

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Директор физтех-школы  
прикладной математики и  
информатики**

**А.М. Райгородский**

**Рабочая программа дисциплины (модуля)**

<b>по дисциплине:</b>	Алгоритмы и структуры данных
<b>по направлению:</b>	Прикладная математика и информатика
<b>профиль подготовки:</b>	Прикладная математика, компьютерные науки и инженерия Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр

Семестры, формы промежуточной аттестации:

1 (осенний) - Экзамен

2 (весенний) - Экзамен

Аудиторных часов: 120 всего, в том числе:

лекции: 60 час.

семинары: 0 час.

лабораторные занятия: 60 час.

Самостоятельная работа: 180 час.

Подготовка к экзамену: 60 час.

Всего часов: 360, всего зач. ед.: 8

Количество контрольных работ, заданий: 6

Программу составил: И.Д. Степанов, ассистент

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 12.02.2024

## Аннотация

В начале вводятся общие математические обозначения, позволяющие работать с асимптотиками и оценивать сложность работы алгоритмов. Семестр посвящён изучению структур данных, необходимых для разнообразных более сложных алгоритмов. Простейшие структуры стек, очередь, вектор анализируются на предмет эффективности и времени выполнения. Вводятся кучи (двоичная, биномиальная и фибоначчьева), описываются границы их применимости. Изучаются деревья поиска (splay, AVL, декартово, В-дерево) вместе с подробными доказательствами корректности и асимптотики, а также с описанием прикладных преимуществ каждой структуры. Рассматриваются наиболее универсальные техники обработки запросов: хэш-таблицы, деревья отрезков, деревья Фенвика (в том числе многомерные), разреженные таблицы. В рамках рассматриваемых тем оттачиваются различные техники оценки временной сложности алгоритмов: метод потенциалов и метод бухгалтерского учёта. Курс в целом рассчитан на изучение базовых структур, реализация которых требуется во множестве более продвинутых алгоритмов.

Слушатели знакомятся с понятиями алгоритмов, асимптотик, графов, а также рассчитывая на умение студентов самостоятельно реализовывать программный код, обеспечивающий работу сформулированного алгоритма, курс раскрывает аспекты теории графов в области поиска максимального потока в транспортных сетях (в том числе минимальной стоимости), затем переходит к изучению базовых и продвинутых алгоритмов на строках (включая оптимальными способами хранения строк и структур над ними в памяти компьютера), и, наконец, завершается блоком о вычислительной геометрии (освещаются технические детали возможного представления геометрических примитивов в цифровом виде, а также обсуждаются многие классические алгоритмы, актуальность которых обусловлена в первую очередь практическими приложениями).

Дисциплина включает подробное освещение теоретической стороны алгоритмов, разбор и тренировка решений практических задач, а также предполагает самостоятельное изучение студентами материала предмета через решение домашних теоретических и практических задач. Для освоения курса необходимы базовые понимания о понятии алгоритма и работе компьютера; также требуется достаточная подкованность в простейших определениях и терминах дискретной математики.

Заключительным этапом всего курса является экзамент, целью которого является проверка теоретических знаний студентов, а также выявление практических навыков применения полученных знаний при выполнении практических заданий.

## 1. Цели и задачи

### Цель дисциплины

- познакомить студентов с основными принципами проектирования и анализа алгоритмов и структур данных, закрепление навыков обоснования корректности алгоритмов, их практической реализации, теоретической и экспериментальной оценки их временной сложности.

### Задачи дисциплины

- научить формулировать задачи в терминах изученных теорий, выбирать подходящий алгоритм для поставленной задачи;
- научить разрабатывать комбинации алгоритмов для решения поставленных задач, оценивать сложности алгоритмов, выбирать подходящие структуры данных для поставленных задач, реализовывать алгоритмы на языке программирования C++.

