

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

**УТВЕРЖДЕНО**

**Директор физтех-школы  
радиотехники и компьютерных  
технологий**

**Д.А. Гаврилов**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Разработка и анализ алгоритмов
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра радиоэлектроники и прикладной информатики
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестры, формы промежуточной аттестации:

3 (осенний) - Дифференцированный зачет

4 (весенний) - Экзамен

Аудиторных часов: 135 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 105 час.

Самостоятельная работа: 195 час.

Подготовка к экзамену: 30 час.

Всего часов: 360, всего зач. ед.: 8

Количество контрольных работ, заданий: 4

Программу составил: И.Р. Дединский, старший преподаватель

Программа обсуждена на заседании кафедры радиоэлектроники и прикладной информатики 30.01.2024

## Аннотация

Курс посвящён разработке и анализу алгоритмов. В курсе изучаются хеш-функции и хеш-таблицы, основы динамического программирования и графы. Рассматриваются варианты построения персистентных хеш-таблиц, в том числе, во внешней памяти. В динамическом программировании вводится понятие уравнения Беллмана, рассматриваются этапы решения задач динамического программирования и их варианты: многомерные, по битовым маскам, по контуру. Значительная часть курса посвящена алгоритмам работы с графами: вводится понятие графа, варианты его организации. Изучаются алгоритмы поиска на графах, определения его компонент связности, планарности, нахождения циклов и многих других.

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.

## 1. Цели и задачи

### Цель дисциплины

Целями дисциплины являются ознакомление студентов с алгоритмами и структурами данных, применяющихся для решения задач на темы хеш-функций и хеш-таблиц, динамического программирования и графов. Закрепляются навыки построения алгоритмов из уже известных составных частей, навыки обоснования корректности алгоритмов, их практической реализации, теоретической и экспериментальной оценки их временной сложности.

### Задачи дисциплины

- изучить такие разделы, как хеш-таблицы, динамическое программирование и графы;
- научить разрабатывать композиции алгоритмов для решения поставленных задач;
- оценивать сложности алгоритмов;
- выбирать подходящие структуры данных для поставленных задач;
- реализовывать алгоритмы на языках программирования C и C++.

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели

новые научные результаты	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

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

В результате освоения дисциплины обучающиеся должны

знать:

- определения асимптотик в O-нотации;
- определения простейших линейных структур данных (стек, очередь, дек, вектор) и сложность обработки запросов в них;
- алгоритм быстрой сортировки;
- алгоритм сортировки слиянием;
- определение и практическую необходимость деревьев поиска;
- классификацию и сферы применения хеш-функций;
- организацию и использование хеш-таблиц;
- условия возникновения задачи динамического программирования;
- сведение задачи динамического программирования к функции Беллмана;
- виды задач динамического программирования и методы их решения;
- понятие графов и организацию их хранения;
- алгоритмы обхода графов;
- алгоритмы нахождения специальных элементов в графах;
- алгоритмы нахождения кратчайших путей в графах;
- алгоритмы нахождения потоков в графах;
- алгоритмы нахождения минимальных разрезов и максимальных потоков в графах.

уметь:

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

владеть:

- разнообразными деревьями поиска и методикой выбора наиболее предпочтительного в каждой конкретной ситуации;
- техникой выбора необходимой хеш-функции для данной задачи;
- техникой сведения уравнения Беллмана к программному коду;
- техникой реализаций изученных алгоритмов на графах;
- методами доказательства корректности утверждений об алгоритмах;
- методами декомпозиции задач на более простые;
- методами синтеза решения сложных задач из простых.

### 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.

		лекции	семинары	лаборат. работы	работа
1	Асимптотики. Мастер-теорема			3	6
2	Линейные алгоритмы			3	6
3	Переборные алгоритмы			3	6
4	Сортировки и порядковые статистики			6	12
5	Задача поиска			3	6
6	Деревья и кучи			6	12
7	Деревья поиска			9	18
8	Дерево отрезков и дерево Фенвика			6	12
9	Хэш-таблицы, фильтры Блума			6	12
10	Задача динамического программирования.	4		4	8
11	Декомпозиция	2		4	8
12	Многомерные варианты	2		4	8
13	Графы	2		4	8
14	Связанность	2		4	8
15	Мосты и точки сочленения	2		4	8
16	Алгоритм Дейкстры	2		4	8
17	Алгоритм Флойда-Уоршалла	2		4	8
18	Алгоритм Прима	2		4	8
19	Двудольные графы	2		4	8
20	Потоки в графах. Обратные рёбра.	2		4	7
21	Слоистая сеть.	2		4	6
22	Деревья.	2		4	6
23	Задача RMQ.	2		8	6
Итого часов		30		105	195
Подготовка к экзамену		30 час.			
Общая трудоёмкость		360 час., 8 зач.ед.			

