

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
Факультет математики и компьютерных наук

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

### **Алгоритмы и анализ сложности**

Кафедра дискретной математики  
факультет математики и компьютерных наук

Образовательная программа  
02.03.02 Фундаментальная информатика и информационные технологии

Профиль подготовки  
Информатика и компьютерные науки

Уровень высшего образования  
бакалавриат

Форма обучения  
очная

Статус дисциплины: входит в обязательную часть ОПОП

Рабочая программа дисциплины «Алгоритмы и анализ сложности» составлена в 2021 году в соответствии с требованиями ФГОС ВО по направлению подготовки 02.03.02 – Фундаментальная информатика и информационные технологии

Приказ Минобрнауки России от 23.08.2017 №808

Разработчик: кафедра дискретной математики,  
Ибрагимов Мурад Гаджиевич, к. ф.-м. н., доцент.

Рабочая программа дисциплины «Алгоритмы и анализ сложности» одобрена: на заседании кафедры дискретной математики от « 27 » сентября 2021 г., протокол № 1.

Зав. кафедрой Мвб Магомедов А.М.

и на заседании Методической комиссии факультета математики и компьютерных наук от « 29 » сентября 2021 г., протокол № 1.

Председатель М.К. Ризаев Ризаев М.К.

Рабочая программа дисциплины согласована с учебно-методическим управлением « 29 » 09 2021 г.

/Начальник УМУ А.Г. Гасангаджиева Гасангаджиева А.Г.

## Аннотация рабочей программы дисциплины

Дисциплина «Алгоритмы и анализ сложности» входит в обязательную часть образовательной программы бакалавриата по направлению 02.03.02 - Фундаментальная информатика и информационные технологии.

Дисциплина реализуется на факультете математики и компьютерных наук кафедрой дискретной математики.

Содержание дисциплины охватывает круг вопросов, связанных с формированием и развитием у студентов общепрофессиональных и профессиональных компетенций, позволяющих им на базе освоенных теоретических и практических основ математического аппарата осуществлять профессиональную деятельность.

Дисциплина нацелена на формирование следующих компетенций выпускника: общепрофессиональных – ОПК-3, профессиональных – ПК-4.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: лекции, практические занятия, самостоятельная работа.

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме контрольной работы, коллоквиума и итогового экзамена.

Объем дисциплины 5 зачетных единиц, в том числе в академических часах по видам учебных занятий 180 часов.

Объем дисциплины в очной форме

Семестр	Учебные занятия						СРС, в том числе экз.	Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен)	
	в том числе								
	Всего	Контактная работа обучающихся с преподавателем				КСР			СРС, в том числе экз.
		Всего	из них						
Лекции	Лабораторные занятия		Практические занятия	консультации					
3	180	64	32	0	32		80+36	экзамен	

## 1. Цели освоения дисциплины

Целями освоения дисциплины «Алгоритмы и анализ сложности» являются: ознакомление студентов с фундаментальными алгоритмами обработки данных; овладение современным математическим аппаратом для дальнейшего использования в прикладных исследованиях; освоение методов исследования алгоритмов и оценки их алгоритмической сложности.

## 2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина «Алгоритмы и анализ сложности» входит в обязательную часть ОПОП, по направлению 02.03.02-Фундаментальная информатика и информационные технологии.

Для успешного освоения курса «Алгоритмы и анализ сложности» обучающемуся необходимы следующие знания и навыки:

- Дискретная математика. Знакомство с понятиями выборки, перестановки, сочетания, графами. Навыки в представлении и перечислении графов, оценки числа неизоморфных графов с  $q$  ребрами; укладки графов;
- Программирование. Знакомство с понятиями информации, алгоритма для ЭВМ, структуры данных, рекурсивных и итерационных алгоритмов обработки данных. Практические навыки в конструировании алгоритмов, использовании простейших алгоритмов обработки данных при написании более сложных вычислительных программ.

