

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ 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: **базовая**

Махачкала - 2017

Рабочая программа дисциплины «Алгоритмы и анализ сложности» составлена в 2017 году в соответствии с требованиями ФГОС ВО по направлению подготовки

02.03.02 – Фундаментальная информатика и информационные технологии (уровень бакалавриат)

Приказ Минобрнауки России от 12.03.2015 №224

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

Ибрагимов

Рабочая программа дисциплины «Алгоритмы и анализ сложности» одобрена: на заседании кафедры дискретной математики от «13» *сентября* 2017 г., протокол № 5.

Зав. кафедрой *А.М.* Магомедов А.М.

на заседании Методической комиссии факультета математики и компьютерных наук от «__» _____ 20__ г., протокол №__.

Председатель *З.Г.* Меджидов З.Г.

Рабочая программа дисциплины согласована с учебно-методическим управлением «__» _____ 20__ г. *А.М.*

Содержание

Аннотация рабочей программы дисциплины

1. Цели освоения дисциплины
2. Место дисциплины в структуре ООП бакалавриата
3. Компетенции обучающегося, формируемые в результате освоения дисциплины (перечень планируемых результатов обучения)
4. Объем, структура и содержание дисциплины
5. Образовательные технологии
6. Учебно-методическое обеспечение самостоятельной работы студентов
7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины
10. Методические указания для обучающихся по освоению дисциплины
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Аннотация рабочей программы дисциплины

Дисциплина «Алгоритмы и анализ сложности» входит в базовую часть образовательной программы бакалавриата по направлению (специальности) 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 профессиональных и специальных 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-9.

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 тестирования.

Промежуточный контроль в форме экзамена.

Объем дисциплины **5** зачетных единиц, в том числе в академических часах по видам учебных занятий

Семестр	Учебные занятия						СРС, в том числе экзамен	Форма промежуточной аттестации (зачет, дифференцированный зачет, экзамен)
	в том числе							
	Контактная работа обучающихся с преподавателем							
	Все	из них						
го	Лекции	Лабораторные занятия	Практические занятия	КСР	консультации			
3	180	36	-	36	-	-	108	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 ООП бакалавриата

Дисциплина «Алгоритмы и анализ сложности» входит в базовую часть образовательной программы бакалавриата, по направлению (специальности)

02.03.02-Фундаментальная информатика и информационные технологии.

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

- Теория функций. Знакомство с понятиями предела функции, непрерывной, разрывной функции. Навыки в отыскании пределов функций;
- Теория рядов. Знакомство с понятиями числового, функционального рядов, степенного ряда, ряда Фурье. Практические навыки в отыскании пределов последовательностей и рядов;
- 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)

Компетенции	Формулировка компетенции из ФГОС ВО	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенций)
ОПК-3	Способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	<p>Знать: детально методы и базовые алгоритмы обработки информационных структур, методов анализа сложности алгоритмов;</p> <p>Уметь: использовать свои знания на практике;</p> <p>Владеть: основными методами, способами и средствами получения, хранения, переработки информации, иметь навыки работы с компьютером как средством управления информацией;</p> <p>пониманием концепций, синтаксической и семантической организации, методов использования современных языков программирования;</p> <p>пониманием концепций, базовых алгоритмов, принципов разработки и функционирования современных операционных систем.</p>
ПК-9	Способностью разрабатывать, оценивать и реализовывать процессы жизненного цикла информационных систем, программного обеспечения, сервисов информационных технологий, а также реализовывать методы и механизмы оценки и анализа функционирования средств и информационных технологий, разрабатывать проектную и программную документацию, удовлетворяющую нормативным требованиям	<p>Знать: основные определения курса; основные методы анализа сложности алгоритмов;</p> <p>Уметь: анализировать нерекурсивные алгоритмы, проводить анализ рекурсивных алгоритмов; применять алгоритмы разбиения, уменьшения на единицу и т.д. для решения прикладных задач;</p> <p>Владеть: навыками построения алгоритмов разбиения, алгоритмов динамического программирования, жадных алгоритмов; техникой анализа сложности алгоритмов.</p>

