Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 09.11.2023 16:23:24
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

УТВЕРЖДАЮ

Декан факультета
информационных технологий
/Д. Г. Демидов/

"28" апреля 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Олимпиадное программирование»

Направление подготовки
10.05.03 «Информационная безопасность автоматизированных систем»

Образовательная программа (профиль)
«Безопасность открытых информационных систем»

Квалификация (степень) выпускника
Специалист по защите информации

Форма обучения
Очная

Москва 2022 г.

Разработчик(и):
К.т.н., доцент /И.В. Калуцкий/

Согласовано:
И.о. заведующего кафедрой «Информационная безопасность», А.Ю. Гневшев

Руководитель образовательной программы, А.Ю. Гневшев

1. Цели освоения дисциплины.

К **основным целям** освоения дисциплины «Олимпиадное программирование» следует отнести:

- изучение методов командной разработки алгоритмов для решения сложных вычислительных задач и оценивания их эффективности.

К **основным задачам** освоения дисциплины «Олимпиадное программирование» следует отнести:

- дать знания о существующих эффективных алгоритмах для решения наиболее известных
- задач комбинаторной оптимизации, об их сложности и требованиям к памяти;
- познакомить с классификацией оптимизационных задач и алгоритмов для их решения,
- особенностями задач комбинаторной оптимизации большой размерности;
- привить навыки работы в команде при разработке алгоритмов и программных комплексов для решения сложных вычислительных задач.

2. Место дисциплины в структуре ООП.

Дисциплина «Олимпиадное программирование» относится к числу факультативных дисциплин образовательной программы.

Изучение дисциплины опирается на знания, умения и навыки, приобретенные в предшествующих дисциплинах: Языки программирования.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы.

В результате освоения дисциплины (модуля) у обучающихся формируются следующие компетенции и должны быть достигнуты следующие результаты обучения как этап формирования соответствующих компетенций:

Код компетенции	**В результате освоения образовательной программы обучающийся должен обладать**	**Перечень планируемых результатов обучения по дисциплине**
ОПК-7	Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности	**Знать:** • различные методы классификации существующих алгоритмов; • наиболее известные алгоритмы для работы с различными структурами данных; • особенности точных, приближенных, эвристических, переборных, «жадных» алгоритмов. **Уметь:** • анализировать существующие алгоритмы с точки зрения их эффективности и • применимости для решения прикладных задач; • разрабатывать новые алгоритмы для решения конкретных задач в области программной инженерии;

		• оценивать сложность разработанных алгоритмов и обосновывать их корректность. **владеть;** • навыками применения известных и разработки собственных алгоритмов для решения • навыками решения практических задач с учетом требований к точности, времени работы алгоритма и вычислительным ресурсам; • навыками командной разработки алгоритмов для решения сложных вычислительных задач и оценивания их эффективности; • навыками формализации и разработки математических моделей и алгоритмов для решения конкретных практических проблем в сфере программной инженерии или их сведения к известным модельным задачам.

4. Структура и содержание дисциплины.

Общая трудоемкость дисциплины составляет **72** академических часов (лабораторные занятия – 36 час, самостоятельная работа - 36 часов, форма контроля – зачет) в 1 и 2 семестре.

В 1 семестре - 36 академических часов (лабораторные занятия – 18 час, самостоятельная работа - 18 часов, форма контроля – зачет).

В 2 семестре - 36 академических часов (лабораторные занятия – 18 час, самостоятельная работа - 18 часов, форма контроля – зачет).

Структура и содержание дисциплины «Олимпиадное программирование» по срокам и видам работы отражены в приложении.

Содержание дисциплины

Тема 1. Классы алгоритмов (точные, приближенные, эффективные, переборные, рекурсивные).

Понятие алгоритма, требования к алгоритму, классы алгоритмов (точные, приближенные, эффективные, переборные, рекурсивные), примеры алгоритмов из разных классов для решения задач оптимизации, анализ их сложности и корректности.

Тема 2. Структуры данных.

Основные абстрактные типы данных, структуры данных (массив, список, стек, очередь, дерево), их реализация в различных языках программирования, примеры использования в стандартных алгоритмах.

Тема 3. Разработка алгоритмов и программ для решения задач по теме «Графы».

