

3к 2011
2.3к 3/6

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Проректор по учебной работе


_____ **А.Л. Толстик** _____

_____ 31.03.2015 _____

(дата утверждения)

Регистрационный № УД-_____ 6 /уч.



ПОСТРОЕНИЕ И АНАЛИЗ АЛГОРИТМОВ

Учебная программа учреждения высшего образования
по учебной дисциплине для специальности:

**1-31 03 08 Математика и информационные технологии
(по направлениям)**

Направление специальности

1-31 03 08-01 Веб-программирование и интернет-технологии

2015 г.

Учебная программа составлена на основе образовательных стандартов высшего образования ОСВО 1-31 03 08-2013 и учебных планов G31-195/уч., G31з-198/уч., G31з-200/уч. от 30.05.2013.

СОСТАВИТЕЛЬ:

К.Г. Кузьмин, доцент кафедры веб-технологий и компьютерного моделирования Белорусского государственного университета, кандидат физико-математических наук, доцент.

РЕЦЕНЗЕНТЫ:

Н.А. Волорова, заведующая кафедрой информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент;

Ю.Н. Сотсков, главный научный сотрудник государственного научного учреждения «Объединенный институт проблем информатики Национальной академии наук Беларуси», доктор физико-математических наук, профессор.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой веб-технологий и компьютерного моделирования Белорусского государственного университета
(протокол № 10 от 14.05.2015);

Научно-методическим советом Белорусского государственного университета
(протокол № 6 от 29.06.2015).



К. Г. Кузьмин



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Развитие современного производства, его сложная разветвленная структура приводит к все более возрастающей потребности современного общества в новейших информационных технологиях. Это вызывает к жизни множество новых задач и проблем, требующих адекватного аппарата для своего решения. Общеизвестно, что комбинаторные методы и алгоритмы являются наиболее удобным и эффективным средством решения этих задач.

Современное программирование невозможно представить себе без эффективных алгоритмов. Это связано с тем, что очень многие возникающие в области информационных технологий задачи характеризуются большим числом составляющих элементов и огромным количеством связей между ними. Такие задачи весьма сложны для решения даже с использованием новейшей и самой совершенной на данный момент вычислительной техники. Поэтому создание сложного программного продукта в настоящее время возможно лишь с использованием мощной алгоритмической базы, за которой стоит соответствующий математический аппарат – аппарат теории комбинаторных алгоритмов и дискретной оптимизации.

Основной целью учебной дисциплины «Построение и анализ алгоритмов» является ознакомление студентов с наиболее часто используемыми комбинаторными алгоритмами, с основными идеями, методами и алгоритмическими стратегиями, что позволит подготовить их к решению реальных задач, возникающих на практике.

Развивающей целью учебной дисциплины является дальнейшее формирование у студентов навыков алгоритмического мышления и умения применять его в конкретных задачах.

Основными задачами, решаемыми в рамках изучения дисциплины «Построение и анализ алгоритмов», являются изучение терминологии, основных утверждений и методов их доказательства, освоение приемов решения типовых задач, а также ознакомление со способами моделирования практических задач в терминах задач из рассматриваемых разделов дискретной математики.

В результате изучения дисциплины студент должен:

знать: основные структуры данных; алгоритмы решения основных задач комбинаторной оптимизации, основные алгоритмические стратегии; методы оценки трудоемкости алгоритмов и методы анализа сложности задач;

уметь: разрабатывать программные реализации основных алгоритмов и структур данных; применять основные алгоритмы и структуры данных для практических задач, возникающих в программировании, экономике, компьютерной графике, защите информации; разрабатывать алгоритмы решения задач на основе известных алгоритмических стратегий;

владеть: методами для создания и реализации структур данных; методами оценки трудоемкости алгоритмов; подходами к решению алгоритмических задач на основе известных алгоритмических стратегий.

Для понимания дисциплины студенту требуется минимум предварительных математических знаний и навыков. В частности, нужно иметь представление о булевых функциях, начальных сведениях из математического анализа, линейной алгебры и теории графов. Учебная программа предназначена для студентов очной и заочной формы получения образования.

В соответствии с образовательным стандартом специальности 1-31 03 08 Математика и информационные технологии (по направлениям), направление 1-31 03 08-01 Веб-программирование и интернет-технологии, учебная программа предусматривает для изучения дисциплины:

– для студентов очной формы обучения – 244 учебных часа, из которых 122 аудиторных, в том числе:

в 5-м семестре 122 учебных часа, из которых 72 аудиторных часа, в том числе лекций – 36 часов, практических занятий – 30 часов, УСР – 6 часов;
в 6-м семестре 122 учебных часа, из которых 50 аудиторных часов, в том числе лекций – 16 часов, практических занятий – 30 часов, УСР – 4 часа.

Текущая аттестация по учебной дисциплине проводится в 5-м семестре в форме зачета и в 6-м семестре в форме экзамена.

