

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И. Носова»



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

*АЛГОРИТМЫ НА СЕТЯХ И ГРАФАХ*

Направление подготовки  
09.03.01 Информатика и вычислительная техника

Профиль программы  
Программное обеспечение средств вычислительной техники и  
автоматизированных систем

Уровень высшего образования – бакалавриат

Программа подготовки – академический бакалавриат

Форма обучения  
Заочная

Институт  
Кафедра  
Курс

*энергетики и автоматизированных систем  
вычислительной техники и программирования*  
3

Магнитогорск  
2017 г.

Рабочая программа составлена на основе ФГОС ВО по направлению подготовки (специальности) 09.03.01 Информатика и вычислительная техника, утвержденного приказом МО и Н РФ от 12.01.2016 № 5.

Рабочая программа рассмотрена и одобрена на заседании кафедры вычислительной техники и программирования « 26 » октября 2017 г., протокол № 2.

Зав. кафедрой  / О.С. Логунова/

Рабочая программа одобрена методической комиссией института энергетики и автоматизированных систем « 27 » октября 2017 г., протокол № 2.

Председатель  / С.И. Лукьянов/


Рабочая программа составлена:

доцентом кафедры вычислительной техники и программирования

 / С.И. Файнштейн /

Рецензент:

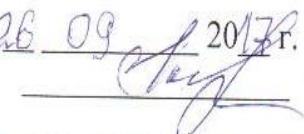
начальник отдела инновационных разработок  
ЗАО «КонсОмСКС», канд. техн. наук

 / А.Н. Панов/

## Лист актуализации рабочей программы

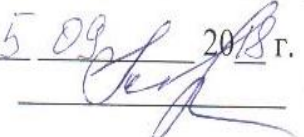
---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2017-2018 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от 26.09 2017 г. № 2  
Зав. кафедрой  О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2018 - 2019 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от 5.09 2018 г. № 1  
Зав. кафедрой  О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2019 - 2020 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от 19.02 2020 г. № 5  
Зав. кафедрой  О.С. Логунова

---

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2020 - 2021 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от 19.02 2021 г. № 5  
Зав. кафедрой  О.С. Логунова

## 1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) «Алгоритмы на сетях и графах» является ознакомление студентов с базовыми понятиями современной теории графов, изучение алгоритмов, используемых при решении задач в сетевых постановках, сравнительный анализ алгоритмов по вычислительной сложности.

Для достижения поставленной цели в курсе «Алгоритмы на сетях и графах» решаются задачи:

- изучение и классификация основных понятий теории графов;
- исследование различных объектов и подструктур в графах;
- освоение методов и алгоритмов решения классических задач на сетях и графах;
- приобретение навыков сетевых постановок задач и анализа трудоёмкости алгоритмов, используемых для их решения.

## 2 Место дисциплины в структуре образовательной программы подготовки бакалавра

Дисциплина входит в вариативную часть блока 1 образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения следующих курсов: информатика, прикладное программирование, математическая логика.

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения следующих дисциплин:

– «Алгоритмы и теория сложности» – при изучении этой дисциплины понадобится умение решать NP-полные задачи алгоритмом с возвратом. Начальные представления о вычислительной сложности алгоритмов и классификации задач по степени сложности будут расширены и углублены;

– «Логическое программирование» - умение работать с графами и проводить логический вывод с помощью алгоритма с возвратом.

Умения и владения, полученные при изучении дисциплины «Алгоритмы на сетях и графах», позволят обучающимся использовать сетевые модели и грамотно делать математическую постановку при выполнении выпускной квалификационной работы в её алгоритмической части.

## 3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Логическое программирование» обучающийся должен обладать следующими компетенциями:

Структурный элемент компетенции	Планируемые результаты обучения
<b>ПК-2 Способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования</b>	
Знать	основные понятия теории графов; основные алгоритмы на сетях и графах; классификацию задач по степени сложности.
Уметь	делать сетевые постановки модельных задач; использовать и программировать основные алгоритмы на сетях и графах; решать NP-полные задачи небольшой размерности алгоритмом с возвратом.
Владеть	навыками сетевых постановок для практических задач.