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук и использовать их в профессиональной деятельности	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.3 Способен определять границы применимости полученных результатов

ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности

### 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

- определения асимптотик в О-нотации;
- определения простейших линейных структур данных (стек, очередь, вектор) и времена обработки запросов в них;
- алгоритм быстрой сортировки;
- определение и практическую необходимость деревьев поиска;
- определения потоков в сети, базовых функций над строками, базовых геометрических объектов;
- алгоритмы для нахождения максимального потока в сети (в т.ч. минимальной стоимости);
- алгоритмы поиска шаблона в тексте;
- способы представления геометрических объектов в памяти компьютера.

уметь:

- оценивать сложность алгоритмов;
- строго доказывать утверждения о корректности алгоритмов;
- применять необходимую технику для решения алгоритмических задач.

владеть:

- разнообразными методами пересечения базовых геометрических примитивов;
- методами доказательства корректности утверждений об алгоритмах;
- приёмами сведения общих задач к более конкретным и простым.

### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

#### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Асимптотики, мастер-теорема	4		4	12
2	Линейные структуры данных	4		4	13
3	Сортировки и порядковые статистики	4		4	13
4	Кучи	4		4	13
5	Деревья поиска	4		4	13
6	Дерево отрезков, дерево Фенвика	4		4	13
7	Хэш-таблицы, фильтры Блума	6		6	13
8	Паросочетания, алгоритм Куна	4		4	12
9	Максимальные потоки в сетях	6		6	13
10	Простейшие строковые алгоритмы	4		4	13
11	Строковые суффиксные структуры	4		4	13
12	Геометрические примитивы	4		4	13
13	Выпуклая оболочка	4		4	13
14	Рандомизированные алгоритмы и комбинаторные игры	4		4	13
Итого часов		60		60	180

Подготовка к экзамену	60 час.
Общая трудоёмкость	360 час., 8 зач.ед.

## 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

### Семестр: 1 (Осенний)

#### 1. Асимптотики, мастер-теорема

Обозначения в О-нотации: о-малое и О-большое, омега-малое и Омега-большое, Тета-большое. Независимость определения О-большого и Омега-большого от начального сдвига. Мастер-теорема, пример применения для рекурренты  $T(n) = 2T(n/2) + O(n)$ .

#### 2. Линейные структуры данных

Структуры данных стек, очередь, вектор, дек. Поиск ближайшего большего справа за  $O(n)$  в массиве. Поиск минимума в стеке и очереди. Метод бухгалтерского учёта для доказательства асимптотики времени обработки запросов в векторе.

#### 3. Сортировки и порядковые статистики

Задача сортировки. Определение стабильной сортировки. Сортировка слиянием, подсчёт числа инверсий в перестановке. Стабильная сортировка подсчётом, цифровая сортировка LSD. Быстрая сортировка со случайным выбором пивота, поиска k-й порядковой статистики. Дерандомизация: детерминированный алгоритм быстрой сортировки с выбором в качестве пивота медианы массива медиан пятёрок.

#### 4. Кучи

Определение кучи и запросы, необходимые для обработки. Двоичная куча: операции siftUp и siftDown. Выражение остальных операций через данные. Асимптотика времени работы. Биномиальные деревья и биномиальная куча: скорость работы и преимущества по сравнению с двоичной кучей. Фибоначчиева куча: асимптотика с помощью метода бухгалтерского учёта.

#### 5. Деревья поиска

Определение дерева поиска, обрабатываемые запросы. Теоретическая реализация и анализ времени работы деревьев: splay-деревья, AVL-деревья, декартового дерева, B-деревья как частного случая (a, b)-деревья. Практические применения и преимущества каждого типа деревьев.

#### 6. Дерево отрезков, дерево Фенвика

Обрабатываемые запросы в дереве отрезков. Отложенные операции. Дерево отрезков снизу. Двумерное дерево отрезков. Динамическое и персистентное дерево отрезков. Дерево Фенвика: булевы операции над битами. Многомерное дерево отрезков, запросы к подотрезкам и подпрямоугольникам.