– для студентов полной заочной формы обучения – 244 учебных часа, из которых 32 аудиторных, в том числе:

в 3-м семестре 102 учебных часа, из которых 16 аудиторных часов, в том числе лекций – 10 часов, практических занятий – 6 часов;
в 4-м семестре 102 учебных часа, из которых 16 аудиторных часов, в том числе лекций – 8 часов, практических занятий – 8 часов;
в 5-м семестре 40 учебных часов, среди которых нет аудиторных часов;

Текущая аттестация по учебной дисциплине проводится в 4-м в форме зачета, в 5-м семестре – в форме экзамена, а также зачетов по контрольной работе в 4-м и 5-м семестрах.

– для студентов заочной формы обучения на основе среднего специального образования – 244 учебных часа, в том числе 32 аудиторных, из них

в 1-м семестре 102 учебных часа, из которых 16 аудиторных часов, в том числе лекций – 10 часов, практических занятий – 6 часов;
в 2-м семестре 102 учебных часа, из которых 16 аудиторных часов, в том числе лекций – 8 часов, практических занятий – 8 часов;
в 3-м семестре 40 учебных часов, среди которых нет аудиторных часов;

Текущая аттестация по учебной дисциплине проводится в 2-м в форме зачета, в 3-м семестре – в форме экзамена, а также зачетов по контрольной работе в 2-м и 3-м семестрах.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

1. Основные структуры данных и сложность алгоритмов. Введение. Предмет теории алгоритмов. Прикладное значение эффективности алгоритмов. Связь с дискретной математикой, математической кибернетикой, программированием. Стеки, очереди, связанные списки, бинарные деревья. Трудоемкость алгоритмов. Необходимость оценки алгоритмов. Принципы оценки трудоемкости комбинаторных алгоритмов. Асимптотики трудоемкости алгоритмов O , Ω , Θ . Трудоемкость «в худшем», трудоемкость «в среднем», трудоемкость «почти всегда». Примеры анализа алгоритмов. Основы теории сложности вычислений. Детерминированные и недетерминированные машины Тьюринга. Понятие о классах P и NP . NP -полные задачи. Теорема Кука–Карпа–Левина. NP -полнота задач «3-выполнимость», «Клика», «Гамильтонов цикл». Списки наиболее известных NP -полных задач. Алгоритмы сортировки. Сортировка выбором, вставками, пузырьковая, Шелла. Быстрая сортировка. Сортировка слиянием. Внешняя сортировка. Линейные сортировки (подсчетом и вычерпыванием). Теорема о невозможности существования алгоритма сортировки в «худшем» и «в среднем» с трудоемкостью лучшей, чем $O(n \log n)$. Алгоритмы поиска и выборки. Последовательный поиск. Бинарный поиск. Деревья бинарного поиска. Сбалансированные деревья и реализуемые с их помощью структуры. Операции над деревьями. Хеширование. Открытая и закрытая адресация. Первичные и вторичные хеш-функции.

2. Алгоритмы на графах. Структуры данных для представления графов: матрицы смежности, матрицы инцидентности, списки смежности, списки ребер. Алгоритмы поиска в ширину и глубину, их реализация. Поиск минимального остовного дерева и кратчайшего пути в графе. Алгоритмы Прима, Краскала. Поиск кратчайших расстояний в графе. Алгоритмы Дейкстры, Флойда, Беллмана–Форда. Алгоритм нахождения эйлерова цикла. Паросочетания в двудольных графах, метод увеличивающей цепи. Потoki в сетях, алгоритм Форда–Фалкерсона.

3. Алгоритмы на строках. Основные определения. Алгоритмы поиска подстроки в строке. Алгоритм Бойера–Мура. Алгоритм Кнута–Морриса–Пратта.

4. Основные алгоритмические стратегии. Эвристики и метаэвристики. Алгоритмы локального поиска, поиска с запретами. Генетические алгоритмы, алгоритмы имитации отжига. Алгоритмы полного перебора, метод ветвей и границ. Динамическое программирование и метод «разделяй и властвуй». Понятие о методах динамического программирования и «разделяй и властвуй». Алгоритм Штрассена умножения матриц. Алгоритмы с гарантированной оценкой точности. Понятие о гарантированной оценке точности алгоритма. Задача упаковки. Задача о максимальном разрезе. Задача о вершинном покрытии. Задача о коммивояжере с неравенством треугольника. Задачи линейного и

целочисленного линейного программирования. Понятие о задачах линейного и целочисленного линейного программирования. Формулировки основных задач дискретной оптимизации как задач целочисленного линейного программирования. Алгоритмы для работы с матрицами. Решение систем линейных уравнений. LU -разложение. Обращение матриц. Нейронные сети. Понятие о нейронной сети. Типы нейронных сетей. Обучение нейронных сетей.