Основные алгоритмы для работы с графами (Прима, Краскала, Дейкстры, Флойда, венгерский, «жадные»), оптимизационные задачи на графы (минимальное остовное дерево, максимальное паросочетание, максимальные поток минимальной стоимости, задача о назначениях).

Тема 4. Разработка алгоритмов и программ для решения задач по теме «Рекурсия и динамическое программирование».

Рекуррентные соотношения, их аналитическое решение и программная реализация, динамическое программирование и его связь с рекуррентными соотношениями, решение задач на составление рекуррентных соотношений, решение оптимизационных задач методом динамического программирования.

Тема 5. Разработка алгоритмов и программ для решения задач по теме «Сортировка и поиск».

Алгоритмы сортировки (сортировка вставками, быстрая сортировка, сортировка слиянием, цифровая сортировка), алгоритмы поиска порядковых статистик, поиск медианы, решение задач на сортировку и поиск.

Тема 6 Разработка алгоритмов и программ для решения задач по теме «Комбинаторика».

Комбинаторные операции (перестановки, сочетания, размещения) и принципы (сложения, умножения, дополнения, включения-исключения, кодирования), алгоритмы генерации комбинаторных объектов и быстрого вычисления числа сочетаний, решение задач на комбинаторику.

Тема 7 Разработка алгоритмов и программ для решения задач по теме «Длинная арифметика».

Операции модулярной арифметики, теорема Ферма, китайская теорема об остатках, системы исчисления с произвольным основанием, моделирование сложения, умножения и деления «длинных» чисел с помощью массивов, решение задач на «длинную арифметику».

5. Образовательные технологии.

Методика преподавания дисциплины «Олимпиадное программирование и» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся:

- подготовка к выполнению лабораторных работ;

Удельный вес занятий, проводимых в интерактивных формах по дисциплине, составляет 20 % аудиторных занятий

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов.

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- зачет.

Образцы заданий к зачету приведены в приложении.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю).

6.1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.
В результате освоения дисциплины (модуля) формируются следующие компетенции:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать
ОПК-7	Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.2. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины (модуля), описание шкал оценивания

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю):

ОПК-7 Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности				
Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
знать: •различные методы классификации существующих алгоритмов; •наиболее известные алгоритмы для работы с различными структурами данных; •особенности точных, приближенных, эвристических, переборных, «жадных» алгоритмов.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: . •различные методы классификации существующих алгоритмов; •наиболее известные алгоритмы для работы с различными структурами данных; •особенности точных, приближенных, эвристических, переборных,	Обучающийся демонстрирует неполное соответствие следующих знаний: •различные методы классификации существующих алгоритмов; •наиболее известные алгоритмы для работы с различными структурами данных; •особенности точных, приближенных, эвристических, переборных, «жадных» алгоритмов. Допускаются значительные ошибки, проявляется недостаточность	Обучающийся демонстрирует частичное соответствие следующих знаний: •различные методы классификации существующих алгоритмов; •наиболее известные алгоритмы для работы с различными структурами данных; •особенности точных, приближенных, эвристических, переборных, «жадных»	Обучающийся демонстрирует полное соответствие следующих знаний: •различные методы классификации существующих алгоритмов; •наиболее известные алгоритмы для работы с различными структурами данных; •особенности точных, приближенных, эвристических, переборных, «жадных» алгоритмов. , свободно оперирует