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

Семестр: 3 (Осенний)

##### 1. Асимптотики. Мастер-теорема

Обозначения в O-нотации: o-малое и O-большое, омега-малое и Омега-большое, Тета-большое. Независимость определения O-большого и Омега-большого от начального сдвига. Мастер-теорема, пример применения для рекурренты  $T(n) = 3T(n/2) + O(n)$ .

##### 2. Линейные алгоритмы

Введение в жадные алгоритмы. Критерии применимости жадных алгоритмов. Доказательство корректности жадных алгоритмов. Задачи об аудиториях и о резервных копиях. Понятие об автоматах. Детерминированный конечный автомат.

##### 3. Переборные алгоритмы

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.

##### 4. Сортировки и порядковые статистики

Задача сортировки. Определение устойчивой сортировки. Сортировки вставками, Шелла, comb. Сортировка слиянием, подсчёт числа инверсий в перестановке. Сортировка подсчётом, устойчивая сортировка подсчётом, радикал-сортировка. Быстрая сортировка. Варианты Ломута и Хоара. Поиск k-й порядковой статистики. Дерандомизация: детерминированный алгоритм быстрой сортировки с выбором в качестве пивота медианы массива медиан пятерок.

## 5. Задача поиска

Обобщённая задача поиска. Линейный поиск. Поиск с сужением зоны. Двоичный и троичный поиск. Подготовка данных для поиска. CRUD-структуры данных. Списки. Списки с пропусками.

## 6. Деревья и кучи

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

## 7. Деревья поиска

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

## 8. Дерево отрезков и 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.

## 9. 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. Задача репликации. Дерево репликации. Roll-hash. Фильтры Блума: применения и реализация. Дедупликация.

## Семестр: 4 (Весенний)

## 10. Задача динамического программирования.

Уравнение Беллмана. Решение задачи динамического программирования в прямом и обратном порядке. Восстановление ответа.

## 11. Декомпозиция

Выбор декомпозиции. Задача Левенштейна. Задача о рюкзаке. Битовые множества и побитовые операции.

## 12. Многомерные варианты

Использование дерева отрезков и дерева Фенвика. Динамическое программирование по контуру. Быстрое возведение матрицы в степень.

## 13. Графы

Графы, их организация. Обход графов. Алгоритмы BFS и DFS. Лемма о белых путях. Топологическая сортировка.

#### 14. Связанность

Отношение сильной связности. Компоненты связности. Алгоритмы Косарайю и Тарджана. Конденсация графа.

#### 15. Мосты и точки сочленения

Классификация рёбер в дереве обхода DFS. Отношение эквивалентности R. Рёберная двусвязность. Мосты и точки сочленения. 2SAT.

#### 16. Алгоритм Дейкстры

Поиск кратчайшего расстояния в графах. Алгоритм Дейкстры. Реализации алгоритма Дейкстры. Алгоритм A\*. Допустимые и монотонные эвристики. Корректность и сложность.

#### 17. Алгоритм Флойда-Уоршалла

Реализация, асимптотика. Нахождение отрицательных циклов. Алгоритм Джонсона. Алгоритм Форда-Беллмана. Матрица транзитивного замыкания.

#### 18. Алгоритм Прима

Поиск минимальных основных деревьев. Алгоритм Прима. Лемма о безопасном ребре. Система непересекающихся множеств. Алгоритм Краскала. Алгоритм Борувки.

#### 19. Двудольные графы

Паросочетания в произвольном графе. Двудольные графы. Понятие увеличивающего пути. Теорема Берга. Алгоритм поиска максимального паросочетания в двудольном графе.

#### 20. Потоки в графах. Обратные рёбра.

Потоки в графах. Определение сети, потока, остаточной сети. Обратные рёбра. Лемма о связи величины произвольного потока и величины произвольного разреза. Теорема Форда-Фалкерсона. Алгоритм Форда-Фалкерсона. Алгоритм Эдмондса-Карпа. Масштабирование в алгоритме Эдмондса-Карпа.

#### 21. Слоистая сеть.

Слоистая сеть. Блокирующий поток. Алгоритм Диница. Единичные сети. Эффективность алгоритма Диница в единичных сетях.

#### 22. Деревья.

Деревья. Диаметр дерева. Определение центроида в дереве. Лемма о количестве центроидов. Изоморфизм графов. Задача LCA. Решение с помощью Эйлерова обхода. Решение LCA с помощью алгоритма Фарах-Колтона и Бендера.