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)				Самостоятельная работа	Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лаб. занят.	Контроль самост. раб.		
1	Модуль 1. Введение в теорию алгоритмов.								
2	Тема 1. Введение в теорию алгоритмов. Понятие алгоритма. Основные требования к алгоритмам. Машина Тьюринга. Универсальная машина Тьюринга. Тезис Тьюринга.	3	1-3	6	6			6	Тестирование, письменная контрольная работа
3	Тема 2. Граф машина. Алгоритмическая модель языка GRAPH. Модель данных. Предметная область программирования. Построение моделей алгоритмов в системе GRAPH.	3	4-6	6	6			6	
4	Итого по модулю 1:	3	1-6	12	12			12	Коллоквиум
5	Модуль 2. Сложность алгоритмов.								
6	Тема 3. Временная и пространственная сложность алгоритма. Класс DTIME и DSPACE. Классы сложности задач. Полиномиальность и эффективность. Алгоритмическая сводимость задач.	3	7-9	6	6			6	Тестирование, письменная контрольная работа
7	Тема 4. Представление абстрактных объектов (последовательностей).	3	10-12	6	6			6	

	Смежное представление последовательностей. Связанное представление последовательностей. Операции добавления и удаления элементов списка. Характеристические векторы. Списки. Деревья. Использование списков для разработки алгоритма «Крестики-нолики».								
8	Итого по модулю 2:	3	7-12	6	6			24	Коллоквиум
9	Модуль 3. Сортировка и поиск.								
10	Тема 5. Сортировка вставками. Инверсии. Сортировка всплытия Флойда. Двоичные деревья на смежной памяти. Процедура всплытия Флойда на бинарном дереве.	3	13-14	4	4			10	Тестирование, письменная контрольная работа
11	Тема 6. Задачи поиска. Последовательный поиск. Логарифмический поиск. Сортировка с вычисляемыми адресами.	3	15-16	4	4			10	
12	Итого по модулю 3:	3	13-16	8	8			20	Коллоквиум
13	Модуль 4. Эффективность методов оптимизации.								
14	Тема 7. Унимодальные, непрерывные одномерные функции. Оптимизация многоэкстремальных функций. Сложность выпуклых экстремальных задач.	3	17-18	4	4			28	Тестирование, письменная контрольная работа
15	Итого по модулю 4:	3	17-18	4	4			28	Коллоквиум
16	Модуль 5. Подготовка к экзамену								
17	Подготовка к экзамену	3	19					36	Экзамен
18	Итого по модулю 5:	3	19					36	Экзамен
19	Итого за 3 семестр:	3	1-19	36	36			108	Экзамен
20	Итого:	3	1-19	36	36			108	Экзамен

4.3. Содержание дисциплины, структурированное по темам (разделам)

Лекции

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.

Лекция 3. Тезис Тьюринга.

Лекция 4. Граф машина. Алгоритмическая модель языка GRAPH.

Лекция 5. Модель данных. Предметная область программирования.

Лекция 6. Построение моделей алгоритмов в системе GRAPH.

Модуль 2. Сложность алгоритмов

Лекция 7. Временная и пространственная сложность алгоритма. Класс DTIME и DSPACE.

Лекция 8. Классы сложности задач.

Лекция 9. Полиномиальность и эффективность. Алгоритмическая сводимость задач.

Лекция 10. Представление абстрактных объектов (последовательностей). Смежное представление последовательностей. Связанное представление последовательностей.

Лекция 11. Операции добавления и удаления элементов списка. Характеристические векторы.

Лекция 12. Списки. Деревья. Использование списков для разработки алгоритма «Крестики-нолики».

Модуль 3. Сортировка и поиск

Лекция 13. Сортировка вставками. Инверсии. Сортировка всплытия Флойда. Двоичные деревья на смежной памяти.

Лекция 14. Процедура всплытия Флойда на бинарном дереве.

Лекция 15. Задачи поиска. Последовательный поиск. Логарифмический поиск.

Лекция 16. Сортировка с вычисляемыми адресами.

Модуль 4. Эффективность методов оптимизации

Лекция 17. Униmodalные, непрерывные одномерные функции.