	«жадных» алгоритмов.	знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	алгоритмов. , но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	приобретенными знаниями.
уметь: •анализировать существующие алгоритмы с точки зрения их эффективности и •применимости для решения прикладных задач; •разрабатывать новые алгоритмы для решения конкретных задач в области программной инженерии; •оценивать сложность разработанных алгоритмов и обосновывать их корректность.	Обучающийся не умеет или в недостаточной степени умеет •анализировать существующие алгоритмы с точки зрения их эффективности и •применимости для решения прикладных задач; •разрабатывать новые алгоритмы для решения конкретных задач в области программной инженерии; •оценивать сложность разработанных алгоритмов и обосновывать их корректность.	Обучающийся демонстрирует неполное соответствие следующих умений: . •анализировать существующие алгоритмы с точки зрения их эффективности и •применимости для решения прикладных задач; •разрабатывать новые алгоритмы для решения конкретных задач в области программной инженерии; •оценивать сложность разработанных алгоритмов и обосновывать их корректность. Допускаются значительные ошибки, проявляется недостаточность умений.	Обучающийся демонстрирует частичное соответствие следующих умений: •анализировать существующие алгоритмы с точки зрения их эффективности и •применимости для решения прикладных задач; •разрабатывать новые алгоритмы для решения конкретных задач в области программной инженерии; •оценивать сложность разработанных алгоритмов и обосновывать их корректность. Умения освоены, но допускаются незначительные ошибки, неточности.	Обучающийся демонстрирует полное соответствие следующих умений: •анализировать существующие алгоритмы с точки зрения их эффективности и •применимости для решения прикладных задач; •разрабатывать новые алгоритмы для решения конкретных задач в области программной инженерии; •оценивать сложность разработанных алгоритмов и обосновывать их корректность. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
владеть: •навыками применения известных и разработки собственных алгоритмов для решения •навыками решения практических задач с учетом требований к точности, времени работы алгоритма и вычислительным ресурсам; •навыками командной	Обучающийся не владеет или в недостаточной степени владеет •навыками применения известных и разработки собственных алгоритмов для решения •навыками решения практических задач с учетом требований к точности, времени работы алгоритма и	Обучающийся владеет •навыками применения известных и разработки собственных алгоритмов для решения •навыками решения практических задач с учетом требований к точности, времени работы алгоритма и вычислительным ресурсам; •навыками командной разработки алгоритмов для решения сложных вычислительных задач	Обучающийся частично владеет •навыками применения известных и разработки собственных алгоритмов для решения •навыками решения практических задач с учетом требований к точности, времени работы алгоритма и вычислительным ресурсам;	Обучающийся в полном объеме владеет •навыками применения известных и разработки собственных алгоритмов для решения •навыками решения практических задач с учетом требований к точности, времени работы алгоритма и вычислительным ресурсам;

разработки алгоритмов для решения сложных вычислительных задач и оценивания их эффективности; •навыками формализации и разработки математических моделей и алгоритмов для решения конкретных практических проблем в сфере программной инженерии или их сведения к известным модельным задачам.	вычислительным ресурсам; •навыками командной разработки алгоритмов для решения сложных вычислительных задач и оценивания их эффективности; •навыками формализации и разработки математических моделей и алгоритмов для решения конкретных практических проблем в сфере программной инженерии или их сведения к известным модельным задачам.	и оценивания их эффективности; •навыками формализации и разработки математических моделей и алгоритмов для решения конкретных практических проблем в сфере программной инженерии или их сведения к известным модельным задачам, но допускаются значительные ошибки, проявляется недостаточность владения	•навыками командной разработки алгоритмов для решения сложных вычислительных задач и оценивания их эффективности; •навыками формализации и разработки математических моделей и алгоритмов для решения конкретных практических проблем в сфере программной инженерии или их сведения к известным модельным задачам, навыки освоены, но допускаются незначительные ошибки, неточности, затруднения.	•навыками командной разработки алгоритмов для решения сложных вычислительных задач и оценивания их эффективности; •навыками формализации и разработки математических моделей и алгоритмов для решения конкретных практических проблем в сфере программной инженерии или их сведения к известным модельным задачам, свободно применяет полученные навыки в ситуациях повышенной сложности.

Шкалы оценивания результатов промежуточной аттестации и их описание:

Форма промежуточной аттестации: зачет

Промежуточная аттестация обучающихся в форме зачёта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка **«зачтено»** или **«не зачтено»**.

Шкала оценивания	**Описание**
Зачтено	Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков, приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

Не зачтено	Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков, приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

Фонды оценочных средств представлены в приложении к рабочей программе.

7. Учебно-методическое и информационное обеспечение дисциплины.