#### 7. Хэш-таблицы, фильтры Блума

Задача хэширования. Определения совершенного и универсального семейства хэш-функций. Вероятность коллизии. Хэш-таблицы с открытой адресацией, хэш-таблицы методом цепочек. Двойное хэширование. Фильтры Блума: применения и реализация.

### Семестр: 2 (Весенний)

## 8. Паросочетания, алгоритм Куна

Определение паросочетания, двудольного графа. Максимальное и наибольшее паросочетание. Теорема Берга, увеличивающие и чередующиеся цепи. Алгоритм Куна, корректность и асимптотика. Реализация. Теорема Дилворта и Мирского. Покрытие частично упорядоченного множества путями. Минимальное вершинное покрытие и максимальное независимое множество.

## 9. Максимальные потоки в сетях

Определение транспортной сети, потока, разреза. Теорема Форда-Фалкерсона. Алгоритм Форда-Фалкерсона, Эдмондса-Карпа. Блокирующий поток, слоистая сеть. Алгоритм Диница. Взвешенная задача, поток минимальной стоимости. Алгоритм поиска максимального потока минимальной стоимости (с использованием алгоритмов Форда-Беллмана и Дейкстры с потенциалами).

## 10. Простейшие строковые алгоритмы

Зет- и префикс-функция строки. Алгоритмы их нахождения за линейное время. Алгоритм Манакера. Структура данных бор. Алгоритм Ахо-Корасик. Поиск вхождения шаблона в строку и шаблонов в текст. Подсчёт числа вхождений.

## 11. Строковые суффиксные структуры

Суффиксное дерево и суффиксный автомат. Нахождение числа всех подстрок строки.

## 12. Геометрические примитивы

Точка, прямая, окружность, отрезок, луч. Представление в памяти компьютера. Скалярное и векторное произведение, две формы выражения. Поиск пересечения двух прямых, прямой и окружности, двух окружностей. Нахождение замечательных точек треугольника: точки пересечения биссектрис, медиан, высот.

## 13. Выпуклая оболочка

Определение выпуклой оболочки. Алгоритмы Джарвиса, Грэхема, Эндрю. Алгоритм Чена. Поиск объемлющей фигуры минимального периметра или площади.

## 14. Рандомизированные алгоритмы и комбинаторные игры

Минимальная покрывающая окружность. Непустота пересечения полуплоскостей. Определение игры ним, эквивалентности игр. Теория Шпрага-Гранди об эквивалентности любой справедливой ациклической игры игре ним. Ретроанализ.

## 5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система), а так же персональными компьютерами для обучающихся.

## 6. Перечень рекомендуемой литературы

### Основная литература

1. Алгоритмы: построение и анализ [Текст] : [учебник для вузов] / Т. Кормен [и др.] ; [пер. с англ. И. В. Красикова и др.] .— 3-е изд. — М. : Вильямс, 2014 .— 1328 с.
2. Программирование: теоремы и задачи [Текст], [учеб. пособие] /А. Шень. -М., МЦНМО, 2017

**7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

1. [http://neerc.ifmo.ru/wiki/index.php?title=Дискретная\\_математика,\\_алгоритмы\\_и\\_структуры\\_данных](http://neerc.ifmo.ru/wiki/index.php?title=Дискретная_математика,_алгоритмы_и_структуры_данных). «Викиконспекты», сайт Санкт-Петербургского Университета ИМТО.
2. <http://e-maxx.ru/>. Maximal algo: личный сайт Максима Иванова, посвящённый алгоритмам и структурам данных.
3. <https://codeforces.com/>. Международная платформа онлайн-конTESTов по спортивному программированию на базе СГУ и ИМТО.

Литература для самостоятельного изучения

Введение в теорию алгоритмов и структур данных [Текст] : [учеб. пособие для вузов] / М. А. Бабенко, М. В. Левин .— М. : МЦНМО ; ФМОП, 2012 .— 144 с.

**8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций. Для успешного освоения дисциплины, студент использует следующие программные средства:

- компилятор языка C++.