#### 23. Задача RMQ.

Задача RMQ. Решение за  $O(N \log N)$  предподсёта. Решение за  $O(N)$  предподсчёта. Heavy-light декомпозиция. Тяжёлые и лёгкие рёбра. Центроидная декомпозиция.

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

Учебная аудитория, оснащенная компьютером и мультимедийным оборудованием (проектор, звуковая система).

## **6.Перечень рекомендуемой литературы**

### Основная литература

Фонд литературы кафедры:

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
3. Алгоритмы [Текст] : [учеб. пособие для вузов] / С. Дасгупта, Х. Пападимитриу, У. Вазирани ; пер. с англ. А. А. Куликова ; под ред. А. Шеня .— М. : МЦНМО, 2014 .— 320 с.

### Дополнительная литература

## **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-контестов по спортивному программированию на базе СГУ и ИМТО.

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

Для успешного освоения дисциплины, студент использует следующие программные средства: компилятор языка С.

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

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя проработку учебного материала (по конспектам лекций, учебной литературе), подготовку к практическим занятиям.

Промежуточный контроль знаний проводится в виде сдачи проектных заданий.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Компьютерные технологии и вычислительная техника Физтех-школа Радиотехники и Компьютерных Технологий кафедра радиоэлектроники и прикладной информатики
<b>курс:</b>	2
<b>квалификация:</b>	бакалавр

Семестры, формы промежуточной аттестации:

3 (осенний) - Дифференцированный зачет  
4 (весенний) - Экзамен

**Разработчик:** И.Р. Дединский, старший преподаватель



## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
ОПК-3 Способен составлять и оформлять научные и (или) технические (технологические, инновационные) отчеты (публикации, проекты)	ОПК-3.3 Владеет методами визуального и графического представления результатов научной (научно-технической, инновационной технологической) деятельности в виде отчетов, научных публикаций
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности
	ПК-1.2 Способен выдвигать гипотезы, строить математические модели для описания изучаемых явлений и процессов, оценивать качество разработанной модели
	ПК-1.3 Способен применять теоретические и (или) экспериментальные методы исследований к конкретной научной задаче и интерпретировать полученные результаты
ПК-2 Способен самостоятельно или в качестве члена (руководителя) малого коллектива организовывать и проводить научные исследования и их апробацию	ПК-2.1 Знает принципы построения научной работы, методы сбора и анализа полученного материала, способы аргументации
	ПК-2.2 Способен планировать и проводить научные исследования самостоятельно или в качестве члена (руководителя) малого научного коллектива
	ПК-2.3 Способен проводить апробацию результатов научно-исследовательской работы посредством публикации научных статей и участия в конференциях

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Разработка и анализ алгоритмов» обучающийся должен:

**знать:**

- определения асимптотик в O-нотации;
- определения простейших линейных структур данных (стек, очередь, дек, вектор) и сложность обработки запросов в них;
- алгоритм быстрой сортировки;
- алгоритм сортировки слиянием;
- определение и практическую необходимость деревьев поиска;
- классификацию и сферы применения хеш-функций;
- организацию и использование хеш-таблиц;
- условия возникновения задачи динамического программирования;
- сведение задачи динамического программирования к функции Беллмана;
- виды задач динамического программирования и методы их решения;
- понятие графов и организацию их хранения;
- алгоритмы обхода графов;
- алгоритмы нахождения специальных элементов в графах;
- алгоритмы нахождения кратчайших путей в графах;
- алгоритмы нахождения потоков в графах;
- алгоритмы нахождения минимальных разрезов и максимальных потоков в графах.

**уметь:**

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

**владеть:**

- разнообразными деревьями поиска и методикой выбора наиболее предпочтительного в каждой конкретной ситуации;
- техникой выбора необходимой хеш-функции для данной задачи;
- техникой сведения уравнения Беллмана к программному коду;
- техникой реализаций изученных алгоритмов на графах;
- методами доказательства корректности утверждений об алгоритмах;
- методами декомпозиции задач на более простые;
- методами синтеза решения сложных задач из простых.

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

Примеры заданий в осеннем семестре:

1. Опишите алгоритм нахождения числа в упорядоченном массиве за время  $O(\log m)$ , где  $m$  – позиция числа в массиве.
2. Предложите максимально эффективный вариант алгоритма radix-сортировки 32-битных чисел. Обоснуйте своё решение.
3. Предложите реализацию очереди с применением необходимого количества стеков. Определите амортизированную сложность операций enqueue и dequeue.