Раздел/ тема Дисциплины	Курс	Аудиторная контактная работа (в acad. часах)			Самостоятельная ра- бота (в acad. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
2.1 Методы систематического обхода графов: поиск в глубину, поиск в ширину, стягивающие деревья	3	0,5	1		10	1. Подготовка к лабораторным занятиям, работа с электронным учебником. 2. Выполнение лабораторной работы, написание программы на языке Prolog. 3. Самостоятельное изучение учебной и научной литературы	1. <i>Беседа - обсуждение</i> 2. <i>Проверка индивидуальных заданий</i> 3. <i>Устный опрос.</i>	ПК-2 – зுவ
2.2 Фундаментальные циклы, блоки и точки сочления, циклы Эйлера.	3	0,5	1		14	1. Подготовка к лабораторным занятиям, работа с электронным учебником. 2. Выполнение лабораторной работы. 3. Самостоятельное изучение учебной и научной литературы	1. <i>Беседа - обсуждение</i> 2. <i>Проверка индивидуальных заданий</i> 3. <i>Устный опрос.</i>	ПК-2 – зுவ
2.3 Классификация задач по степени сложности. Алгоритм с возвратом. Задача коммивояжера	3	0,5	–		18	Самостоятельное изучение учебной и научной литературы, работа с электронным учебником.	1. <i>Беседа - обсуждение</i> 3. <i>Устный опрос</i>	ПК-2 – зுவ
Итого по разделу		<b>1,5</b>	<b>2</b>		<b>42</b>		<b>Проверка индивидуальных заданий</b>	
3. Алгоритмы на взвешенных ориентированных графах								
3.1 Общая постановка задачи о поиске кратчайшего пути. Алгоритм Форда–	3	0,5	1(1И)		14	1. Самостоятельное изучение учебной и научной литерату-	1. <i>Беседа - обсуждение</i> 2. <i>Проверка индивидуаль-</i>	ПК-2 – зுவ

Раздел/ тема Дисциплины	Курс	Аудиторная контактная работа (в акад. часах)			Самостоятельная ра- бота (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
Беллмана. Алгоритм Дейкстры						ры, работа с электронным учебником. 2. Выполнение лабораторной работы	<i>ных заданий</i> 3. <i>Устный опрос.</i>	
3.2 Алгоритм Флойда. Оптимальное размещение различных типов обслуживающих центров на сильносвязных ориентированных нагруженных графах с целыми положительными весами дуг и вершин	3	1	–		19,4	1. Самостоятельное изучение учебной и научной литературы, работа с электронным учебником. 2. Выполнение курсовой работы	1. <i>Беседа – обсуждение.</i> <i>Проверка курсовой работы.</i> 3. <i>Устный опрос.</i>	ПК-2 – зுவ
Итого по разделу		<b>1,5</b>	<b>1</b>		<b>33,4</b>		<b><i>Проверка курсовой работы</i></b>	
<b>Итого за семестр</b>		<b>4</b>	<b>4(2И)</b>		<b>95,4</b>		<b><i>Зачёт, курсовая работа</i></b>	
<b>Итого по дисциплине</b>		<b>4</b>	<b>4(2И)</b>		<b>95,4</b>		<b><i>Зачёт, курсовая работа</i></b>	

## 5 Образовательные и информационные технологии

1. **Традиционные образовательные технологии**, ориентированные на организацию образовательного процесса и предполагающую прямую трансляцию знаний от преподавателя к аспиранту.

### **Формы учебных занятий с использованием традиционных технологий:**

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. **Технологии проблемного обучения** – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности аспирантов.

3. **Интерактивные технологии** – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе лично значимого для них образовательного результата.

### **Формы учебных занятий с использованием специализированных интерактивных технологий:**

Лекция «обратной связи» – лекция–провокация (изложение материала с заранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция-прессконференция.

4. **Информационно-коммуникационные образовательные технологии** – организация образовательного процесса, основанная на применении программных сред и технических средств работы с знаниями в различных предметных областях.

## 6 Учебно-методическое обеспечение самостоятельной работы обучающихся

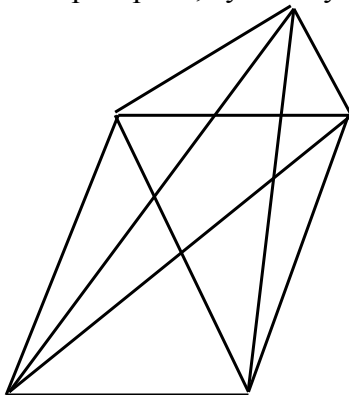
### *Перечень вопросов для подготовки к зачету*