5. Элементы криптографии. Криптосистемы с открытым ключом, цифровая подпись. Криптосистема RSA . Алгоритмы проверки чисел на простоту. Алгоритмы сжатия информации. Коды Хаффмана, алгоритм Хаффмана.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ ДЛЯ СТУДЕНТОВ ОЧНОЙ ФОРМЫ ОБУЧЕНИЯ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	Основные структуры данных и сложность алгоритмов	22	16				2	
1.1	Введение. Предмет теории алгоритмов. Прикладное значение эффективности алгоритмов. Связь с дискретной математикой, математической кибернетикой, программированием. Стеки, очереди, связанные списки, бинарные деревья.	2						
1.2	Трудоёмкость алгоритмов. Необходимость оценки алгоритмов. Принципы оценки трудоёмкости комбинаторных алгоритмов. Асимптотики трудоёмкости алгоритмов O , Ω , Θ . Трудоёмкость «в худшем», трудоёмкость «в среднем», трудоёмкость «почти всегда». Примеры анализа алгоритмов.	2	2					
1.3	Основы теории сложности вычислений. Детерминированные и недетерминированные машины Тьюринга.	2	2					
1.4	Понятие о классах P и NP . NP -полные задачи. Теорема Кука–Карпа–Левина. NP -полнота задач «3-выполнимость», «Клика», «Гамильтонов цикл». Списки наиболее известных NP -полных задач.	4	2				1	Самостоят. работа
1.5	Алгоритмы сортировки.	2	2					

	Сортировка выбором, вставками, пузырьковая, Шелла. Быстрая сортировка.							
1.6	Сортировка слиянием. Внешняя сортировка. Линейные сортировки (подсчетом и вычерпыванием). Теорема о невозможности существования алгоритма сортировки в «худшем» и «в среднем» с трудоемкостью лучшей, чем $O(n \log n)$.	2	2					
1.7	Алгоритмы поиска и выборки. Последовательный поиск. Бинарный поиск.	2	1					
1.8	Деревья бинарного поиска. Сбалансированные деревья и реализуемые с их помощью структуры. Операции над деревьями.	4	2				1	Устный опрос
1.9	Хеширование. Открытая и закрытая адресация. Первичные и вторичные хеш-функции.	2	1					
2	Алгоритмы на графах	10	10				3	
2.1	Структуры данных для представления графов: матрицы смежности, матрицы инцидентности, списки смежности, списки ребер. Алгоритмы поиска в ширину и глубину, их реализация	2	2					
2.2	Поиск минимального остовного дерева и кратчайшего пути в графе. Алгоритмы Прима, Краскала.	2	2				1	Устный опрос
2.3	Поиск кратчайших расстояний в графе. Алгоритмы Дейкстры, Флойда, Беллмана–Форда.	2	2					
2.4	Алгоритм нахождения эйлера цикла.	2	2					
2.5	Паросочетания в двудольных графах, метод увеличивающей цепи. Потoki в сетях, алгоритм Форда–Фалкерсона.	2	2				2	Контрольная работа
3	Алгоритмы на строках	4	4				1	
3.1	Основные определения. Алгоритмы поиска подстроки в строке. Алгоритм Бойера–Мура.	2	2					
3.2	Алгоритм Кнута–Морриса–Пратта.	2	2				1	Устный опрос

4	Основные алгоритмические стратегии	12	24				3	
4.1	Эвристики и метаэвристики. Алгоритмы локального поиска, поиска с запретами. Генетические алгоритмы, алгоритмы имитации отжига. Алгоритмы полного перебора, метод ветвей и границ.	2	4					
4.2	Динамическое программирование и метод «разделяй и властвуй». Понятие о методах динамического программирования и «разделяй и властвуй». Алгоритм Штрассена умножения матриц.	2	4				1	Устный опрос
4.3	Алгоритмы с гарантированной оценкой точности. Понятие о гарантированной оценке точности алгоритма. Задача упаковки. Задача о максимальном разрезе. Задача о вершинном покрытии.	2	6					
4.4	Задачи линейного и целочисленного линейного программирования. Понятие о задачах линейного и целочисленного линейного программирования. Формулировки основных задач дискретной оптимизации как задач целочисленного линейного программирования.	2	4				1	Самостоят. работа
4.5	Алгоритмы для работы с матрицами. Решение систем линейных уравнений. LU - разложение. Обращение матриц.	2	2					
4.6	Нейронные сети. Понятие о нейронной сети. Типы нейронных сетей. Обучение нейронных сетей.	2	4				1	Контрольная работа
5	Элементы криптографии	4	6				1	
5.1	Криптосистемы с открытым ключом, цифровая подпись. Криптосистема RSA . Алгоритмы проверки чисел на простоту.	2	4					
5.2	Алгоритмы сжатия информации. Коды Хаффмана, алгоритм Хаффмана.	2	2				1	Самостоят. работа
	ИТОГО	52	60				10	

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ ДЛЯ СТУДЕНТОВ ЗАОЧНОЙ ФОРМЫ ОБУЧЕНИЯ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	Основные структуры данных и сложность алгоритмов	10	6					
1.1	Введение. Предмет теории алгоритмов. Прикладное значение эффективности алгоритмов. Связь с дискретной математикой, математической кибернетикой, программированием.	1						
1.2	Трудоемкость алгоритмов. Необходимость оценки алгоритмов. Принципы оценки трудоемкости комбинаторных алгоритмов. Асимптотики трудоемкости алгоритмов O , Ω , Θ . Трудоемкость «в худшем», трудоемкость «в среднем», трудоемкость «почти всегда». Примеры анализа алгоритмов.	1	1					Устный опрос
1.3	Основы теории сложности вычислений. Детерминированные и недетерминированные машины Тьюринга.	1	1					
1.4	Понятие о классах P и NP . NP -полные задачи. Теорема Кука–Карпа–Левина. NP -полнота задач «3-выполнимость», «Клика», «Гамильтонов цикл». Списки наиболее известных NP -полных задач.	2	1					
1.5	Алгоритмы сортировки. Сортировка выбором, вставками, пузырьковая, Шелла. Быстрая	1						Устный опрос