Примеры заданий в весеннем семестре:

1. Определите, есть ли в неориентированном графе гамильтонов цикл за время  $O(2^n \cdot n)$ .
2. Найдите минимальное вершинное покрытие графа.
3. Приведите пример графа, в котором значения, находимые алгоритмом Флойда, экспоненциально растут.

За каждое задание выставляется оценка в соответствии со следующими критериями оценивания:

Отлично

10) Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы, код оформлен в едином удобочитаемом стиле.

9) Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач, реализованы оптимальные алгоритмы.

8) Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач.

Хорошо

7) Полностью решены все задачи. Допущены несущественные ошибки

6) Полностью решено большинство задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

5) Полностью решено две трети задач. В некоторых задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

Удовлетворительно.

4) Полностью решено более половины задач. В остальных задачах допущены и не исправлены ошибки, либо некоторые задачи решены частично.

3) Полностью решено более половины задач.

Неудовлетворительно

2) Решено менее половины задач.

1) Не решено ни одной задачи.

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

Перечень контрольных вопросов для сдачи дифференцированного зачета:

1. Асимптотики. Операции над асимптотиками. Мастер-теорема.
2. Детерминированный конечный автомат. Определение. Примеры построения.
3. Структура данных стек. Варианты реализации. Асимптотики. Персистентный стек.
4. Структура данных очередь. Варианты реализации. Реализация очереди на стеках.
5. Задача сортировки. Квадратичные сортировки. Инверсии. Сортировки Шелла и comb.
6. Быстрая сортировка. Выбор ведущего элемента. Варианты разбиения на партии Ломута и Хоара. Оценка сложности при рандомизированном выборе ведущего элемента.
7. Алгоритм нахождения k-й порядковой статистики. Оценка сложности при рандомизированном выборе ведущего элемента.
8. Задача поиска. Линейный поиск и поиск с сужением зоны. Поиск с использованием свойств элемента.
9. Связные списки. CRUD. Операции CRUD в списках. Алгоритмы операций CRUD в структуре данных списки с пропусками.
10. Бинарная куча. Определение и свойства. Операции SiftUp и SiftDown. Асимптотика.
11. Heapsort. Реализация через модифицированные операции SiftUp и SiftDown. Асимптотика. Применимость.
12. Биномиальные деревья. Биномиальная куча. Алгоритмы операций CRUD над биномиальной кучей. Сравнение с бинарной кучей.
13. Деревья поиска. Простое дерево поиска. Асимптотика времени работы рандомизированного времени поиска. Операция вставки в корень. Повороты. Рандомизация деревьев поиска. Удаление узла.
14. Splay-дерево. Операции над splay-деревом. Асимптотика времени работы CRUD-операций. Преимущества и недостатки по сравнению с обычным деревом поиска.
15. Сбалансированные деревья поиска. Варианты балансировки. Оценка высоты дерева в зависимости от организации балансировки.

16. AVL-деревья. Повороты – малый и большой. Балансировка AVL-деревя. Асимптотическая сложность CRUD-операций.
17. Декартовы деревья. Реализация операций split и merge. Алгоритмы CRUD над Декартовыми деревьями.
18. В- и В+-деревья. Организация. Применение. Алгоритмы CRUD над В-деревьями и их сложность.
19. Дерево отрезков. Моноиды. Организация. Прямые операции и их реализация. Отложенные операции. Динамическое дерево отрезков. Персистентное дерево отрезков.
20. Дерево Фенвика. Побитовые операции и их применение для дерева Фенвика. Операции над деревом Фенвика.
21. Обобщённый быстрый поиск. Партиционирование пространства поиска. Хеш-функции. Задача хеширования. Требуемые свойства хеш-функций. Универсальное хеширование.
22. Хеш-таблицы. Организация хеш-таблиц с открытой и закрытой адресацией. Рехеширование. Амортизированная сложность CRUD операций.
23. Задача репликации данных. Дерево репликации. Roll-хеширование.
24. Фильтр Блума. Обоснование эффективности и применение. Реализация. Дедупликация.