**9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Успешное освоение дисциплины требует:

- посещения студентом всех видов аудиторных занятий;
- ведения конспекта в ходе лекционных занятий;
- качественной самостоятельной подготовки к практическим занятиям, активной работы на них;
- активной самостоятельной и аудиторной работы студента;
- своевременной сдачи преподавателю заданий по аудиторным видам работ.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Прикладная математика и информатика
<b>профиль подготовки:</b>	Прикладная математика, компьютерные науки и инженерия Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	<u>1</u>
<b>квалификация:</b>	бакалавр

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Экзамен
- 2 (весенний) - Экзамен

**Разработчик:** И.Д. Степанов, ассистент

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук и использовать их в профессиональной деятельности	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.3 Знает основные требования информационной безопасности

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Алгоритмы и структуры данных» обучающийся должен:

### знать:

- определения асимптотик в O-нотации;
- определения простейших линейных структур данных (стек, очередь, вектор) и времена обработки запросов в них;
- алгоритм быстрой сортировки;
- определение и практическую необходимость деревьев поиска;
- определения потоков в сети, базовых функций над строками, базовых геометрических объектов;
- алгоритмы для нахождения максимального потока в сети (в т.ч. минимальной стоимости);
- алгоритмы поиска шаблона в тексте;
- способы представления геометрических объектов в памяти компьютера.

### уметь:

- оценивать сложность алгоритмов;
- строго доказывать утверждения о корректности алгоритмов;
- применять необходимую технику для решения алгоритмических задач.

### владеть:

- разнообразными методами пересечения базовых геометрических примитивов;
- методами доказательства корректности утверждений об алгоритмах;
- приёмами сведения общих задач к более конкретным и простым.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

Для 1 семестра:

1. Что такое алгоритм?
2. Какие основные характеристики алгоритма?
3. Что такое сложность алгоритма? Какие виды сложности существуют?
4. Что такое структура данных?
5. Какие основные типы структур данных вы знаете?
6. В чем разница между стеком и очередью?
7. Что такое связанный список? Каковы его основные преимущества и недостатки по сравнению с массивом?
8. Что такое дерево в контексте структур данных?
9. Что такое хеш-таблица и каковы ее преимущества?
10. Что такое сортировка массива? Назовите несколько известных алгоритмов сортировки.
11. Пусть  $c > 0$  - константа. Докажите, что решение рекуррентности  $T(n) = T(n/3) + T(2n/3) + cn$  ведёт себя как  $\Omega(n \log n)$ .



12. Разработайте стек, который умеет прибавлять ко всем хранящимся значениям произвольную поправку  $x$  за  $O(1)$ . Иными словами, нужно реализовать операцию увеличения всех чисел в стеке на  $x$ .
13. В  $2n - 1$  ящиках лежат яблоки и апельсины. Требуется выбрать  $n$  ящиков так, что в них окажется не менее половины всех яблок и не менее половины всех апельсинов. Докажите, что такой выбор всегда существует.
14. Двоичная куча с минимумом в корне на  $n$  элементах расположена в памяти в виде массива. По числу  $k$  найдите минимальные  $k$  элементов в куче за  $O(k \log k)$ .

Для 2 семестра:

1. Что такое поиск в массиве? Какие методы поиска существуют?
2. Что такое рекурсия? Приведите пример использования рекурсии.
3. Что такое граф в контексте структур данных? Какие алгоритмы работы с графами вы знаете?
4. Что такое динамическое программирование? В каких задачах оно может быть использовано?
5. Что такое алгоритмы жадного выбора (greedy algorithms)?
6. Какой алгоритм используется для нахождения кратчайшего пути в графе?
7. Что такое алгоритмы поиска в ширину и поиска в глубину?
8. Какова основная идея алгоритма быстрой сортировки (QuickSort)?
9. Что такое бинарное дерево и каковы его основные свойства?
10. Какие алгоритмы можно использовать для поиска оптимального пути в графе?
11. Пусть  $G$  - ациклический ориентированный граф. Предложите алгоритм поиска наименьшего количества вершинно непересекающихся путей, которые покрывают все вершины  $G$ . Асимптотика:  $O(nm)$ , где  $n$  -- число вершин в  $G$ , а  $m$  -- число его рёбер.
12. Модифицируйте алгоритм Куна так, чтобы его время работы составляло  $O(am)$ , где  $a$  -- размер максимального паросочетания, а  $m$  -- число рёбер графа.
13. Докажите теорему о декомпозиции потока: если в сети  $G$  течёт некий поток  $f$ , то его можно разбить на несколько путей из  $s$  в  $t$ , а также на несколько (замкнутых) циклов.
14. Как реализовать алгоритм Ахо-Корасик на динамически расширяющемся множестве строк? Используйте идею разложения  $n$  по степеням двойки.
15. Предложите алгоритм нахождения пересечения двух выпуклых многоугольников за  $O(nm)$ , если  $n$  -- это количество сторон в первом из них, а  $m$  -- во втором.

#### 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Для 1 семестра:

1. По данному числу  $n$  найдите все пары целых положительных чисел  $(a, b)$ , такие что  $a \leq b \leq n$ , и  $a \mid b$ . Оцените асимптотическое поведение числа таких пар.
2. Число 0 записано в  $n$ -разрядной двоичной системе. К нему  $2^{\{n\}} - 1$  раз прибавляется единица. Будем считать, что время, необходимое на прибавление единицы, равно количеству единиц в двоичной записи числа, которые становятся нулями. Оцените среднюю сложность всех таких операций. Какие операции являются самыми дешёвыми, а какие -- самыми дорогими?
3. Предложите метод хранения минимального значения в очереди с помощью структуры deque. Если к очереди поступило  $n$  запросов, время работы программы должно составлять  $O(n)$ .
4. Напомним, что процедура  $\text{Partition}(A, x)$  переупорядочивает элементы массива  $A$  так, что сначала идут все элементы, не превосходящие  $x$ , в некотором порядке, а затем -- все элементы, большие  $x$ . Покажите, как реализовать  $\text{Partition}(A, x)$  с привлечением  $O(1)$  дополнительной памяти.
5. Опишите принцип работы алгоритма поиска кратчайшего пути в графе (алгоритм Дейкстры).
6. Дан неориентированный граф, представленный матрицей смежности. Напишите функцию на Python, которая определяет, есть ли в графе цикл.
7. Задана строка. Напишите функцию на Python, которая находит самую длинную подстроку-палиндром в данной строке.
8. Опишите принцип работы алгоритмов сортировки "пузырьком" и "вставками". Какая временная сложность этих алгоритмов?
9. Что такое стек? Опишите принцип работы стека и основные операции над ним (push, pop).

10. Что такое бинарное дерево? Опишите основные типы бинарных деревьев (полное, сбалансированное).
11. Опишите принцип работы алгоритма поиска кратчайшего пути в графе (алгоритм Дейкстры).
12. Приведите пример задачи, которую можно решить с помощью динамического программирования.

Примеры экзаменационных билетов:

1. а) Нижняя оценка на число сравнений в сортировке, основанной на сравнениях.  
б) Лемма о корректности siftDown и siftUp.
2. а) Удаление в куче (по указателю и по значению).  
б) Двумерное дерево отрезков снизу: прибавление в точке и сумма в прямоугольнике.
3. а) Фибоначчиева куча: операций decreaseKey. Асимптотика и корректность.  
б) Фильтры Блума: асимптотика и корректность.