Освоение дисциплины «Алгоритмы и анализ сложности» необходимо при последующем изучении дисциплин «Программная инженерия» «Интеллектуальные системы», «Архитектура вычислительных систем» и ряда других. Дисциплина изучается на 2 курсе во 3 семестре.

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения)

Код и наименование компетенции из ОПОП	Код и наименование индикатора достижения компетенций	Планируемые результаты обучения	Процедура освоения
ОПК-3 Способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных	ОПК-3.1. Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в области математических, информационных и имитационных моделей. ОПК-3.2. Умеет соотносить	Знает: детально методы и базовые алгоритмы обработки информационных структур, методов анализа сложности алгоритмов; Умеет: использовать свои знания на практике; Владеет: основными методами, способами и средствами получения, хранения,	Конспектирование и проработка лекционного материала. Участие в практических занятиях. Самостоятельная работа.

<p>сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям</p>	<p>знания в области программирования, интернет прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем. ОПК-3.3. Имеет практический опыт применения разработки программного обеспечения.</p>	<p>переработки информации, иметь навыки работы с компьютером как средством управления информацией; пониманием концепций, синтаксической и семантической организации, методов использования современных языков программирования; пониманием концепций, базовых алгоритмов, принципов разработки и функционирования современных операционных систем.</p>	
<p>ПК-4. Способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии.</p>	<p>ПК-4.1. Знает современные языки программирования и методы параллельной обработки данных. Знаком с содержанием Единого Реестра Российских программ для электронных вычислительных машин и баз данных. ПК-4.2. Умеет реализовывать численные методы решения прикладных задач в профессиональной сфере деятельности, пакеты программного обеспечения, операционные системы, электронные библиотеки, сетевые</p>	<p>Знает: этапы подготовки программ, подробную структуру программы, простые и структурированные данные, управляющие структуры. Умеет составлять программы средней сложности, воплощать в исполняемые приложения простые базовые алгоритмы. Владеет: навыками компиляции, отладки и тестирования программ</p>	<p>Конспектирование и проработка лекционного материала. Участие в практических занятиях. Самостоятельная работа.</p>

	технологии. ПК-4.3. Имеет практический опыт разработки интеграции информационных систем.		
--	--	--	--

#### 4. Объем, структура и содержание дисциплины

4.1. Объем дисциплины составляет **5 зачетных единиц, 180 академических часов**

#### 4.2. Структура дисциплины

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					Формы текущего контроля
				Всего	Лекции	Практич. занятия	СРС	КСР	
1	<b>Модуль 1. Введение в теорию алгоритмов.</b>								
2	<b>Тема 1.</b> Введение в теорию алгоритмов. Понятие алгоритма. Основные требования к алгоритмам. Машина Тьюринга. Универсальная машина Тьюринга. Тезис Тьюринга.	3	1-2	18	4	4	10		Устный опрос, письменная контрольная работа
3	<b>Тема 2.</b> Граф машина. Алгоритмическая модель языка GRAPH. Модель данных. Предметная область программирования. Построение моделей алгоритмов в системе GRAPH.	3	3-5	18	6	6	6		
4	<b>Итого по модулю 1:</b>	<b>3</b>	<b>1-5</b>	<b>36</b>	<b>10</b>	<b>10</b>	<b>16</b>		Коллоквиум
5	<b>Модуль 2. Сложность алгоритмов.</b>								
6	<b>Тема 3.</b> Временная и пространственная сложность алгоритма. Класс DTIME и DSPACE. Классы сложности задач. Полиномиальность и эффективность. Алгоритмическая	3	6-7	18	4	4	10		Устный опрос,