1. Машинное представление графов. Списки инцидентности.
2. Различные стратегии систематического обхода графов.
3. Поиск в глубину на неориентированном графе.
4. Поиск в ширину на неориентированном графе.
5. Построение стягивающих деревьев неориентированных графов.
6. Фундаментальное множество циклов неориентированного графа.
7. Блоки, точки сочленения неориентированного графа.
8. Эйлеров путь, эйлеров цикл в неориентированном графе.
9. Переносимость алгоритмов поиска в глубину и ширину на ориентированные графы.
10. Классификация задач по степени сложности.
11. Алгоритм с возвратом. Генерация всех гамильтоновых циклов полного неориентированного графа.
12. Алгоритм с возвратом. Задача коммивояжера.
13. Математическая постановка задачи о поиске кратчайших путей на взвешенном ориентированном графе.
14. Алгоритм Форда-Беллмана на взвешенном орграфе.
15. Алгоритм Дейкстры на взвешенном орграфе.
16. Восстановление кратчайшего пути от источника до всех остальных вершин по вектору расстояний.
17. Алгоритм Флойда на взвешенном орграфе.
18. Восстановление кратчайшего пути между двумя вершинами по матрице расстояний.
19. Переносимость алгоритмов поиска кратчайших путей на неориентированные графы.
20. Центры и медианы. Оптимальное размещение центра на взвешенном сильносвязном орграфе с неотрицательными весами дуг и вершин.
21. Центры и медианы. Оптимальное размещение медианы на сильносвязном орграфе с неотрицательными весами дуг и вершин.

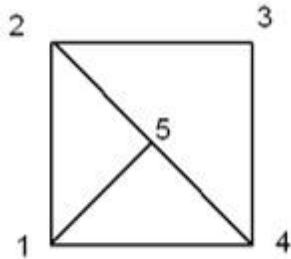


**Перечень заданий для подготовки к зачету**

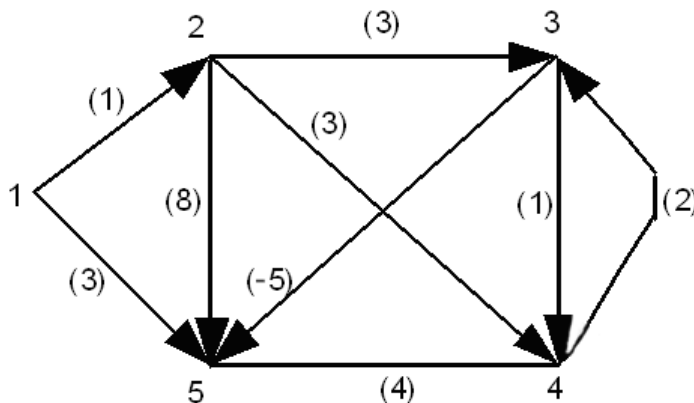
1. Изобразить списки инцидентности графа, заданного списком рёбер.
2. Изобразить списки инцидентности графа, заданного матрицей смежности.
3. Изобразить списки инцидентности графа, заданного матрицей инцидентностей.
4. Изобразить списки инцидентности модельного графа, заданного рисунком.
5. Протрассировать поиск в глубину на модельном графе.
6. Протрассировать поиск в ширину на модельном графе.
7. Разделить неориентированный граф на компоненты связности с помощью поиска в глубину.
8. Разделить неориентированный граф на компоненты связности с помощью поиска в ширину.
9. На модельном ненагруженном неориентированном графе найти кратчайший путь между парой фиксированных вершин.
10. Найти стягивающее дерево модельного неориентированного графа поиском в глубину.
11. Найти стягивающее дерево модельного неориентированного графа поиском в ширину.
12. Найти множество фундаментальных циклов модельного неориентированного графа.
13. Найти компоненты двусвязности и точки сочленения модельного неориентированного графа.
14. Проверить, существует ли эйлеров цикл в графе:



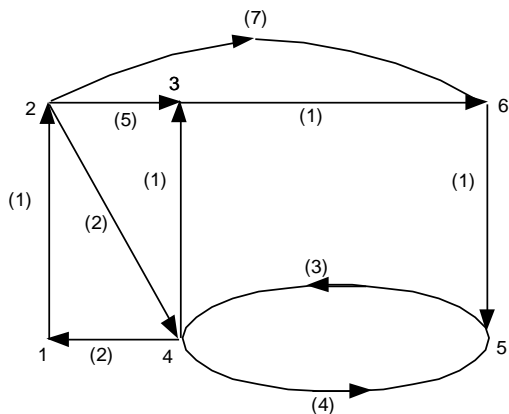
15. Найти все гамильтоновы циклы в графе:



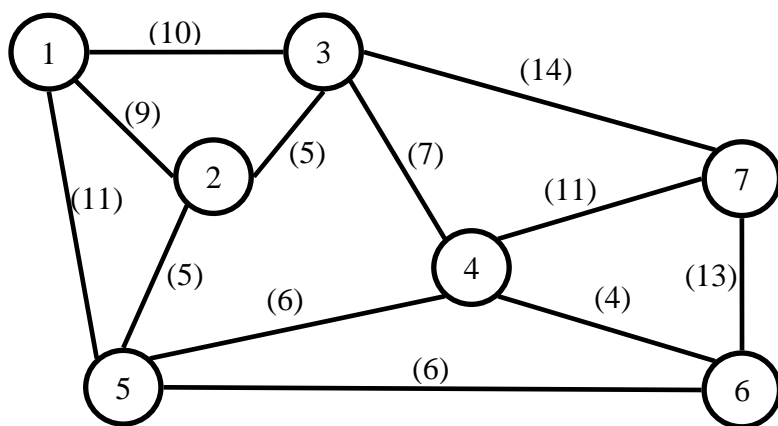
16. Протрассировать алгоритм Форда-Беллмана на модельном графе:



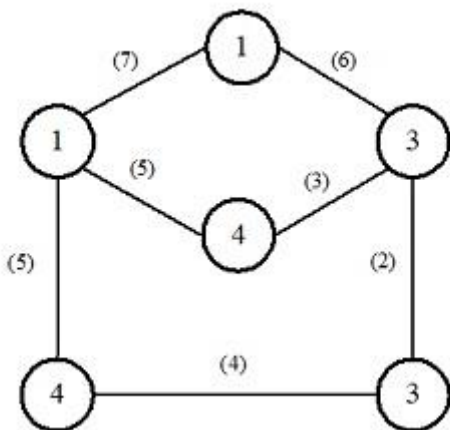
17. Протрассировать алгоритм Дейкстры на модельном графе:



18. Разместить медиану на модельном графе, считая веса вершин равными 1:



19. Разместить центр на модельном графе, считая веса вершин равными 1:



### Тесты

#### I. Графы

##### 1. Списки инцидентности для машинного представления графа это

- а) список рёбер графа в виде пар (начало ребра; конец ребра);
- б) двумерный массив  $n \times n$ , где  $n$  – количество вершин графа,  $A[i, j] = 1$ , если ребро  $i-j$  существует; 0, если не существует;
- в) двумерный массив  $n \times n$ , где  $n$  – количество вершин графа,  $A[i, j] =$  весу ребра или  $+\infty$ , если ребро не существует;
- \*г) списки инцидентных вершин.

## 2. Списки инцидентности в программе

- \*а) создаются динамически;
- б) задаются константами;
- в) хранятся в статических переменных;
- г) хранятся в стеке.

## 3. Поиск в глубину в графе

- а) просматривают каждое ребро ровно один раз;
- \*б) просматривают каждую вершину ровно один раз;
- в) просматривают только заданное множество вершин;
- г) просматривают только заданное множество рёбер.

## 4. Поиск в ширину в графе

- а) просматривают каждое ребро ровно один раз;
- \*б) просматривают каждую вершину ровно один раз;
- в) просматривают только заданное множество вершин;
- г) просматривают только заданное множество рёбер.

## 5. Поиск в глубину в графе

- а) хранит просмотренные, но ещё не использованные вершины в очереди;
- б) хранит просмотренные, но ещё не использованные вершины в деке;
- \*в) хранит просмотренные, но ещё не использованные вершины в стеке;
- г) не запоминает информацию о просмотре вершин.

## 6. Поиск в ширину в графе

- \*а) хранит просмотренные, но ещё не использованные вершины в очереди;
- б) хранит просмотренные, но ещё не использованные вершины в деке;
- в) хранит просмотренные, но ещё не использованные вершины в стеке;
- г) не запоминает информацию о просмотре вершин.

## 7. Поиск в глубину в графе имеет вычислительную сложность

- а)  $O(n*m)$ , где  $n$ - число вершин,  $m$  – количество рёбер;
- б)  $O(2^n)$ , где  $n$ - число вершин;
- в) растущую как экспонента от количества вершин графа;
- \*г)  $O(n+m)$ , где  $n$ - число вершин,  $m$  – количество рёбер.

## 8. Поиск в ширину в графе имеет вычислительную сложность

- а)  $O(n*m)$ , где  $n$ - число вершин,  $m$  – количество рёбер;
- б)  $O(2^n)$ , где  $n$ - число вершин;
- в) растущую как экспонента от количества вершин графа;
- \*г)  $O(n+m)$ , где  $n$ - число вершин,  $m$  – количество рёбер.

## 9. Поиск в глубину в графе

- а) нельзя найти путь между двумя заданными вершинами;
- \*б) можно найти произвольный путь между двумя заданными вершинами;
- в) можно найти кратчайший путь между двумя заданными вершинами;
- г) можно найти все пути между двумя заданными вершинами.

## 10. Поиск в ширину в графе

- а) нельзя найти путь между двумя заданными вершинами;
- б) можно найти произвольный путь между двумя заданными вершинами;
- \*в) можно найти кратчайший путь между двумя заданными вершинами;
- г) можно найти все пути между двумя заданными вершинами.

## 11. Разделить неор. граф на компоненты связности можно

- а) только поиском в глубину;
- б) только поиском в ширину;
- \*в) и поиском в глубину, и поиском в ширину;
- г) неориентированный граф не имеет компонент связности.

