется перед промежуточным или итоговым контролем -  $O_{аудиторная}$ .

Преподаватель оценивает самостоятельную работу студентов: правильность выполнения, соответствие работы заданию, полнота рассмотрения проблемы. Оценки за самостоятельную работу студента преподаватель выставляет в рабочую ведомость. Накопленная оценка по 10-ти балльной шкале за самостоятельную работу определяется перед итоговым контролем –  $O_{сам. работа}$ .

Накопленная оценка за текущий контроль учитывает результаты студента по текущему контролю следующим образом:

$$O_{накопленная} = 0,4 * O_{ауд} + 0,6 * O_{сам. работа}$$

Результирующая оценка за дисциплину рассчитывается следующим образом:

$$O_{результ} = 0,6 * O_{накопл} + 0,4 * O_{экз}$$

Способ округления накопленной оценки итогового контроля - арифметический.

Оценка за итоговый контроль не является блокирующей.

### IV. ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ

#### Оценочные средства для текущего контроля студента

*Тематика лабораторных работ:*

1. Реализация динамической структуры данных,
2. Реализация представления числа в двоичном виде
3. Реализация алгоритмов сортировки для динамических структур данных, оценка сложности алгоритма, сравнение с другими алгоритмами
4. Реализация структуры данных «сбалансированное дерево», реализация операций вставки/удаления узлов, анализ производительности, сравнение сложности алгоритма.
5. Реализация алгоритма сжатия Хаффмана, оценка скорости работы.
6. Различные задачи на описание алгоритмов и оценку их сложности.

Варианты лабораторной работы для каждого студента утверждаются преподавателем в индивидуальном порядке.

#### Оценочные средства для промежуточной аттестации

*Примерный перечень вопросов к экзамену по всему курсу.*

1. Какими параметрами характеризуется скорость работы сортировки (написать)
2. Минимальное количество перестановок для пузырьковой сортировки
3. В чем разница между внешней и внутренней сортировками.
4. Принцип сортировки слиянием
5. Напишите на языке C++ функцию, которая не рекурсивно считает факториал.

6. Почему желательно использовать итерационные алгоритмы, а не рекурсивные при реализации
7. Определение хэш-функции
8. Как будет вести себя программа, использующая поиск с хешированием, если из-за ошибки программирования значением хеш-функции всегда будет одна и та же константа?
9. Можно ли использовать хеширование в случае хранения данных на внешнем устройстве?
10. Почему требуется, чтобы хеш-функция принимала все значения из множества индексов  $I$ ? Чем плохо, если она будет принимать, например, только четные значения?
11. Как можно реализовать удаление ключа из хешируемой таблицы? Для каких способов разрешения коллизий это сделать легче?
12. Почему значения функции двойного хеширования  $hh(x)$  должны быть взаимно просты с размером хеш-таблицы?
13. Разместить в хеш-таблице из 11 элементов следующие ключи: 60, 24, 77, 126, 100, 239, 2, 93. Использовать алгоритм деления и алгоритм линейных проб.
14. Напишите процедуру поиска трех минимальных элементов массива за один проход.
15. Напишите процедуру поиска  $k$ -й порядковой статистики при помощи неполного алгоритма QuickSort.
16. Выполните поиск медианы в массиве  $A = (48, 72, 3, 14, 35, 65, 88, 89, 95, 6, 5, 65, 21, 24, 77)$ , используя неполный алгоритм QuickSort. В качестве разделяющего использовать первый элемент массива.
17. Дан массив дат  $A$ . Для упрощения принято, что номер года и номер дня лежат в пределах от 01 до 10. Выполнить поразрядную сортировку массива. Показать результаты каждого прохода.  $A = (06.12.03; 09.03.07; 01.01.01; 01.05.01; 10.10.10; 02.05.10; 06.01.03; 07.04.05; 07.11.05; 09.12.06; 09.12.02; 05.12.03)$ .
18. Докажите, что длины всех ветвей ИС-дерева различаются не более, чем на 1.
19. Можно ли при удалении вершины дерева поиска, имеющей двух сыновей, вместо ближайшей вершины слева использовать ближайшую справа?
20. Докажите, что число листьев для наихудших АВЛ-деревьев при увеличении высоты дерева образует последовательность Фибоначчи.
21. Сформулируйте алгоритм вставки в  $B^+$ -дерево
22. Известна модификация страничных деревьев поиска, называемая  $B^{++}$ -деревьями. Для этих деревьев каждая страница, кроме корня, заполнена не менее, чем на  $3/4$ . Объясните, как должна работать вставка в  $B^{++}$ -дерево и сколько ключей может содержать корень такого дерева.
23. Чем отличается операция изменения значения ключевого поля записи базы данных от изменения неключевого поля?
24. Сколько сравнений выполняется при сортировке массива из  $n$  элементов по алгоритму пузырька?
25. При каком условии алгоритм пузырька может завершить сортировку массива за  $n-2$  прохода? за  $n-3$  прохода? На сколько операций сравнения меньше будет выполнено в этих случаях?
26. Предположим, что в алгоритме вставок значение переменной  $r$  сравнивается со значениями элементов массива не в обратном порядке ( $a_k, a_{k-1}, a_{k-2} \dots, a_1$ ), а в прямом ( $a_1, a_2, a_3 \dots, a_k$ ). Повлияет ли это на правильность и эффективность алгоритма? Какой из двух вариантов больше подходит для сортировки Шелла?
27. Объясните, почему в нерекурсивном варианте QuickSort при занесении в стек более длинных отрезков глубина стека оказывается меньше, чем при занесении более коротких.