Лекция 18. Оптимизации многоэкстремальных функций. Сложность выпуклых экстремальных задач.

Модуль 5. Подготовка к экзамену

Практические занятия

3 семестр

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

Занятие 2. Машина Тьюринга. Универсальная машина Тьюринга.

Занятие 3. Тезис Тьюринга.

Занятие 4. Граф машина. Алгоритмическая модель языка GRAPH.

Занятие 5. Модель данных. Предметная область программирования.

Занятие 6. Построение моделей алгоритмов в системе GRAPH.

Занятие 7. Временная и пространственная сложность алгоритма. Класс DTIME и DSPACE.

Занятие 8. Классы сложности задач.

Занятие 9. Полиномиальность и эффективность. Алгоритмическая сводимость задач.

Занятие 10. Представление абстрактных объектов (последовательностей). Смежное представление последовательностей. Связанное представление последовательностей.

Занятие 11. Операции добавления и удаления элементов списка. Характеристические векторы.

Занятие 12. Списки. Деревья. Использование списков для разработки алгоритма «Крестики-нолики».

Занятие 13. Сортировка вставками. Инверсии. Сортировка всплытия Флойда. Двоичные деревья на смежной памяти.

Занятие 14. Процедура всплытия Флойда на бинарном дереве.

Занятие 15. Задачи поиска. Последовательный поиск. Логарифмический поиск.

Занятие 16. Сортировка с вычисляемыми адресами.

Занятие 17. Унимодальные, непрерывные одномерные функции.

Занятие 18. Оптимизации многоэкстремальных функций. Сложность выпуклых экстремальных задач.

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.
5. Круглые столы.

6. Учебно-методическое обеспечение самостоятельной работы студентов

Учебно-методические пособия для самостоятельной работы

1. Практикум по методам параллельных вычислений: учебник для вузов / А. В. Старченко и др.; под ред. А. В. Старченко; Томский гос. ун-т. - М.: Изд-во Московского университета, 2010. - 199 с.
2. Шандаков Ю.Д. Программирование в среде Visual Basic. - Кемерово: Сибирская издательская группа, 2009. - 74 с.

Задания для самостоятельной работы

Разделы (модули) и темы для самостоятельного изучения	Виды и содержание самостоятельной работы
Первый семестр	
Модуль 1. Введение в теорию алгоритмов.	
Тема 1. Введение в теорию алгоритмов. Понятие алгоритма. Основные требования к алгоритмам. Машина Тьюринга. Универсальная машина Тьюринга. Тезис Тьюринга.	Доклад на тему: «История развития понятия алгоритм». Решение задач и упражнений.
Тема 2. Граф машина. Алгоритмическая модель языка GRAPH. Модель данных. Предметная область программирования. Построение моделей алгоритмов в системе GRAPH.	Доклад на тему: «Тьюринг-великий математик». Решение задач и упражнений.
Модуль 2. Сложность алгоритмов.	
Тема 3. Временная и пространственная сложность алгоритма. Класс DTIME и DSPACE. Классы сложности задач. Полиномиальность и эффективность. Алгоритмическая сводимость задач.	Доклад на тему: «Сложность алгоритма – что это такое». Решение задач и упражнений.
Тема 4. 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».	Решение задач и упражнений.
Модуль 3. Сортировка и поиск.	
Тема 5. Сортировка вставками. Инверсии. Сортировка всплытия Флойда. Двоичные деревья на смежной памяти. Процедура всплытия Флойда на бинарном дереве.	Доклад на тему: «Флойд – великий математик». Решение задач и упражнений.
Тема 6. Задачи поиска. Последовательный поиск. Логарифмический поиск. Сортировка с вычисляемыми адресами.	Доклад на тему: «Сортировка и поиск». Решение задач и упражнений.
Модуль 4. Эффективность методов оптимизации.	
Тема 7. Унимодалльные, непрерывные одномерные функции. Оптимизация многоэкстремальных функций. Сложность выпуклых экстремальных задач.	Доклад на тему: «Задача математического программирования - задача оптимизации». Решение задач и упражнений.