**12. Разделить орграф на компоненты связности можно**

- а) только поиском в глубину;
- б) только поиском в ширину;
- в) и поиском в глубину, и поиском в ширину;
- \*г) ориентированный граф не имеет компонент связности.

**13. Поиск в глубину и в ширину можно проводить**

- а) только для орграфов;
- б) только для неориентированных графов;
- в) только для взвешенных графов;
- \*г) для любых графов.

**14. Фундаментальное множество циклов определяется**

- \*а) только для неориентированных графов;
- б) только для ориентированных графов;
- в) только для ациклических графов;
- г) для любых графов.

**15. Фундаментальное множество циклов**

- а) максимальное множество всех пересекающихся циклов графа;
- \*б) базис линейного пространства всех циклов графа;
- в) максимальное множество всех непересекающихся циклов графа;
- г) множество всех циклов, проходящих через одну вершину.

**16. Фундаментальное множество циклов можно построить**

- \*а) добавлением всех хорд к каркасу графа;
- б) сгенерировав все циклы графа и выбрав непересекающиеся;
- в) сгенерировав все циклы графа и выбрав пересекающиеся;
- г) удалив из графа все висячие вершины.

**17. Вычислительная сложность алгоритма построения и печати всех Ф-циклов графа равна**

- \*а)  $O(n*m)$ , где  $n$  - число вершин,  $m$  - количество рёбер;
- б)  $O(2^m)$ , где  $m$  - число вершин;
- в) растёт как экспонента от количества вершин графа;
- г)  $O(n+m)$ , где  $n$  - число вершин,  $m$  - количество рёбер.

**18. Точка сочленения**

- а) имеет не менее двух сыновей при поиске в глубину;
- б) является корнем при поиске в глубину;
- \*в) содержится в любом пути между двумя вершинами для некоторой пары вершин;
- г) является точкой пересечения всех фундаментальных циклов.

**19. Компоненты двусвязности (блоки) графа**

- а) состоят из висячих вершин и инцидентных им рёбер;
- \*б) не содержат точек сочленения;
- в) состоят из вершин степени 2;
- г) состоят из точек сочленения.

**20. Вычислительная сложность алгоритма построения и печати всех блоков графа равна**

- а)  $O(n*m)$ , где  $n$ - число вершин,  $m$  – количество рёбер;
- б)  $O(2^m)$ , где  $m$ - число вершин;
- в) растёт как экспонента от количества вершин графа;
- \*г)  $O(n+m)$ , где  $n$ - число вершин,  $m$  – количество рёбер.

**II. Классификация задач по степени сложности**

**21. Вычислительная сложность алгоритма**

- а) равна числу машинных команд;
- б) равна числу команд транслятора;
- в) измеряется встроенным таймером;
- \*г) показывает рост числа шагов, необходимых для решения задачи, при неограниченном увеличении размерности входных данных.

**22. Сложность задачи**

- \*а) равна вычислительной сложности наилучшего алгоритма, известного для её решения;
- б) зависит от времени, потраченного на написание программы;
- в) зависит от времени, потраченного на разработку алгоритма;
- г) измеряется в человеко-часах.

**23. Класс P полиномиальных алгоритмов**

- а) алгоритмы для решения систем линейных уравнений;
- б) алгоритмы для решения задач линейного программирования;
- \*в) задачи, вычислительная сложность которых выражается многочленом фиксированной степени от размерности входных данных;
- г) задачи, вычислительная сложность которых выражается экспонентой от размерности входных данных.

**24. Класс E экспоненциальных алгоритмов**

- а) алгоритмы для решения комбинаторных задач;
- б) алгоритмы для решения задач статистики и теории вероятности;
- \*в) задачи, имеющие экспоненциальное число ответов;
- г) задачи, вычислительная сложность которых выражается экспонентой от размерности входных данных.

**25. NP-полные задачи**

- а) решаются точными алгоритмами за полиномиальное время;
- \*б) известными точными алгоритмами решаются за экспоненциальное время;
- в) не решаются точными алгоритмами;
- г) задачи, имеющие экспоненциальное число ответов.

**26. Алгоритм с возвратом**

- а) применяется для решения задач из класса P;
- б) применяется для решения задач из класса E;
- в) применяется для решения NP-полных задач;
- \*г) применяется для решения NP-полных задач и задач из класса E.

**27. Задача о пути коммивояжера через сеть городов**

- а) принадлежит классу E;
- б) принадлежит классу P;
- \*в) принадлежит к классу NP-полных задач;
- г) решается за линейное время от суммы вершин и рёбер графа.