	сводимость задач.								письменная контрольная работа
7	<b>Тема 4.</b> Представление абстрактных объектов (последовательностей). Смежное представление последовательностей. Связанное представление последовательностей. Операции добавления и удаления элементов списка. Характеристические векторы. Списки. Деревья. Использование списков для разработки алгоритма «Крестики-нолики».	3	8-10	18	6	6	6		
8	<b>Итого по модулю 2:</b>	<b>3</b>	<b>6-10</b>	<b>36</b>	<b>10</b>	<b>10</b>	<b>16</b>		Коллоквиум
9	<b>Модуль 3. Сортировка и поиск.</b>								
10	<b>Тема 5.</b> Сортировка вставками. Инверсии. Сортировка всплытия Флойда. Двоичные деревья на смежной памяти. Процедура всплытия Флойда на бинарном дереве.	3	11-12		4	4	10		Устный опрос, письменная контрольная работа
11	<b>Тема 6.</b> Задачи поиска. Последовательный поиск. Логарифмический поиск. Сортировка с вычисляемыми адресами.	3	13-14	18	4	4	10		
12	<b>Итого по модулю 3:</b>	<b>3</b>	<b>11-14</b>	<b>36</b>	<b>8</b>	<b>8</b>	<b>20</b>		Коллоквиум
13	<b>Модуль 4. Эффективность методов оптимизации.</b>								
14	<b>Тема 7.</b> Унимодальные, непрерывные одномерные функции. Оптимизация многоэкстремальных функций. Сложность выпуклых экстремальных задач.	3	15-16	36	4	4	28		Устный опрос, письменная контрольная работа
15	<b>Итого по модулю 4:</b>	<b>3</b>	<b>15-16</b>	<b>36</b>	<b>4</b>	<b>4</b>	<b>28</b>		Коллоквиум
16	<b>Модуль 5. Подготовка к экзамену</b>								
17	Подготовка к экзамену	3	17	36			36		Экзамен
18	<b>Итого по модулю 5:</b>	<b>3</b>	<b>17</b>	<b>36</b>			<b>36</b>		Экзамен
19	<b>Итого за 3 семестр:</b>	<b>3</b>	<b>1-17</b>	<b>180</b>	<b>32</b>	<b>32</b>	<b>116</b>		Экзамен
20	<b>Итого:</b>	<b>3</b>	<b>1-17</b>	<b>180</b>	<b>32</b>	<b>32</b>	<b>116</b>		Экзамен

## **4.3. Содержание дисциплины, структурированное по темам (разделам)**

### **4.3.1. Содержание лекционных занятий по дисциплине**

#### **3 семестр**

##### **Модуль 1. Введение в теорию алгоритмов**

**Лекция 1.** Введение в теорию алгоритмов. Понятие алгоритма. Основные требования к алгоритмам.

**Лекция 2.** Машина Тьюринга. Универсальная машина Тьюринга. Тезис Тьюринга.

**Лекция 3.** Граф машина. Алгоритмическая модель языка GRAPH.

**Лекция 4.** Модель данных. Предметная область программирования.

**Лекция 5.** Построение моделей алгоритмов в системе GRAPH.

##### **Модуль 2. Сложность алгоритмов**

**Лекция 6.** Временная и пространственная сложность алгоритма. Класс DTIME и DSPACE. Классы сложности задач.

**Лекция 7.** Полиномиальность и эффективность. Алгоритмическая сводимость задач.

**Лекция 8.** Представление абстрактных объектов (последовательностей). Смежное представление последовательностей. Связанное представление последовательностей.