Для 2 семестра:

1. Сведите задачу поиска максимального паросочетания в двудольном графе к поиску максимального потока в некоторой сети. Сравните время работы алгоритмов Форда--Фалкерсона и Эдмондса--Карпа. Можно ли проделать то же в произвольном (необязательно двудольном) графе?
2. На фабрику поступило  $n$  заказов, реализация  $i$ -го из которых принесёт ей прибыль в  $a_{\{i\}}$  рублей. Каждый заказ для производства требует некоторого набора инструментов. Пусть всего зависимостей “заказ--инструмент” ровно  $k$ . Инструменты можно переиспользовать, то есть задействовать в нескольких заказах. Однако в данный момент на фабрике вообще нет инструментов, так что покупка  $j$ -го из них обойдётся в  $b_{\{j\}}$  рублей. Конечно, от приёма каких-то заказов или покупки каких-то инструментов фабрика может отказаться. Какие заказы следует реализовать для максимизации прибыли (с учётом затрат на инструменты)?
3. Задан полный взвешенный двудольный граф, в обеих долях которого находится по  $n$  вершин. Предложите алгоритм поиска самого дешёвого совершенного паросочетания.
4. С помощью суффиксного автомата найдите количество различных подстрок строки  $s$  за время, линейное от её длины.
5. Пусть из  $n$  объектов только один обладает хорошим свойством. Смоделируем выбор с возвращением: на каждом шаге выбирается случайный равновероятный объект. Если он не обладает хорошим свойством, предмет кладётся обратно в кучу. В противном случае игра останавливается. Определите математическое ожидание числа шагов до конца игры.
6. Напишите функцию на Python, которая реализует алгоритм бинарного поиска в отсортированном массиве.
7. Напишите функцию на Python, которая принимает на вход два связанных списка и возвращает новый связный список, содержащий элементы обоих списков, объединённые в порядке возрастания.
8. Опишите принцип работы алгоритма обхода бинарного дерева в ширину (BFS).
9. Опишите принцип работы алгоритма обхода бинарного дерева в глубину (DFS).
10. Что такое граф? Опишите основные типы графов (ориентированный, неориентированный, взвешенный).
11. Опишите принцип работы алгоритма поиска подстроки в строке (алгоритм Рабина-Карпа).
12. Опишите принцип работы алгоритма вычисления расстояния Левенштейна.

Примеры экзаменационных билетов:

1. а) Поиск минимального вершинного покрытия с доказательством корректности.  
б) Теоремы Карзанова. Леммы для первой теоремы (б/д), вывод из них самой теоремы. Вторая теорема -- б/д. Единичные сети и алгоритм Хопкрофта-Карпа.
2. а) Поиск окружности минимального радиуса, покрывающей данное множество точек.  
б) Зет-функция: определение, линейный алгоритм построения и доказательство корректности. Поиск всех вхождений шаблона  $s$  в текст  $t$ .
3. а) Алгоритм Ахо--Корасик: поиск всех вхождений шаблона  $s$  (в котором допускается не более  $k$  вопросов) в текст  $t$ .

б) Поток минимальной стоимости. Корректность алгоритма, асимптотика. Применение алгоритмов Форда-Беллмана и Дейкстры (с потенциалами).

#### Критерии оценивания

Оценка "Отлично" (10) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы, код оформлен в едином удобочитаемом стиле.

Оценка "Отлично" (9) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы.

Оценка "Отлично" (8) - полностью и вовремя решены все задачи без ошибок. Продemonстрирован грамотный подход к решению задач.

Оценка "Хорошо" (7) - полностью решены все задачи. Допущены несущественные ошибки.

Оценка "Хорошо" (6) - полностью решено большинство задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Хорошо" (5) - полностью решено две трети задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Удовлетворительно" (4) - полностью решено более половины задач. В остальных задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Оценка "Удовлетворительно" (3) - полностью решено более половины задач.

Оценка "Неудовлетворительно" (2) - решено менее половины задач.

Оценка "Неудовлетворительно" (1) - не решено ни одной задачи.

#### **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Экзамен может проводиться по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины, и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме.

На экзамене обучающийся решает задачи и защищает их у семинариста. Процесс защиты заключается в демонстрации кода решения задачи и объяснения его работы.

Неправильно решенная задача или задача, имеющая ошибки, отправляется на доработку с возможностью повторной защиты. Количество повторных защит регламентируется преподавателем (семинаристом).

Защита может выполняться удаленно с использованием электронной почты, внешним репозиторием системы контроля версий и др.