**III. Кратчайшие пути**

**30. Алгоритм Форда-Беллмана применим для**

- а) произвольных неориентированных графов;
- б) ациклических графов;
- в) ориентированных графов с положительными весами;
- \*г) ориентированных графов без контуров отрицательного веса.

**31. Алгоритм Дейкстры применим для**

- а) произвольных неориентированных графов;
- б) ациклических графов;
- \*в) ориентированных графов с положительными весами;
- г) ориентированных графов без контуров отрицательного веса.

**32. Алгоритм Флойда применим для**

- а) произвольных неориентированных графов;
- б) ациклических графов;
- в) ориентированных графов с положительными весами;
- \*г) ориентированных графов без контуров отрицательного веса.

**33. Результатом работы алгоритма Форда-Беллмана является**

- а) кратчайший путь между двумя выделенными вершинами;
- б) кратчайшие пути для любой пары вершин;
- \*в) вектор расстояний от источника до остальных вершин графа;
- г) матрица расстояний для всех пар вершин.

**34. Результатом работы алгоритма Дейкстры является**

- а) кратчайший путь между двумя выделенными вершинами;
- б) кратчайшие пути для любой пары вершин;
- \*в) вектор расстояний от источника до остальных вершин графа;
- г) матрица расстояний для всех пар вершин.

**35. Результатом работы алгоритма Флойда является**

- а) кратчайший путь между двумя выделенными вершинами;
- б) кратчайшие пути для любой пары вершин;
- в) вектор расстояний от источника до остальных вершин графа;
- \*г) матрица расстояний для всех пар вершин.

**36. Вычислительная сложность алгоритма Форда – Беллмана равна**

- \*а)  $O(n*m)$ , где  $n$ - число вершин,  $m$  – количество рёбер;
- б)  $O(2^n)$ , где  $n$ - число вершин;
- в)  $O(n^2)$ , где  $n$ - число вершин;
- г)  $O(n+m)$ , где  $n$ - число вершин,  $m$  – количество рёбер.

**37. Вычислительная сложность алгоритма Дейкстры равна**

- а)  $O(n*m)$ , где  $n$ - число вершин,  $m$  – количество рёбер;
- \*б)  $O(n^2)$ , где  $n$ - число вершин;
- в)  $O(n^3)$ , где  $n$ - число вершин;
- г)  $O(n+m)$ , где  $n$ - число вершин,  $m$  – количество рёбер.

**38. Вычислительная сложность алгоритма Флойда равна**

- а)  $O(n*m)$ , где  $n$ - число вершин,  $m$  – количество рёбер;
- б)  $O(n^2)$ , где  $n$ - число вершин;
- \*в)  $O(n^3)$ , где  $n$ - число вершин;
- г)  $O(n+m)$ , где  $n$ - число вершин,  $m$  – количество рёбер.

**39.** Алгоритмом Форда-Беллмана можно найти расстояние между парой вершин неориентированного графа

- а) в произвольном неориентированном графе;
- \*б) в неориентированном графе с неотрицательными весами;
- \*в) в ациклическом неориентированном графе с произвольными весами;
- г) алгоритм Форда-Беллмана не применим для неориентированных графов.

**40.** Алгоритмом Дейкстры можно найти расстояние между парой вершин неориентированного графа

- а) в произвольном неориентированном графе;
- \*б) в неориентированном графе с неотрицательными весами;
- в) в ациклическом неориентированном графе с произвольными весами;
- г) алгоритм Дейкстры не применим для неориентированных графов.

**41.** Алгоритмом Флойда можно найти расстояние между любой парой вершин неориентированного графа

- а) в произвольном неориентированном графе;
- \*б) в неориентированном графе с неотрицательными весами;
- \*в) в ациклическом неориентированном графе;
- г) алгоритм Флойда не применим для неориентированных графов.