а) основная литература:

- Синицын, С.В. Основы разработки программного обеспечения на примере языка С / С.В. Синицын, О.И. Хлытчиев. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 212 с. : схем., ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=429186 (дата обращения: 18.08.2019). – Библиогр. в кн. – Текст : электронный.
- Объектно-ориентированное программирование с примерами на C# : учеб. пособие / П.Б. Хорев. — М. : ФОРУМ : ИНФРА-М, 2019. — 200 с. — (Высшее образование: Бакалавриат)
- Программирование на языке С++ в среде Qt Creato / Е.Р. Алексеев, Г.Г. Злобин, Д.А. Костюк и др. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 716 с. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=428929 (дата обращения: 18.08.2019). – Библиогр. в кн. – Текст : электронный.
- Абрамян, М.Э. Введение в стандартную библиотеку шаблонов С++. Описание, примеры использования, учебные задачи: учебник по курсу «Стандартная библиотека С++» для студентов направления 02.03.02 «Фундаментальная информатика и информационные технологии» (бакалавриат) / М.Э. Абрамян ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет». – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2017. – 179 с. : ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=499454 (дата обращения: 18.08.2019). – Библиогр. в кн. – ISBN 978-5-9275-2374-0. – Текст : электронный.

б.) Дополнительная литература:

- Сорокин, А.А. Объектно-ориентированное программирование : учебное пособие (курс лекций) / А.А. Сорокин ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». – Ставрополь : СКФУ, 2014. – 174 с. : ил. – Режим доступа: по подписке. – URL:

http://biblioclub.ru/index.php?page=book&id=457696 (дата обращения: 18.08.2019). – Текст : электронный.

- Хиценко, В.П. Основы программирования : учебное пособие / В.П. Хиценко ; Министерство образования и науки Российской Федерации, Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2015. – 83 с. : схем. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=438365 (дата обращения: 18.08.2019). – Библиогр. в кн. – ISBN 978-5-7782-2706-4. – Текст : электронный.

в) программное обеспечение и интернет-ресурсы:

Средства разработки, тестирования, отладки программ, написанных на языке C++.
Система контроля стиля программирования Style Checker;
LMS, как основа для организации дистанционной поддержки дисциплины;

8. Материально-техническое обеспечение дисциплины.

Для проведения всех видов занятий необходимо презентационное оборудование (мультимедийный проектор, экран) – 1 комплект.

Для проведения лабораторных занятий необходимо наличие компьютерных классов оборудованных современной вычислительной техникой из расчета одно рабочее место на одного обучаемого.

9. Методические рекомендации для самостоятельной работы студентов

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются лекции.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, готовятся к экзамену, а также самостоятельно изучают отдельные темы учебной программы.

10. Методические рекомендации для преподавателя

При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

Приложение.

Структура и содержание дисциплины «Олимпиадное программирование»
по направлению подготовки
10.03.01 «Информационная безопасность»
(бакалавр)

n/n	Раздел	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы студентов					Формы аттестации	
				Л	П/С	Лаб	СРС	КСР	К.Р.	К.П.	ДЗ	Реферат	К/р	Э	З
	1 семестр														
1	Тема 1. Классы алгоритмов (точные, приближенные, эффективные, переборные, рекурсивные).	1	1-2			4									
2	Тема 2. Структуры данных.		3-4			4									
3	Тема 3. Разработка алгоритмов и программ для решения задач по теме «Графы».		5-10			14									
4	Тема 4. Разработка алгоритмов и программ для решения задач по теме «Рекурсия и динамическое программирование».		11-18			14									
	Форма аттестации		19-21							.				З	
	2 семестр														
1	Тема 5. Разработка алгоритмов и программ для решения задач по теме	2	1-6			12									

	«Сортировка и поиск».														
2	Разработка алгоритмов и программ для решения задач по теме «Комбинаторика».		7-12			12									
3	Тема 7 Разработка алгоритмов и программ для решения задач по теме «Длинная арифметика».		13-18			12									
	Форма аттестации		19-21							.				З	
	Всего часов по дисциплине					72									

Таблица 1

ПОКАЗАТЕЛЬ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Олимпиадное программирование					
ФГОС ВО **10.03.01 «Информационная безопасность»**					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **общепрофессиональные и профессиональные компетенции**:					
КОМПЕТЕНЦИИ		**Перечень компонентов**	**Технология формирования компетенций**	**Форма оценоч-ного**	**Степени уровней освоения компетенций**
ИН-ДЕКС	**ФОРМУЛИРОВКА**				