Упражнения

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 α формирует слово $\alpha * \alpha$.

Упражнение 3. Постройте машину Тьюринга, которая вычисляет предикат « α - 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. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

Перечень компетенций с указанием этапов их формирования приведен в описании образовательной программы

Компетенция	Знания, умения, навыки	Процедура освоения
ОПК-3	Знать: детально методы и базовые алгоритмы обработки информационных структур, методов анализа сложности алгоритмов.	Устный опрос, письменный опрос, тестирование.
	Уметь: использовать свои знания на практике.	Письменный опрос, коллоквиум.
	Владеть: основными методами, способами и средствами получения, хранения, переработки информации, иметь навыки	Круглый стол.

	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.	
ПК-9	Знать: основные определения курса; основные методы анализа сложности алгоритмов.	Устный опрос, письменный опрос, тестирование.
	Уметь: анализировать нерекурсивные алгоритмы, проводить анализ рекурсивных алгоритмов; применять алгоритмы разбиения, уменьшения на единицу и т.д. для решения прикладных задач.	Письменный опрос, коллоквиум.
	Владеть: навыками построения алгоритмов разбиения, алгоритмов динамического программирования, жадных алгоритмов; техникой анализа сложности алгоритмов.	Круглый стол

7.2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

ОПК-3 - Способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Пороговый	Знать: детально методы и базовые алгоритмы обработки информационных структур, методов анализа сложности алгоритмов. Уметь: использовать свои знания на практике. Владеть: основными методами, способами и	Демонстрация частичных знаний без грубых математических ошибок	Умение анализировать алгоритм решения заданий и объяснять его коллективу	Умение обоснованно анализировать ответ, приводя собственные примеры

	<p>средствами получения, хранения, переработки информации, иметь навыки работы с компьютером как средством управления информацией;</p> <p>пониманием концепций, синтаксической и семантической организации, методов использования современных языков программирования;</p> <p>пониманием концепций, базовых алгоритмов, принципов разработки и функционирования современных операционных систем.</p>			
--	--	--	--	--

ПК-9 - Способностью разрабатывать, оценивать и реализовывать процессы жизненного цикла информационных систем, программного обеспечения, сервисов информационных технологий, а также реализовывать методы и механизмы оценки и анализа функционирования средств и информационных технологий, разрабатывать проектную и программную документацию, удовлетворяющую нормативным требованиям

Уровень	Показатели (что обучающийся должен продемонстрировать)	Оценочная шкала		
		Удовлетворительно	Хорошо	Отлично
Пороговый	<p>Знать: основные определения курса; основные методы анализа сложности алгоритмов.</p> <p>Уметь: анализировать нерекурсивные алгоритмы, проводить анализ рекурсивных алгоритмов; применять алгоритмы разбиения, уменьшения на единицу и т.д. для решения прикладных задач.</p> <p>Владеть: навыками построения алгоритмов</p>	<p>Демонстрирует частичное знание содержания процессов самоорганизации и самообразования, их особенностей и технологий реализации, исходя из целей совершенствования профессиональной деятельности.</p> <p>При планировании и установлении</p>	<p>Демонстрирует знание содержание процессов самоорганизации и самообразования, их особенностей и технологий реализации, исходя из целей совершенствования профессиональ</p>	<p>Владеет полной системой знаний о содержании, особенностях процессов самоорганизации и самообразования, аргументированно обосновывает принятые решения при выборе технологий их реализации с учетом целей</p>