28. Объясните, почему в рекурсивном варианте QuickSort глубина используемого стека оказывается значительно больше, чем в нерекурсивном.
29. Запрограммируйте смешанный вариант QuickSort, в котором разделение массива и сортировка меньшего из получившихся отрезков выполняется в цикле, а для сортировки большего отрезка используется рекурсивный вызов (при этом нет необходимости явно использовать переменную-стек).
30. Какая глубина стека может потребоваться для реализации QuickSort, описанной в предыдущем упражнении?
31. Алгоритм сортировки называется устойчивым, если элементы массива, имеющие одно и то же значение ключа, сохраняют после сортировки свое взаимное положение. Какие из рассмотренных в разделе алгоритмов являются устойчивыми?
32. Дан массив чисел  $A = (10, 50, 30, 32, 11, 40, 20, 5, 16, 37, 12, 1)$ . Выполнить сортировку массива по алгоритму ShellSort, используя значения  $h = 7, 3, 1$ . Показать состояние массива после каждого прохода.
33. Дан массив чисел  $A = (20, 13, 5, 25, 16, 18, 40, 32, 21, 11, 1, 30)$ . Выполнить сортировку массива по алгоритму QuickSort. В качестве разделяющего выбирать первый элемент отрезка. Показать состояние массива после каждой операции разделения.
34. Дан массив чисел  $A = (35, 8, 24, 12, 42, 15, 31, 40, 14, 50)$ . Выполнить преобразование массива в пирамиду (1-я фаза алгоритма HeapSort) и два первых прохода второй фазы алгоритма. Показать состояние массива после каждой операции просеивания.
35. Как следует изменить алгоритм барьерного поиска, если свободная позиция находится не в конце, а в начале массива?
36. Есть такая математическая игра. Один человек задумывает число от 1 до 1 000, а другой должен определить это число, задав десять вопросов, на которые первый отвечает «Да» или «Нет». Какие вопросы следует задавать? Сколько потребуется вопросов, если задумано число от 1 до 1 000 000?
37. Дан массив целых чисел  $A = (-5, -2, 3, 8, 12, 12, 15, 20, 30, 35, 40, 41, 41, 49, 50)$ . Выполните вручную алгоритм бинарного поиска для ключа  $x = 12$  и для ключа  $x = 42$ , выписывая значения  $i, j, q, A[q]$ .
38. Приведите пример “плохого” изображения для алгоритма Хаффмана.

## V. РЕСУРСЫ

### 5.1 Основная литература

1. Белов, В.В. Алгоритмы и структуры данных [Электронный ресурс]: учебник / В.В.Белов, В.И.Чистякова; ЭБС Znanium. - М.:КУРС, НИЦ ИНФРА-М, 2017. - 240 с. – Режим доступа: <http://znanium.com/bookread2.php?book=766771>. – Загл. с экрана.

### 5.2 Дополнительная литература

1. 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
- 2.
3. 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

### 5.3 Программное обеспечение

№ п/п	Наименование	Условия доступа
1.	Microsoft Office 2013 Prof +	<i>Государственный контракт</i>

### 5.4 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.	Электронно-библиотечная система Юрайт	URL: <a href="https://biblio-online.ru/">https://biblio-online.ru/</a>
	<i>Интернет-ресурсы (электронные образовательные ресурсы)</i>	
1.	Открытое образование	URL: <a href="https://openedu.ru/">https://openedu.ru/</a>

### 5.5 Материально-техническое обеспечение дисциплины

Для выполнения и демонстрации лабораторных работ предполагается использовать ресурсы вычислительного кластера НИУ ВШЭ – Нижний Новгород. Для лекций предполагается использование проектора, экрана, ноутбука. Изучение дисциплины частично основано на использовании ресурсов LMS.