	сортировка.							
1.6	Сортировка слиянием. Внешняя сортировка. Линейные сортировки.	1	1					
1.7	Алгоритмы поиска и выборки. Последовательный поиск. Бинарный поиск.	1						Устный опрос
1.8	Деревья бинарного поиска. Сбалансированные деревья и реализуемые с их помощью структуры. Операции над деревьями.	1	1					
1.9	Хеширование. Открытая и закрытая адресация. Первичные и вторичные хеш-функции.	1	1					Устный опрос
2	Алгоритмы на графах	4	4					
2.1	Структуры данных для представления графов: матрицы смежности, матрицы инцидентности, списки смежности, списки ребер. Алгоритмы поиска в ширину и глубину, их реализация	1	1					
2.2	Поиск минимального остовного дерева и кратчайшего пути в графе. Алгоритмы Прима, Краскала.	1	1					
2.3	Поиск кратчайших расстояний в графе. Алгоритмы Дейкстры, Флойда, Беллмана–Форда.	1	1					Устный опрос
2.4	Алгоритм нахождения эйлера цикла.	1	1					
3	Алгоритмы на строках	2	2					
3.1	Основные определения. Алгоритмы поиска подстроки в строке. Алгоритм Бойера–Мура	1	1					
3.2	Алгоритм Кнута–Морриса–Пратта.	1	1					Устный опрос
4	Основные алгоритмические стратегии	2	2					
4.1	Алгоритмы локального поиска, поиска с запретами. Генетические алгоритмы, алгоритмы имитации отжига.	1	1					
4.2	Динамическое программирование и метод «разделяй и властвуй». Понятие о методах динамического программирования и «разделяй и властвуй».	1	1					Устный опрос
	ИТОГО	18	14					

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Литература

Основная

1. Ахо, А. Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. – М.: Мир, 1979. – 546 с.
2. Ахо, А. Построение и анализ вычислительных алгоритмов / А. Ахо, Дж. Хопкрофт, Дж. Ульман. – М.: Издательский дом «Вильямс», 2001. – 384 с.
3. Гэри, М. Вычислительные машины и труднорешаемые задачи / М. Гэри, Д. Джонсон. – М.: Мир, 1982. – 416 с.
4. Кормен, Т. Х. Алгоритмы. Построение и анализ / Т. Х. Кормен., Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн. – М.: Вильямс, 2005. – 1296 с.
5. Кнут, Д. Искусство программирования. Т. 1. Основные алгоритмы / Д. Кнут. – М.: Вильямс, 2006. – 720 с.
6. Кнут, Д. Искусство программирования. Т. 3. Сортировка и поиск / Д. Кнут. – М.: Вильямс, 2007. – 824 с.

Дополнительная

7. Бахтин, В. И. Исследование операций / В. И. Бахтин, А. П. Коваленок, А. В. Лебедев, Ю. В. Лысенко. – Мн.: БГУ, 2003. – 199 с.
8. Вирт, Н. Алгоритмы и структуры данных / Н. Вирт. – Санкт-Петербург: Невский диалект, 2001. – 352 с.
9. Емеличев, В.А. Лекции по теории графов / В.А. Емеличев, О.И. Мельников, В.И. Сарванов, Р.И. Тышкевич. – М.: Наука, 1990. – 384 с.
10. Кнут, Д. Искусство программирования. Т. 2. Получисленные алгоритмы / Д. Кнут. – М.: Вильямс, 2007. – 832 с.

Примерный перечень заданий для УСР

1. QuickSort

Реализуйте сортировку массива заданной размерности с помощью алгоритма быстрой сортировки. В качестве опорного элемента выберите:

- а) последний элемент массива,
- б) первый элемент массива,
- в) средний элемент массива,
- г) случайный элемент массива.

2. MergeSort

Реализуйте сортировку слиянием для массива заданной размерности.

3. Дерево бинарного поиска

Реализуйте структуру данных «дерево бинарного поиска». При этом обязательно должны присутствовать как базовые функции: добавить элемент,

удалить элемент, найти элемент, – так и дополнительные: ротировать два узла, вставить в корень, обойти дерево слева направо, просмотреть все дерево.

4. Хеш-таблица: разрешение коллизий методом цепочек

Реализуйте хеш-таблицу размерности 128, в которой коллизии разрешаются методом цепочек. Первичное хеширование осуществите при помощи функций:

а) $h(i) = i \bmod 128$,

б) $h(i) = \left\lceil 128 \left\{ (\sqrt{5} - 1)i/2 \right\} \right\rceil$.