ОПК-7	Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности	**Знать:** • о различных методах классификации существующих алгоритмов; • наиболее известные алгоритмы для работы с различными структурами данных; • особенности точных, приближенных, эвристических, переборных, «жадных» • алгоритмов. **Уметь:** • анализировать существующие алгоритмы с точки зрения их эффективности и • применимости для решения прикладных задач; • разрабатывать новые алгоритмы для решения конкретных задач в области • программной инженерии; • оценивать сложность разработанных алгоритмов и обосновывать их корректность. **владеть;** • навыками применения известных и разработки собственных алгоритмов для решения • навыками решения практических задач с учетом требований к точности, времени работы алгоритма ивычислительным ресурсам; • навыками командной разработки алгоритмов для решения сложных вычислительных задач и оценивания их эффективности; • навыками формализации и разработки математических моделей и алгоритмов для решения конкретных практических проблем в сфере программной инженерии или их сведения к известным модельным задачам.	, лабораторные занятия	зачет	Базовый уровень: Знать: • о различных методах классификации существующих алгоритмов; • наиболее известные алгоритмы для работы с различными структурами данных; Уметь: • анализировать существующие алгоритмы с точки зрения их эффективности и • применимости для решения прикладных задач; • разрабатывать новые алгоритмы для решения конкретных задач в области программной инженерии; владеть; • навыками применения известных и разработки собственных алгоритмов для решения • навыками решения практических задач с учетом требований к точности, времени работы алгоритма и вычислительным ресурсам; • навыками командной разработки алгоритмов для решения сложных вычислительных задач и оценивания их эффективности; Повышенный уровень: Знать: • особенности точных, приближенных, эвристических, переборных, «жадных» алгоритмов. Уметь: •оценивать сложность разработанных алгоритмов и обосновывать их корректность. владеть; навыками формализации и разработки математических моделей и алгоритмов для решения конкретных практических проблем в сфере программной инженерии или их сведения к известным модельным задачам.

Приложение 2
к рабочей программе

Оценочные средства для промежуточной аттестации
Зачет.
Список вопросов для зачета по дисциплине

1. Какие свойства относятся к основным свойствам алгоритма?
2. Дан одномерный массив из N элементов. Какова оценка сложности процедуры его сортировки методом «пузырька» (сортировка простого обмена)?
3. Какова оценка сложности быстрой рекурсивной процедуры сортировки массива?
4. Даны две квадратные матрицы размера 3x3, для вычисления их произведения используется стандартный алгоритм. Сколько операторов умножения будет выполнено при вычислении произведения?
5. Граф задан матрицей расстояний. Какова длина его минимального остовного дерева?
6. Граф задан матрицей расстояний. Какова длина его кратчайшего гамильтонова цикла, полученного применением «жадного» алгоритма?
7. Двудольный граф задан списком ребер. Каково количество ребер в максимальном паросочетании в этом графе?
8. Транспортная сеть задана матрицей расстояний. Какова величина максимального потока в этой сети?
9. Граф задан матрицей расстояний. Какова длина кратчайшего пути между вершинами 1 и N
10. Какие из перечисленных ниже задач относятся к классу NP-полных?
    а) нахождение минимального остовного дерева;
    б) нахождение кратчайшего гамильтонова цикла;
    в) нахождение минимально потока в сети;
    г) нахождение хроматического числа графа;
    д) определение изоморфности двух графов.
11. Определить количество различных способов раскраски вершин полного бинарного дерева высоты 3 в 2 цвета (с точностью до переобозначения потомков).
12. Задача «о рюкзаке» решается методом ветвей и границ с использованием переменной степени ветвления. Каковы будут оценки вершин на первом шаге алгоритма (после проведения первого ветвления)?
13. Какие из перечисленных ниже задач относятся к числу алгоритмически неразрешимых?
    а) перемножение двух квадратных матриц;
    б) нахождение кратчайшего гамильтонова цикла;
    в) задача «о переносе Ханойской башни»;
    г) решение задачи останова (остановится ли заданная машина Тьюринга на заданном наборе входных данных);
    д) определение эквивалентности двух алгоритмов.
14. Имеется одномерный массив из N элементов, требуется найти в нем максимальный элемент. Насколько быстрее поставленная задача может быть выполнена на параллельном компьютере с использованием параллельного алгоритма по сравнению с последовательным алгоритмом?
15. В чем заключается «венгерский алгоритм»?