**Лекция 9.** Операции добавления и удаления элементов списка. Характеристические векторы.

**Лекция 10.** Списки. Деревья. Использование списков для разработки алгоритма «Крестики-нолики».

##### **Модуль 3. Сортировка и поиск**

**Лекция 11.** Сортировка вставками. Инверсии. Сортировка всплытия Флойда. Двоичные деревья на смежной памяти.

**Лекция 12.** Процедура всплытия Флойда на бинарном дереве.

**Лекция 13.** Задачи поиска. Последовательный поиск. Логарифмический поиск.

**Лекция 14.** Сортировка с вычисляемыми адресами.

##### **Модуль 4. Эффективность методов оптимизации**

**Лекция 15.** Унимодальные, непрерывные одномерные функции.

**Лекция 16.** Оптимизации многоэкстремальных функций. Сложность выпуклых экстремальных задач.

##### **Модуль 5. Подготовка к экзамену**



## 4.3.2. Содержание лабораторно-практических занятий по дисциплине

### 3 семестр

#### Модуль 1. Введение в теорию алгоритмов

**Занятие 1.** Введение в теорию алгоритмов. Понятие алгоритма. Основные требования к алгоритмам.

**Занятие 2.** Машина Тьюринга. Универсальная машина Тьюринга. Тезис Тьюринга.

**Занятие 3.** Граф машина. Алгоритмическая модель языка GRAPH.

**Занятие 4.** Модель данных. Предметная область программирования.

**Занятие 5.** Построение моделей алгоритмов в системе GRAPH.

#### Модуль 2. Сложность алгоритмов

**Занятие 6.** Временная и пространственная сложность алгоритма. Класс DTIME и DSPACE. Классы сложности задач.

**Занятие 7.** Полиномиальность и эффективность. Алгоритмическая сводимость задач.

**Занятие 8.** Представление абстрактных объектов (последовательностей). Смежное представление последовательностей. Связанное представление последовательностей.

**Занятие 9.** Операции добавления и удаления элементов списка. Характеристические векторы.

**Занятие 10.** Списки. Деревья. Использование списков для разработки алгоритма «Крестики-нолики».

#### Модуль 3. Сортировка и поиск

**Занятие 11.** Сортировка вставками. Инверсии. Сортировка всплытия Флойда. Двоичные деревья на смежной памяти.

**Занятие 12.** Процедура всплытия Флойда на бинарном дереве.

**Занятие 13.** Задачи поиска. Последовательный поиск. Логарифмический поиск.

**Занятие 14.** Сортировка с вычисляемыми адресами.

#### Модуль 4. Эффективность методов оптимизации

**Занятие 15.** Унимодальные, непрерывные одномерные функции.

**Занятие 16.** Оптимизации многоэкстремальных функций. Сложность выпуклых экстремальных задач.

#### Модуль 5. Подготовка к экзамену

### 5. Образовательные технологии

В ходе освоения дисциплины предусматривается применение следующих активных методов обучения

1. Выполнение практических заданий с элементами исследования.
2. Отчетные занятия по разделам.

3. Выполнение студентами индивидуальной исследовательской работы по анализу заданий с поиском и выбором метода их решения.
4. Разбор конкретных заданий.

## **6. Учебно-методическое обеспечение самостоятельной работы студентов**

### **6.1. Примерное распределение времени самостоятельной работы студентов**

Вид самостоятельной работы	Примерная трудоёмкость, а.ч.
Введение в теорию алгоритмов. Понятие алгоритма. Основные требования к алгоритмам. Машина Тьюринга. Универсальная машина Тьюринга. Тезис Тьюринга.	10
Граф машина. Алгоритмическая модель языка GRAPH. Модель данных. Предметная область программирования. Построение моделей алгоритмов в системе GRAPH.	6
Временная и пространственная сложность алгоритма. Класс DTIME и DSPACE. Классы сложности задач. Полиномиальность и эффективность. Алгоритмическая сводимость задач.	10
Представление абстрактных объектов (последовательностей). Смежное представление последовательностей. Связанное представление последовательностей. Операции добавления и удаления элементов списка. Характеристические векторы. Списки. Деревья. Использование списков для разработки алгоритма «Крестики-нолики».	6
Сортировка вставками. Инверсии. Сортировка всплытия Флойда. Двоичные деревья на смежной памяти. Процедура всплытия Флойда на бинарном дереве.	10
Задачи поиска. Последовательный поиск. Логарифмический поиск. Сортировка с вычисляемыми адресами.	10
Унимодальные, непрерывные одномерные функции. Оптимизация многоэкстремальных функций. Сложность выпуклых экстремальных задач.	
Подготовка к экзамену	36
Итого СРС:	116

### **6.2. Виды и порядок выполнения самостоятельной работы**

1. Изучение лекционных материалов (электронные варианты) и рекомендованной литературы.
2. Выполнение индивидуальных заданий на составление программ и подготовка к отчету по ним.
3. Решение задач и упражнений, сформулированных в электронных приложениях к лекции
4. Подготовка к текущему и промежуточному контролю.
5. Подготовка к экзамену.