При этом обязательно должны присутствовать три базовые функции: добавить элемент, удалить элемент, найти элемент, – а также функция просмотра всей таблицы.

5. Хеш-таблица с открытой адресацией

Реализуйте хеш-таблицу с открытой адресацией размерности 128, в которой осуществляется:

а) линейное пробирование,

б) квадратичное пробирование.

При этом обязательно должны присутствовать три базовые функции: добавить элемент, удалить элемент, найти элемент, – а также функция просмотра всей таблицы.

6. Граф: поиск в ширину и в глубину

Реализуйте неориентированный граф, заданный при помощи списка ребер и количества вершин, в виде:

а) списка смежности вершин,

б) матрицы смежности,

в) матрицы инцидентности.

Начиная с некоторой заданной вершины этого графа осуществите поиск в ширину и в глубину.

7. Эйлеров цикл

Для связного n -вершинного графа, заданного матрицей инцидентности, определите, эйлеров ли он. Если да, то выведите его эйлеров цикл.

8. Построение минимального остовного дерева

Для связного взвешенного n -вершинного графа, заданного матрицей инцидентности, реализуйте алгоритмы Краскала и Прима.

9. Алгоритм Дейкстры

Для связного взвешенного графа с неотрицательными весами ребер, заданного матрицей смежности, и некоторой его вершины реализуйте алгоритм Дейкстры.

10. Алгоритм Флойда

Для связного взвешенного графа с неотрицательными весами ребер, заданного матрицей смежности, реализуйте алгоритм Флойда.

11. Алгоритм Куна

Для связного двудольного графа с заданными долями X и Y найдите наибольшее паросочетание по алгоритму Куна.

12. Алгоритм Форда–Фалкерсона

Для транспортной сети $G = G(V, E)$, $V = \{v_0, v_1, \dots, v_n\}$, с истоком v_0 , стоком v_n , заданными пропускными способностями дуг $c_1, c_2, \dots, c_{|E|}$ и заданным допустимым потоком $f_1, f_2, \dots, f_{|E|}$ постройте поток максимальной мощности. В качестве выходных данных выведите величину потока максимальной мощности и разрез, который определяет этот поток.

13. Задачи динамического программирования

Проход по матрице штрафов

Задана матрица натуральных чисел $A = [a_{ij}]_{n \times m}$. За каждый проход через клетку матрицы с индексами (i, j) взимается штраф a_{ij} . Необходимо с минимальным суммарным штрафом пройти из первой клетки первой строки в клетку (n, m) . Движение по матрице осуществляется следующим образом: из текущей клетки с координатами (i, j) можно перейти в любую из трех соседних клеток (если они существуют): $(i+1, j)$, $(i+1, j-1)$, $(i+1, j+1)$. Найдите величину минимального суммарного штрафа.

Проход по матрице платежей

Задана матрица натуральных чисел $A = [a_{ij}]_{n \times m}$. За каждый проход через клетку матрицы с индексами (i, j) выплачивается вознаграждение a_{ij} . Необходимо найти максимальную сумму, которую можно получить, если двигаться из клетки $(1, 1)$ в клетку (n, m) . Движение по матрице осуществляется следующим образом: из текущей клетки с координатами (i, j) можно перейти в любую из двух соседних клеток (если они существуют): $(i+1, j)$, $(i, j+1)$.

Файловая система

В файловой системе персонального компьютера файлы организованы в каталоги. В компьютере полное имя файла является строкой, состоящей из имен каталогов и имени файла, разделенных символом "\", причем "\" не может быть первым, последним символом, а также идти два раза подряд. Имя файла (каталога) может быть произвольной длины, но длина полного имени файла не может быть длиннее n символов. В качестве символов, допустимых к употреблению в именах файлов (каталогов), могут использоваться символы из алфавита, состоящего из k букв (символ "\" не входит в их число). Необходимо для данных k ($1 \leq k \leq 13$) и n ($1 \leq n \leq 30$) определить максимальное число файлов, которое можно записать на данный компьютер.

Радиоактивные отходы

При переработке радиоактивных материалов образуются отходы двух видов: опасные и неопасные. Для их хранения используются одинаковые контейнеры.

После помещения отходов в контейнеры, последние укладываются одной вертикальной стопкой. Стопка считается опасной, если в ней подряд идет не менее трех опасных контейнеров, в противном случае стопка считается безопасной. Для заданного количества контейнеров $1 \leq n \leq 30$ определите число вариантов формирования безопасной стопки.

Системы счисления

Вычислите количество n -значных чисел в системе счисления с основанием k ($2 \leq k \leq 10$, $k + n \leq 18$) таких, что их запись не содержит двух подряд идущих нулей.

Локомотивы

На железнодорожной колее размещены n локомотивов с номерами от 1 до n , $1 \leq n \leq 16$. Они начинают двигаться в одну сторону, причем локомотив номер i движется со скоростью i м/с. Если локомотив, движущийся с большей скоростью, нагоняет более медленный локомотив, дальше они движутся один за другим со скоростью впереди идущего локомотива. Определите, сколько начальных расстановок дадут в результате ровно k ($1 \leq k \leq n$) групп движущихся локомотивов?