	<p>разбиения, алгоритмов динамического программирования, жадных алгоритмов; техникой анализа сложности алгоритмов.</p>	<p>приоритетов целей профессиональной деятельности не полностью учитывает внешние и внутренние условия их достижения.</p> <p>Владеет отдельными методами и приемами отбора необходимой для усвоения информации, давая не полностью аргументированное обоснование ее соответствия целям самообразования.</p> <p>Владеет отдельными приемами саморегуляции, но допускает существенные ошибки при их реализации, не учитывая конкретные условия и свои возможности при принятии решений.</p> <p>Владеет отдельными приемами организации собственной познавательной деятельности, осознавая перспективы профессионального развития, но не давая аргументированное обоснование адекватности отобранной для усвоения</p>	<p>ной деятельности.</p> <p>Планируя цели деятельности с учетом условий их достижения, дает не полностью аргументированное обоснование соответствия выбранных способов выполнения деятельности намеченным целям.</p> <p>Владеет системой отбора содержания обучения в соответствии с намеченными целями самообразования, но при выборе методов и приемов не полностью учитывает условия и личные возможности овладения этим содержанием.</p> <p>Демонстрирует возможность и обоснованность реализации приемов саморегуляции при выполнении деятельности в конкретных заданных</p>	<p>профессионального и личностного развития.</p> <p>Готов и умеет формировать приоритетные цели деятельности, давая полную аргументацию принимаемым решениям при выборе способов выполнения деятельности.</p> <p>Умеет строить процесс самообразования с учетом внешних и внутренних условий реализации.</p> <p>Демонстрирует обоснованный выбор приемов саморегуляции при выполнении деятельности в условиях неопределенности .</p> <p>Демонстрирует возможность переноса технологии организации процесса самообразования, сформированной в одной сфере деятельности, на другие сферы, полностью обосновывая выбор используемых методов и приемов.</p>
--	--	--	---	---

		информации целям самообразования.	условиях. Владеет системой приемов организации процесса самообразования только в определенной сфере деятельности.	
--	--	-----------------------------------	--	--

Если хотя бы одна из компетенций не сформирована, то положительная оценки по дисциплине быть не может.

7.3. Типовые контрольные задания

Примерные задания для текущего контроля знаний

Вариант контрольной работы

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 меньшее 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

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 $p(x) = a_n x^n + a_1 x^{n-1} + \dots + a_1 x + a_0$ 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 равнодушен к капусте, а волк и собака не могут оставаться наедине с козлом.

Вариант контрольной работы

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?

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.

7.4. 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 - 30 баллов,
- участие на практических занятиях - 40 баллов,
- выполнение домашних работ - 30 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 40 баллов,
- письменная контрольная работа - 30 баллов.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная литература:

1. Коварцев А.Н. Автоматизация разработки и тестирования программных средств. – Самара, СГАУ. 1999.
2. Кузнецов О.П., Адельсон-Вельский Г.М. Дискретная математика для инженера. – М.: Энергоатомиздат, 1988.
3. Кузюрин Н.Н., Фомин С.А. Эффективные алгоритмы и сложность вычислений. 2008.
4. Немировский А.С., Юдин Д.Б. Сложность и эффективность методов оптимизации. М.: Наука, 1979.
5. Практикум по методам параллельных вычислений: учебник для вузов / А. В. Старченко и др.; под ред. А. В. Старченко; Томский гос. ун-т. - М.: Изд-во Московского университета, 2010. - 199 с.
6. Шандаков Ю.Д. Программирование в среде Visual Basic. - Кемерово: Сибирская издательская группа, 2009. - 74 с.

б) дополнительная литература:

1. Абрамов С.А. Вычислительная сложности алгоритмов. М.: Изд-во МГУ им. М.В. Ломоносова, 2005.
2. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Наука, 1979.
3. Кнут В. Искусство программирования. Основные алгоритмы. М.: Наука, 1984.
4. Разборов А.А. О сложности вычислений. М.: Просвещение, 2003.

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

<http://eqworld.ipmnet.ru/ru/solutions/ode.htm> Мир математических уравнений библиотека;

<http://www.poiskknig.ru/> поиск электронных книг;

<http://discopal.ispras.ru/ru.book-advanced-algorithms.htm>;

www.mathnet.ru - общероссийский математический портал;

www.library.kemsu.ru - электронный каталог НБ КемГУ;

www.elibrary.ru - научная электронная библиотека;

www.lib.mexmat.ru - электронная библиотека механико-математического факультета МГУ;

www.newlibrary.ru - новая электронная библиотека;

www.edu.ru - федеральный портал российского образования.

10. Методические указания для обучающихся 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 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 C/C++, а также учебные аудитории для проведения лекционных занятий.