### **6.3. Порядок контроля:**

1. Блиц-опрос на лабораторных занятиях, 2. Проверка выполнения пакета заданий и прием отчета по ним, 3. Текущий контроль за выполнением задач,

сформулированных в электронных вариантах к лекции, 4. Промежуточный отчет (коллоквиумы, к.р.), 5. Экзамен.

Текущий контроль включает систематический блиц-опрос и проверку домашнего задания.

Промежуточный контроль проводится в виде отчета по пакетам заданий, предварительная проверка решений практикуется по файлам, отправленным по электронной почте.

Итоговый контроль проводится в виде устного экзамена с обязательным устным собеседованием.

Критерии выставления оценок:

«отлично» - владение теоретическим материалом, возможно, за исключением деталей справочного плана, и наличие навыков грамотного программирования;

«хорошо» - владение разделами «Задачи поиска», «Сортировка» и умение составлять простые программы (ориентировочно 10-20 строк);

«удовлетворительно» - знания по разделам «Алгоритмы и их сложность», «Инверсии», умение составлять элементарные программы и посещение занятий.

Пакет заданий для самостоятельной работы выдается по истечению месяца с начала семестра, определяются предельные сроки их выполнения и сдачи.

### 6.3. Примеры заданий для самостоятельного решения

**Упражнение 1.** Постройте машину Тьюринга, которая записывает входное двоичное слово в обратном порядке.

**Упражнение 2.** Постройте машину Тьюринга, которая удваивает исходное слово, т.е. из слова  $\alpha$  формирует слово  $\alpha * \alpha$ .

**Упражнение 3.** Постройте машину Тьюринга, которая вычисляет предикат « $\alpha$  - четное число».

**Упражнение 4.** Постройте машину Тьюринга, которая вычисляет для сложения  $n$  чисел.

**Упражнение 5.** Постройте машину Тьюринга для распознавания строки с одинаковым количеством 0 и 1.

**Упражнение 6.** Найдите перестановку для таблицы инверсии  $D = (3,4,4,5,1,0,2,1,0)$ .

**Упражнение 7.** Напишите программу, которая по заданной таблице инверсии восстанавливает перестановку.

**7. Фонд оценочных средств, для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.**

**7.1. Типовые контрольные задания**

**7.1.1. Темы рефератов:**

История развития понятия алгоритм;

Алан Тьюринги машина Тьюринга;  
Сложность алгоритма – что это такое;  
Флойд – великий математик;  
Сортировка и поиск;  
Задача математического программирования - задача оптимизации.

## 7.1.2. Примерные упражнения и задания для текущего контроля

### Вариант контрольной работы 1

Дан массив, содержащий  $n$  чисел.

1. вычислить среднее арифметическое отрицательных чисел, предшествующих максимальному по абсолютной величине элементу;
2. определить количество элементов массива, расположенных между максимальным и минимальным элементами;
3. отсортировать и вставить в него одно заданное число так, чтобы сохранилась упорядоченность;
4. Дано число, не меньше 1000. Определить:
  - а) является ли число палиндромом («перевертышем»);
  - б) количество трехзначных чисел, сумма цифр которого равна сумме цифр заданного числа;
5. Массив содержит все целые числа от 0 до  $n$ , кроме одного. Найдите недостающее число на основе метода грубой силы и метода декомпозиции. Оцените сложность каждого из предложенных алгоритмов.