Блок из единиц

В таблице размера n на m , состоящей из 0 и 1, найдите квадратный блок максимального размера, состоящий только из 1.

Обмен валют

Матрица $A = [a_{ij}]_{n \times m}$ заполнена неотрицательными вещественными числами.

Число a_{ij} определяет курс обмена валюты i на валюту j . Так, например, если $a_{ij} = 4$, то это значит, что за единицу валюты i дают 4 единицы валюты j . Если $a_{ij} = 0$, то полагаем, что курс обмена валюты i на валюту j прямо не установлен. Необходимо определить, можно ли, имея некоторую сумму денег в одной из валют, получить большую сумму денег в той же валюте, совершив несколько обменов.

Лента

Задана лента шириной в одну клетку и длиной в n клеток. На каждой клетке написано некоторое целое число. Играют два игрока, которые ходят поочередно. За один ход игрок отрезает от ленты одну из крайних клеток и забирает ее. Игра останавливается, когда лента заканчивается. Выигрыш игрока равен сумме чисел находящихся у него клеток ленты. Необходимо определить

величину выигрыша, которую может гарантировать себе игрок, начинающий игру первым.

Игра в числа

Задается некоторое натуральное число $n \leq 998$. Двое играющих называют по очереди числа, меньшие 1000, по следующим правилам. Начиная с числа n , каждое новое число должно увеличивать одну из цифр предыдущего числа (возможно, незначащий нуль) на 1, 2 или 3. Проигравшим считается тот, кто называет число 999. Для заданного n необходимо определить, может ли выиграть игрок, делающий первый ход, при наилучших последующих ходах противника. В случае возможности выигрыша первым игроком, требуется напечатать все его возможные выигрышные первые ходы.

Удали до палиндрома

Вводится непустая строка, которая имеет длину не более 100 символов и состоит только из прописных латинских букв. Палиндромом называется такая строка символов, которая читается слева направо и справа налево одинаково. Необходимо удалить из строки минимальное количество символов так, чтобы получился палиндром. Этот палиндром нужно вывести на печать.

Добавь до палиндрома

Вводится непустая строка, которая имеет длину не более 100 символов и состоит только из прописных латинских букв. Палиндромом называется такая строка символов, которая читается слева направо и справа налево одинаково. Необходимо вставить в строку минимальное количество символов так, чтобы получился палиндром. Этот палиндром нужно вывести на печать.

Последовательная упаковка дисков

Имеется n дисков одинаковой толщины с радиусами $r_1, r_2, \dots, r_n$. Эти диски упаковываются в коробку таким образом, что каждый из них стоит ребром на дне коробки, и все диски находятся в одной плоскости. Для последовательности дисков необходимо найти длину коробки, в которую все диски могут быть упакованы, при условии, что диски можно упаковывать только в порядке их следования.

Произвольная упаковка дисков

Имеется n дисков одинаковой толщины с радиусами $r_1, r_2, \dots, r_n$. Эти диски упаковываются в коробку таким образом, что каждый из них стоит ребром на дне коробки, и все диски находятся в одной плоскости. Для последовательности дисков необходимо определить минимальную длину коробки, если порядок

упаковки дисков в коробку может быть произвольным (указать при этом порядок, в котором упаковываются диски).

Триангуляция многоугольника

Выпуклый n -угольник ($n \geq 3$) задается координатами своих вершин в порядке обхода по контуру. Необходимо разбить n -угольник на треугольники $n - 3$ диагоналями, не пересекающимися, кроме как в вершинах многоугольника, таким образом, чтобы сумма их длин была минимальной.

Триангуляция многоугольника с минимаксной диагональю

Выпуклый n -угольник ($n \geq 3$) задается координатами своих вершин в порядке обхода по контуру. Необходимо разбить n -угольник на треугольники $n - 3$ диагоналями, не пересекающимися, кроме как в вершинах многоугольника, таким образом, чтобы максимальная из диагоналей имела наименьшую длину.

Невозможная покупка

Покупатель имеет n купюр достоинством $a_1, a_2, \dots, a_n$ и продавец имеет m купюр достоинством $b_1, b_2, \dots, b_m$. Необходимо найти максимальную стоимость товара, которую покупатель не может купить, потому что нет возможности точно рассчитаться за этот товар с продавцом, хотя денег на покупку у него достаточно.

Точная сдача

Покупатель имеет n купюр достоинством $a_1, a_2, \dots, a_n$ и продавец имеет m купюр достоинством $b_1, b_2, \dots, b_m$. Необходимо определить, может ли покупатель приобрести вещь стоимости s так, чтобы у продавца нашлась точная сдача (если она необходима).

Железная дорога

Железная дорога с односторонним движением имеет n станций. Известны цены билетов от i -ой станции до j -ой. Найдите минимальную стоимость проезда от начала до конца с учетом возможной экономии за счет пересадок.

Возрастающая подпоследовательность без разрывов

Необходимо из заданной числовой последовательности a , состоящей из n элементов, вычеркнуть минимальное количество элементов так, чтобы оставшиеся элементы образовали строго возрастающую подпоследовательность элементов. Построенный алгоритм должен иметь трудоемкость $O(n \log n)$.