***Перечень заданий для курсовых работ по теме «Оптимальное размещение обслуживающих центров»***

***Задание к курсовой работе***

***1. Оптимально разместить заданный тип обслуживающего центра на графе заданного типа.***

***2. Придумать реальную задачу, соответствующую математической постановке.***

1. Размещение внешнего центра на взвешенном ориентированном графе.
2. Размещение внутреннего центра на взвешенном ориентированном графе.
3. Размещение внутреннего центра на взвешенном смешанном графе.
4. Размещение внешнего центра на взвешенном смешанном графе.
5. Размещение внешне-внутреннего центра на взвешенном ориентированном графе.
6. Размещение внешне-внутреннего центра на взвешенном смешанном графе.
7. Размещение внешней медианы на взвешенном ориентированном графе.
8. Размещение внутренней медианы на взвешенном ориентированном графе.
9. Размещение внешней медианы на взвешенном смешанном графе.
10. Размещение внутренней медианы на взвешенном смешанном графе.
11. Размещение внешне-внутренней медианы на взвешенном ориентированном графе.
12. Размещение внешне-внутренней медианы на взвешенном смешанном графе.
13. Размещение главной внешней медианы на взвешенном ориентированном графе.
14. Размещение главной внутренней медианы на взвешенном ориентированном графе.
15. Размещение главной внутренней медианы на взвешенном смешанном графе.
16. Размещение абсолютного внешнего центра на взвешенном ориентированном графе.
17. Размещение абсолютной внешней медианы на взвешенном ориентированном графе.
18. Размещение абсолютной внешне-внутренней медианы на взвешенном ориентированном графе.
19. Размещение абсолютной внутренней медианы на взвешенном ориентированном графе.
20. Размещение абсолютного внутреннего центра на взвешенном ориентированном графе.
21. Размещение абсолютного внешне-внутреннего центра на взвешенном ориентированном графе.

## 7 Оценочные средства для проведения промежуточной аттестации

### а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
<b>ПК-2 Способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования</b>		
Знать	<ul style="list-style-type: none"> <li>- основные понятия теории графов;</li> <li>- основные алгоритмы на сетях и графах;</li> <li>- классификацию задач по степени сложности</li> </ul>	<p><i>Перечень теоретических вопросов</i></p> <ol style="list-style-type: none"> <li>1. Машинное представление графов. Списки инцидентности.</li> <li>2. Различные стратегии систематического обхода графов. Поиск в глубину на неориентированном графе. Поиск в ширину на неориентированном графе.</li> <li>3. Способы построения стягивающего дерева неориентированного графа.</li> <li>4. Фундаментальное множество циклов неориентированного графа.</li> <li>5. Блоки, точки сочленения неориентированного графа.</li> <li>6. Эйлеров путь, эйлеров цикл в неориентированном графе.</li> <li>7. Переносимость алгоритмов поиска в глубину и ширину на ориентированные графы.</li> <li>8. Классификация задач по степени сложности.</li> <li>9. Сравнить алгоритмы Форда-Беллмана, Дейкстры, Флойда по следующим критериям: <ul style="list-style-type: none"> <li>- тип графа;</li> <li>- результат работы;</li> <li>- вычислительная сложность.</li> </ul> </li> </ol>
Уметь	<ul style="list-style-type: none"> <li>- делать сетевые постановки модельных задач;</li> <li>- использовать и программировать основные алгоритмы на сетях и графах;</li> <li>- решать NP-полные задачи небольшой размерности алгоритмом с возвратом.</li> </ul>	<p><i>Практические задания</i></p> <ol style="list-style-type: none"> <li>1. Разделить неориентированный граф на компоненты связности с помощью поиска в глубину и поиска в ширину.</li> <li>2. На модельном ненагруженном неориентированном графе найти кратчайший путь между парой фиксированных вершин.</li> <li>3. Задан неориентированный связный граф и вершина <math>v</math>. За полиномиальное время определить, проходит ли через <math>v</math> хотя бы один цикл.</li> <li>4. Найти компоненты двусвязности и точки сочленения модельного неориентированно-</li> </ol>



Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p><i>го графа.</i></p> <p><i>6. На модельном нагруженном оргграфе найти кратчайший путь между парой фиксированных вершин.</i></p> <p><i>7. Найти в московском метро кратчайший путь между двумя станциями. Сколькими известными вам алгоритмами это можно сделать?</i></p> <p><i>8. Раскрасить географическую карту в минимальное число цветов так, чтобы соседние страны не были закрашены одним цветом.</i></p>
Владеть	- навыками сетевых постановок для практических задач	<p><i>Задание к курсовой работе «Оптимальное размещение обслуживающих центров»</i></p> <p><i>1. Оптимально разместить заданный тип обслуживающего центра на графе заданного типа.</i></p> <p><i>2. Придумать реальную задачу, соответствующую математической постановке.</i></p>

## **б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:**

Промежуточная аттестация по дисциплине «Алгоритмы на сетях и графах» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме зачета.

Зачет по дисциплине проводится по результатам отчетности на практических занятиях с опросом в устной форме по этапам выполнения и активного выступления в беседе-обсуждении на лекционных занятиях.

### **Показатели и критерии оценивания зачета:**

– на оценку «зачтено» – обучающийся демонстрирует пороговый уровень сформированности компетенций;

– на оценку «не зачтено» – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

Курсовая работа выполняется под руководством преподавателя, в процессе ее написания обучающийся развивает навыки к научной работе, закрепляя и одновременно расширяя знания, полученные при изучении курса «Математическая логика и теория алгоритмов». При выполнении курсовой работы обучающийся должен показать свое умение работать с нормативным материалом и другими литературными источниками, а также возможность систематизировать и анализировать фактический материал и самостоятельно творчески его осмысливать.