### Вариант контрольной работы 2

1. Определите верхнюю и нижнюю асимптотические границы функции  $T(n) = 2T\left(\frac{n}{2}\right) + n^3$ . Считаем, что  $n \leq 2T(n)$  - константа. Сделайте оценку как можно более точной.
2. Покажите, что  $O(\lg n)$  является решением рекуррентного соотношения  $T(n/2) = T(n) + 1$ .
3. Вычислите  $2101 * 1130$  с применением метода декомпозиции.
4. Разработать алгоритм для вычисления значения полинома  $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$  с помощью метода грубой силы в заданной точке и определите его класс эффективности в наихудшем случае.
5. Разработайте такой алгоритм перестановки элементов в заданном массиве из  $n$  действительных чисел, чтобы все отрицательные предшествовали положительным.
6. Есть трехместная лодка, одно из мест забронировано человеком. Нужно переправить на другой берег козла, капусту, двух волков и собаку, причем собака в ссоре с волком, козел неравнодушен к капусте, а волк и собака не могут оставаться наедине с козлом.

### Вариант контрольной работы 3

1. Опишите стандартный алгоритм преобразования положительного числа в двоичное: а) словами; б) на псевдокоде.
2. Напишите алгоритм поиска минимальной разницы между двумя элементами массива.
3. Можно ли в одну прогулку обойти семь мостов Кенигсберга и вернуться в отправную точку?
  - а) сформулируйте эту задачу в терминах теории графов.
  - б) имеет ли задача решение? Если да, начертите схему обхода мостов. Если нет, то какое минимальное количество новых мостов нужно построить, чтобы задача имела решение?
4. Разработайте алгоритм решения следующей задачи. Предположим, на плоскости находится  $n$  точек, заданных их координатами  $(x,y)$ . Определите, лежат ли все точки на одной и той же окружности?

### 7.1.3. Примерные задания к промежуточному контролю (коллоквиуму)

#### Коллоквиум 1

1. Понятие алгоритма. Определения.
2. Основные требования к алгоритмам.
3. Машина Тьюринга.
4. Универсальная машина Тьюринга.
5. Тезис Тьюринга.
6. Граф машина.
7. Алгоритмическая модель языка GRAPH.

#### Коллоквиум 2

1. Модель данных.
2. Предметная область программирования.
3. Построение моделей алгоритмов в системе GRAPH.
4. Временная сложность алгоритма. Класс DTIME.
5. Пространственная сложность алгоритма. Класс DSPACE.
6. Классы сложности задач.
7. Полиномиальность и эффективность.
8. Алгоритмическая сводимость задач.
9. Представление абстрактных объектов (последовательностей).
10. Смежное представление последовательностей.
11. Связанное представление последовательностей.
12. Операции добавления и удаления элементов списка.
13. Характеристические векторы.
14. Использование списков для разработки алгоритма «Крестики-нолики».

#### Коллоквиум 3

1. Сортировка вставками.
2. Инверсии. Пример инверсии.

3. Сортировка всплытия Флойда.
4. Двоичные деревья на смежной памяти.
5. Процедура всплытия Флойда на бинарном дереве.
6. Задачи поиска Последовательный поиск.
7. Задачи поиска Логарифмический поиск.
8. Сортировка с вычисляемыми адресами.

#### **Коллоквиум 4**

1. Эффективность методов оптимизации.
2. Уни-modalные, непрерывные одномерные функции.
3. Оптимизации многоэкстремальных функций.
4. Сложность выпуклых экстремальных задач.
5. Графическое решение задачи линейного программирования
6. Симплекс метод решения задачи линейного программирования