Возрастающая подпоследовательность с разрывом

Необходимо из заданной числовой последовательности a , состоящей из n элементов, вычеркнуть минимальное количество элементов так, чтобы в оставшейся подпоследовательности каждый последующий элемент был строго больше предыдущего, кроме не более одной пары соседних элементов (одного «разрыва»). Построенный алгоритм должен иметь трудоемкость $O(n \log n)$.

Выбор подпоследовательности

Из элементов заданной последовательности a целых чисел длины n необходимо выбрать элементы таким образом, чтобы они образовали подпоследовательность наибольшей длины, в которой каждый последующий элемент делился бы нацело на предыдущий. Порядок следования элементов можно менять.

Бинарная операция

Задано конечное непустое множество с бинарной операцией $\square$ (вообще говоря, некоммутативной и неассоциативной). Имеется n элементов $a_1, a_2, \dots, a_n$ этого множества и еще один элемент x . Проверьте, можно ли так расставить скобки в выражении $a_1 \square a_2 \square \dots \square a_n$, чтобы в результате получился x . Построенный алгоритм должен иметь трудоемкость $O(n^3)$.

Задача о рюкзаке

Имеется n положительных чисел $a_1, a_2, \dots, a_n$ – веса различных предметов, и число m – максимально допустимый вес рюкзака. Выясните, можно ли получить m , складывая некоторые из чисел $a_1, a_2, \dots, a_n$. Построенный алгоритм должен иметь трудоемкость $O(nm)$.

Преобразование строк

На вход подаются две символьные строки S_1 и S_2 . Необходимо преобразовать строку S_2 в строку S_1 с минимальным суммарным штрафом, который определяется следующим образом:

- штраф за удаление символа из строки S_2 равен a баллов;
- штраф за вставку символа в строку S_1 – b баллов;
- штраф за выполнение замены символа в строке S_2 на любой другой символ – c баллов.

14. Программирование на машине Тьюринга

а) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: массив из 0 и 1.

Выход: ответ «да» (1), если число 0 и число 1 отличаются не более, чем на 3; «нет» (0) – в противном случае.

б) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: массив из 0 и 1.

Выход: ответ «да» (1), если в массиве есть последовательность из трех подряд идущих 0; «нет» (0) – в противном случае.

в) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: массив из 0 и 1.

Выход: массив, записанный в обратном порядке.

г) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: массив из 0 и 1.

Выход: ответ «да» (1), если число 0 и число 1 отличаются на два; «нет» (0) – в противном случае.

д) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: массив из 0 и 1.

Выход: сумма элементов массива по модулю 2.

е) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: массив из символов a, b, c .

Выход: массив, из которого удалены все вхождения символа c .

ж) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: массив из 0, 1, 2, 3.

Выход: «да» (1), если если четных элементов больше, чем нечетных; «нет» (0) – в противном случае.

з) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: массив из 0 и 1.

Выход: ответ «да» (1), если сумма элементов массива делится на три; «нет» (0) – в противном случае.

и) Напишите программу для машины Тьюринга, реализующую следующее преобразование.

Вход: натуральное число x в унарной системе счисления.

Выход: число $x + 5$.

15. Одномерная упаковка

Дано множество $A = \{a_1, a_2, \dots, a_n\}$ предметов и их веса (длины) $l(a_i) \in (0, 1)$, $i \in N_n = \{1, 2, \dots, n\}$. Нужно найти разбиение множества A на минимальное число подмножеств (контейнеров) $A_1, A_2, \dots, A_m$ такое, что для каждого контейнера A_j , $j \in N_m$, выполняется ограничение по вместимости $\sum_{a_i \in A_j} l(a_i) \leq 1$.

16. Двумерная упаковка

Дано множество $A = \{a_1, a_2, \dots, a_n\}$ прямоугольных предметов длины $l(a_i) \in (0, 1)$ и ширины $w(a_i) \in (0, 1)$, $i \in N_n = \{1, 2, \dots, n\}$. Поворачивать предметы нельзя. Стороны длины $l(a_i)$ параллельны оси абсцисс, а стороны ширины $w(a_i)$ параллельны оси ординат. Необходимо найти упаковку

а) в положительном квадранте с минимальной площадью охватывающего прямоугольника;

б) в полубесконечную полосу ширины 1 с минимальной длиной занимаемой полосы;

в) в минимальное число контейнеров размера 1×1 .

17. Раскраски

Дан n -граф G с множеством вершин $VG = \{v_1, v_2, \dots, v_n\}$. Нужно найти его хроматическое число $\chi(G)$.

Перечень используемых средств диагностики результатов учебной деятельности

С целью текущего контроля предусматривается проведение контрольных и самостоятельных работ, устные опросы.

По итогам 5-го семестра у студентов очной формы обучения проводится зачет, по итогам и 6-го семестра – экзамен.

С целью текущего контроля студентов полной заочной формы обучения предусматривается написание контрольных работ в 4-м и 5-м семестре. По итогам 4-го семестра проводится зачет, 5-го семестра – экзамен.