В процессе написания курсовой работы обучающийся должен разобраться в теоретических вопросах избранной темы, самостоятельно проанализировать практический материал, разобрать и обосновать практические предложения.

Показатели и критерии оценивания курсовой работы:

– на оценку «отлично» (5 баллов) – работа выполнена в соответствии с заданием, обучающийся показывает высокий уровень знаний не только на уровне воспроизведения и объяснения информации, но и интеллектуальные навыки решения проблем и задач, нахождения уникальных ответов к проблемам, оценки и вынесения критических суждений;

– на оценку «хорошо» (4 балла) – работа выполнена в соответствии с заданием, обучающийся показывает знания не только на уровне воспроизведения и объяснения информации, но и интеллектуальные навыки решения проблем и задач, нахождения уникальных ответов к проблемам;

– на оценку «удовлетворительно» (3 балла) – работа выполнена в соответствии с заданием, обучающийся показывает знания на уровне воспроизведения и объяснения информации, интеллектуальные навыки решения простых задач;

– на оценку «неудовлетворительно» (2 балла) – задание преподавателя выполнено частично, в процессе защиты работы обучающийся допускает существенные ошибки, не может показать интеллектуальные навыки решения поставленной задачи.

– на оценку «неудовлетворительно» (1 балл) – задание преподавателя выполнено частично, обучающийся не может воспроизвести и объяснить содержание, не может показать интеллектуальные навыки решения поставленной задачи.

## **8 Учебно-методическое и информационное обеспечение дисциплины (модуля)**

### **а) Основная литература:**

1. Миков, А. Ю. Алгоритмы на сетях и графах : учебное пособие / А. Ю. Миков, С. И. Файнштейн ; МГТУ. - Магнитогорск : МГТУ, 2016. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL:

<https://magtu.informsystema.ru/uploader/fileUpload?name=2475.pdf&show=dcatalogues/1/1130219/2475.pdf&view=true> (дата обращения: 23.10.2020). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.

### б) Дополнительная литература:

1. Гданский, Н. И. Основы теории и алгоритмы на графах : учебное пособие / Н. И. Гданский. — Москва : ИНФРА-М, 2020. — 206 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-014386-6. - Текст : электронный. - URL:

<https://znanium.com/catalog/product/978686> (дата обращения: 03.11.2020). – Режим доступа: по подписке.

2. Белов, В. В. Алгоритмы и структуры данных : учебник / В. В. Белов, В. И. Чистякова. - Москва : КУРС : ИНФРА-М, 2020. - 240 с. - (Бакалавриат). - ISBN 978-5-906818-25-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1057212> (дата обращения: 03.11.2020). – Режим доступа: по подписке

### в) Программное обеспечение и Интернет-ресурсы:

*Программное обеспечение:* лицензионное программное обеспечение: операционная система; офисные программы; математические пакет, статистические пакеты, установленные на каждом персональном компьютере вычислительного центра ФГБОУ ВПО «МГТУ».

Перечень лицензионного программного обеспечения по ссылке:

<http://sps.vuz.magtu.ru/Shared%20Documents/Forms/AllItems.aspx?RootFolder=%2FShared%20Documents%2F%D0%9F%D0%BE%D0%B4%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%BA%D0%B0%20%D0%BA%20%D0%B0%D0%BA%D0%BA%D1%80%D0%B5%D0%B4%D0%B8%D1%82%D0%B0%D1%86%D0%B8%D0%B8%202020%2F%D0%A1%D0%B0%D0%BC%D0%BE%D0%BE%D0%B1%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%202019%D0%B3%2F%D0%9B%D0%B8%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%9F%D0%9E&InitialTabId=Ribbon.Document&VisibilityContext=WSSTabPersistence>

Официальные сайты промышленных предприятий и организаций: <http://www.mmk.ru>, <http://www.creditural.ru>, <http://www.magtu.ru>, <http://www.gks.ru> и т.п.; разработчиков программных продуктов: <http://www.statsoft.ru>, <http://www.microsoft.com>, <http://www.ptc.com> и т.п.; сайты лабораторий компьютерной графики <http://graphics.cs.msu.ru>, <http://cgm.graphicon.ru>.

### 9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Тип и название аудитории	Оснащение аудитории
Лекционная аудитория	Мультимедийные средства хранения, передачи и представления информации
Компьютерный класс	Персональные компьютеры с пакетом Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета
Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки	Все классы УИТ и АСУ с персональными компьютерами, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета
Аудиторий для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Ауд. 282 и классы УИТ и АСУ
Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к	Классы УИТ и АСУ

Тип и название аудитории	Оснащение аудитории
сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации	
Помещения для хранения и профилактического обслуживания учебного оборудования	Центр информационных технологий – ауд. 379