#### **7.1.4. Экзаменационные вопросы**

1. Понятие алгоритма. Определения.
2. Основные требования к алгоритмам.
3. Машина Тьюринга.
4. Универсальная машина Тьюринга.
5. Тезис Тьюринга.
6. Граф машина.
7. Алгоритмическая модель языка GRAPH.
8. Модель данных.
9. Предметная область программирования.
10. Построение моделей алгоритмов в системе GRAPH.
11. Временная сложность алгоритма. Класс DTIME.
12. Пространственная сложность алгоритма. Класс DSPACE.
13. Классы сложности задач.
14. Полиномиальность и эффективность.
15. Алгоритмическая сводимость задач.
16. Представление абстрактных объектов (последовательностей).
17. Смежное представление последовательностей.
18. Связанное представление последовательностей.
19. Операции добавления и удаления элементов списка.
20. Характеристические векторы.
21. Использование списков для разработки алгоритма «Крестики-нолики».
22. Сортировка вставками.
23. Инверсии. Пример инверсии.
24. Сортировка всплытия Флойда.
25. Двоичные деревья на смежной памяти.
26. Процедура всплытия Флойда на бинарном дереве.
27. Задачи поиска Последовательный поиск.
28. Задачи поиска Логарифмический поиск.

29. Сортировка с вычисляемыми адресами.
30. Эффективность методов оптимизации.
31. Унимодальные, непрерывные одномерные функции.
32. Оптимизации многоэкстремальных функций.
33. Сложность выпуклых экстремальных задач.
34. Графическое решение задачи линейного программирования
35. Симплекс метод решения задачи линейного программирования

## **7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 30% и промежуточного контроля - 70%.

Текущий контроль по дисциплине включает:

- посещение занятий - 10 баллов,
- участие на практических занятиях - 10 баллов,
- выполнение домашних работ - 0 баллов.

Промежуточный контроль по дисциплине включает:

- коллоквиум - 40 баллов,
- письменная контрольная работа - 30 баллов.

## **8. Учебно-методическое обеспечение дисциплины**

а) основная литература:

1. Поляков В.И. Основы теории алгоритмов [Электронный ресурс] : учебное пособие по дисциплине «Математическая логика и теория алгоритмов» / В.И. Поляков, В.И. Скорубский. — Электрон.текстовые данные. — СПб.: Университет ИТМО, 2012. — 50 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/67504.html>
2. Крупский, Владимир Николаевич. Теория алгоритмов : [учеб. пособие для студентов вузов] / Крупский, Владимир Николаевич, В. Е. Плиско. - М. : Академия, 2009. - 205,[3] с. - (Университетский учебник. Серия "Прикладная математика и информатика"/ отв. ред.: Ю.И.Димитриенко). - Допущено НМС по математике МО РФ. - ISBN 978-5-7695-5293-9 : 434-61.  
Местонахождение: Научная библиотека ДГУ
3. Ездаков, Андрей Леонидович. Функциональное и логическое программирование : учеб. пособие / Ездаков, Андрей Леонидович. - М. : БИНОМ. Лаб. знаний, 2009. - 118,[1] с. - ISBN 978-5-94774-964-9 : 152-00.  
Местонахождение: Научная библиотека ДГУ
4. Соколов, Александр Павлович. Системы программирования: теория, методы, алгоритмы : [учеб. пособие по направлению 654600 "Информатика и вычисл. техника"] / Соколов, Александр Павлович. - М. : Финансы и статистика, 2004. - 319,[1] с. : ил. ; 21 см. - Библиогр.: с. 309-310.- Предм. указ.: с. 313-320. - Рекомендовано УМО. - ISBN 5-279-02770-7 : 101-97.