Студенты заочной формы обучения на основе среднего специального образования пишут контрольные работы в 2-м и 3-м семестре и сдают зачет во 2-м семестре и экзамен в 3-м семестре.

Рекомендуемые темы контрольных работ

1. Асимптотики трудоемкости алгоритмов; доказательство NP -полноты алгоритмических задач; основные структуры данных и операции над ними; основные алгоритмы на графах: Прима, Краскала, поиск кратчайших расстояний в графе, нахождения эйлера цикла и паросочетания в двудольных графах.
2. Динамическое программирование и метод «разделяй и властвуй»; формулировки основных задач дискретной оптимизации как задач целочисленного линейного программирования; алгоритмы для работы с матрицами; LU -разложение.

Организация самостоятельной работы студентов

Самостоятельная работа студентов – это любая деятельность, связанная с воспитанием мышления будущего профессионала. В широком смысле под самостоятельной работой следует понимать совокупность всей самостоятельной деятельности студентов, как в учебной аудитории, так и вне её, в контакте с преподавателем или в его отсутствии.

Самостоятельная работа реализуется:

1. Непосредственно в процессе аудиторных занятий на лекциях и практических занятиях.
2. В контакте с преподавателем вне рамок расписания на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.

3. В библиотеке, дома, в общежитии, на кафедре при выполнении студентом учебных и творческих задач.

При изучении дисциплины организация самостоятельной работы студентов должна представлять единство трех взаимосвязанных форм:

1. Внеаудиторная самостоятельная работа;
2. Аудиторная самостоятельная работа, которая осуществляется под непосредственным руководством преподавателя;
3. Творческая, в том числе научно-исследовательская работа.

Виды внеаудиторной самостоятельной работы студентов могут принимать разнообразные формы: изучение рекомендуемой литературы по дисциплине, подготовка и написание рефератов, докладов, очерков и других письменных работ на заданные темы.

Аудиторная самостоятельная работа может реализовываться при проведении практических занятий, а также во время чтения лекций.

При чтении лекционного курса непосредственно в аудитории иногда возникает необходимость проконтролировать усвоение материала основной массой студентов путем проведения экспресс-опросов по конкретным темам.

На практических занятиях нужно значительную часть времени отводить на самостоятельное решение задач. Типичное практическое занятие целесообразно строить следующим образом:

1. Вводное слово преподавателя (цели занятия, основные вопросы, которые должны быть рассмотрены).
2. Беглый опрос.
3. Решение 1 – 2 типовых задач (с вызовом студента к доске).
4. Самостоятельное решение задач.
5. Разбор типовых ошибок при решении (в конце текущего занятия или в начале следующего).

Результативность самостоятельной работы студентов во многом определяется наличием активных методов ее контроля. В рамках дисциплины существуют следующие виды контроля:

- входной контроль знаний и умений студентов в начале изучения дисциплины;
- текущий контроль, то есть регулярное отслеживание уровня усвоения материала на лекциях и практических занятиях;
- промежуточный контроль по окончании изучения раздела курса;
- самоконтроль, осуществляемый студентом в процессе изучения дисциплины при подготовке к контрольным мероприятиям;
- итоговый контроль по дисциплине в виде экзамена.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

[illegible]

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УВО

на ____/____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры математической кибернетики (протокол № ____ от _____ 201_ г.)

Заведующий кафедрой,
кандидат физ.-мат. наук, доцент

В.С. Романчик

УТВЕРЖДАЮ

Декан факультета,
кандидат физ.-мат. наук, доцент

Д.Г. Медведев

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ
на 2016/2017 учебный год

№ пп	Дополнения и изменения	Основание
	В пояснительную записку внести требования к компетенциям специалиста: академические – АК-1, АК-2, АК-3, АК-4, АК-5, АК-6, АК-7, АК-8, АК-9; социально-личностные – СЛК-1, СЛК-2, СЛК-3, СЛК-4, СЛК-5, СЛК-6; профессиональные – ПК-1, ПК-2, ПК-3, ПК-4, ПК-5, ПК-6, ПК-7, ПК-8, ПК-9, ПК-10, ПК-11, ПК-12, ПК-13, ПК-14, ПК-15, ПК-16, ПК-17, ПК-18, ПК-19, ПК-20, ПК-21, ПК-22, ПК-23, ПК-24, ПК-25, ПК-26, ПК-27.	

Учебная программа пересмотрена и одобрена на заседании кафедры веб-технологий и компьютерного моделирования (протокол № 9 от 27.05.2016г.)

Заведующий кафедрой

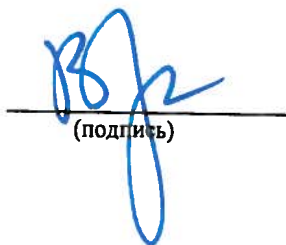
канд. физ.-мат. наук, доцент
(ученая степень, ученое звание)


(подпись)

В.С. Романчик
(И.О.Фамилия)

394 УТВЕРЖДАЮ
Декан факультета

канд. физ.-мат. наук, доцент
(ученая степень, ученое звание)


(подпись)

Д.Г. Медведев
(И.О.Фамилия)