Перечень контрольных вопросов для сдачи экзамена:

1. Классификация хеш-функций. Cookoo-hash таблицы.
2. Персистентные хеш-таблицы. Хеш-таблицы во внешней памяти.
3. Задача динамического программирования. Уравнение Беллмана. Решение задачи динамического программирования в прямом и обратном порядке. Восстановление ответа.
4. Декомпозиция задачи в динамическом программировании. Выбор декомпозиции. Критерии выбора. Примеры удачного и неудачного выбора.
5. Задача о расстоянии редактирования. Декомпозиция задачи. Решение. Восстановление ответа.
6. Задача о наполнении целочисленного рюкзака. Декомпозиция задачи. Восстановление ответа. Оценка эвристик.
7. Задача о счастливых билетах. Декомпозиция задачи. Решение.
8. Задача о замощении прямоугольника домино. Декомпозиция задачи. Решение.
9. Задача о красивых ковриках (раскраска в два цвета без квадратов  $2 \times 2$  одного цвета) за время  $O(4^n * (n+m))$  или  $O(8^n * \log m)$ .
10. Задача о наибольшей возрастающей последовательности. Решение за  $O(N \log N)$  с помощью дерева отрезков.
11. Задача о наибольшей возрастающей последовательности. Решение за  $O(N \log N)$  с помощью бинарного поиска.
12. Вычисление N-го числа Фибоначчи по заданному модулю за  $O(\log N)$
13. Нахождение  $A + A^2 + \dots + A^K$  за  $O(N^3 \log K)$  для матриц  $A(N \times N)$
14. Использование битового представления для кодирования множеств. Реализация операций над множествами.
15. Определение ориентированного и неориентированного графа, пути, вершинно простого пути, рёберно простого пути. Понятие цикла. Определение достижимости между вершинами. Определение связности.
16. Способы организации хранения графа, преимущества и недостатки.
17. Обход графов. Алгоритмы BFS и DFS.
18. Топологическая сортировка. Применение и алгоритм.
19. Обход DFS. Нерекурсивный DFS. Лемма о белых путях.
20. Отношение сильной связности. Компоненты связности. Алгоритм Косарайю. Конденсация графа.
21. Отношение сильной связности. Компоненты связности. Алгоритм Тарджана. Конденсация графа.
22. Классификация рёбер в обходе DFS. Отношение эквивалентности R. Нахождение мостов в графе.
23. Классификация рёбер в обходе DFS. Отношение эквивалентности R. Нахождение точек сочленения в графе.

Примеры билетов для проведения экзамена:

Билет 1.

1. Классификация хеш-функций. Cookoo-hash таблицы.

2. Два игрока играют в следующую игру: на доске написан ряд чисел. За один ход игрок стирает любое из чисел, находящихся с краю ряда и записывает его себе. Кто набрал большую сумму - выигрывает. Определите результат игры при наилучшей игре обеих сторон.

Билет 2.

1. Задача динамического программирования. Уравнение Беллмана.

2. На заданной клетчатой доске часть клеток недоступна. Определите наибольшее количество непересекающихся домино, которые можно разместить на данной доске.

### Критерии оценивания

Оценка "отлично (10)" выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины, проявляющему интерес к данной предметной области, продемонстрировавшему умение уверенно и творчески применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка "отлично (9)" выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка "отлично (8)" выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильное обоснование принятых решений, с некоторыми недочетами.

Оценка "хорошо (7)" выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но недостаточно грамотно обосновывает полученные результаты.

Оценка "хорошо (6)" выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности.

Оценка "хорошо (5)" выставляется студенту, если он в основном знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач достаточно большое количество неточностей.

Оценка "удовлетворительно (4)" выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он освоил основные разделы учебной программы, необходимые для дальнейшего обучения, и может применять полученные знания по образцу в стандартной ситуации.

Оценка "удовлетворительно (3)" выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, допускающему ошибки в формулировках базовых понятий, нарушения логической последовательности в изложении программного материала, слабо владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и с трудом применяет полученные знания даже в стандартной ситуации.

Оценка "неудовлетворительно (2)" выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных принципов и не умеет использовать полученные знания при решении типовых задач.

Оценка "неудовлетворительно (1)" выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубейшие ошибки в формулировках базовых понятий дисциплины и вообще не имеет навыков решения типовых практических задач.

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

Дифференцированный зачет проводится по итогам текущей успеваемости и сдачи заданий, предусмотренных программой дисциплины.

Дифференцированный зачет проводится в устной форме.

При проведении дифференцированного зачета обучающемуся предоставляется 20 минут на подготовку. Опрос обучающегося проводится в течение 30 минут.

Экзамен проводится в устной форме.

При проведении устного экзамена обучающемуся предоставляется один академический час на подготовку.

В случае недостаточно полных или неточных ответов на вопросы билета, преподаватель задает уточняющие вопросы.

Опрос обучающегося по билету на устном экзамене не должен превышать двух астрономических часов.

Во время проведения дифференцированного зачёта и экзамена обучающиеся могут пользоваться программой дисциплины, а также необходимыми для выполнения анализа программными и аппаратными моделями. Использование справочной литературы, конспектов лекций и др. не предусмотрено.