Местонахождение: Научная библиотека ДГУ

б) дополнительная литература:

1. Ключарев П.Г. Введение в теорию алгоритмов [Электронный ресурс] : учебное пособие / П.Г. Ключарев, Д.А. Жуков. — Электрон.текстовые данные. — М. : Московский государственный технический университет имени Н.Э. Баумана, 2012. — 39 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/31383.html>
2. Иванов, Борис Николаевич. Дискретная математика. Алгоритмы и программы : Учеб. пособие / Иванов, Борис Николаевич. - М. : Лаб. Баз. Знаний, 2001. - 288 с. - ISBN 5-93208-093-0 : 0-0.  
Местонахождение: Научная библиотека ДГУ
3. Бежанова, Майя Михайловна. Практическое программирование. Структуры данных и алгоритмы : Учеб. для вузов / Бежанова, Майя Михайловна ; Л.А.Москвина, И.В.Поттосин. - М. : Логос, 2001. - 223 с. - ISBN 5-94010-124-0 : 0-0.  
Местонахождение: Научная библиотека ДГУ
4. Гашков, Сергей Борисович. Арифметика. Алгоритмы. Сложность вычислений : учеб. пособие / Гашков, Сергей Борисович, Чубариков, Владимир Николаевич ; под ред. В.А.Садовниченко. - 2-е изд., перераб. - М. : Высш. шк., 2000. - 320 с. - Рекомендовано МО. - ISBN 5-06-003613-8 : 29-90.  
Местонахождение: Научная библиотека ДГУ

## **9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», использование которых целесообразно для освоения дисциплины**

<http://www.elib.dgu.ru/>  
<http://www.iprbookshop.ru/>  
<http://intuit.ru/>

## **10. Методические указания по освоению дисциплины**

Для понимания материала и качественного его усвоения рекомендуется следующая последовательность действий:

1. После прослушивания лекции и окончания учебных занятий, при подготовке к занятиям следующего дня, нужно сначала просмотреть и обдумать текст предыдущей лекции – 10-15 минут.
2. В течение недели выбрать время – 1 час для работы с литературой по программированию и анализу алгоритмов.
3. При подготовке к лабораторным занятиям, необходимо сначала прочитать основные понятия, изучить алгоритмы по теме домашнего задания. При написании программы нужно сначала понять, что требуется, какой теоретический материал нужно использовать, наметить план решения задачи. Алгоритм решения задачи – это не программа ее решения, а способ дать человеку (а не машине) представление о структуре алгоритма, о смысле его



шагов и их логической взаимосвязи. Поэтому шаги алгоритма должны описываться в терминах тех объектов и отношений между ними, о которых идет речь в условии задачи (это, конечно, не исключает использования математической и другой условной символики). Структура алгоритма станет более ясной, если ее описывать в наглядной и достаточно формализованной (напоминающей конструкции языка программирования) форме. Поэтому требуемой формой описания алгоритма в данном лабораторном практикуме является либо графическое представление алгоритма на языке блок-схем, либо на специальном языке описания алгоритмов, например, в школьном алгоритмическом языке.

4. Основная часть теоретического материала курса дается в ходе лекционных занятий, хотя часть материала может изучаться на лабораторных занятиях, либо самостоятельно по учебной литературе.

## **11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем**

При осуществлении образовательного процесса по дисциплине «Алгоритмы и анализ сложности» рекомендуется использовать следующие информационные технологии.

Во-первых, должны проводиться занятия с компьютерным тестированием, что приучит студентов хорошо ориентироваться с работой на компьютере для выполнения заданий.

Во-вторых, демонстрационный материал также будет показан с помощью мультимедийных устройств и интерактивной доски.

В-третьих, компьютерные классы с набором лицензионного базового программного обеспечения для проведения занятий.

## **12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

При освоении дисциплины для выполнения лабораторных работ необходимы классы персональных компьютеров с приложениями программирования на языках С/С++, а также учебные аудитории для проведения лекционных занятий.

В библиотеке ДГУ имеется необходимая литература, созданы и размещены на сайте кафедры учебно-методические пособия, на каждой лекции используется стационарное мультимедийное презентационное оборудование (ауд. 3-72).

Практические занятия проводятся в компьютерных классах с современным аппаратным и программным обеспечением.

Часть лекций предоставляется студенту в электронном формате. Студент имеет также возможность копировать литературу с сайта кафедры.