

**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**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Практикум по разработке и анализу алгоритмов
<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр

Семестры, формы промежуточной аттестации:

2 (весенний) - Дифференцированный зачет

3 (осенний) - Дифференцированный зачет

4 (весенний) - Дифференцированный зачет

Аудиторных часов: 90 всего, в том числе:

лекции: 0 час.

семинары: 90 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 180 час.

Всего часов: 270, всего зач. ед.: 6

Программу составил: С.Л. Бабичев, канд. физ.-мат. наук, доцент

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 22.07.2021

## Аннотация

Курс существенно опирается на математический аппарат, полученный студентами на курсе <<Математические основания алгоритмов и сложность вычислений>> и в качестве теоретической основы --- на курс <<Разработка и анализ алгоритмов>>.

В нём закрепляются на практике все теоретические положения, выдвинутые на указанных курсах. Реализуются все необходимые базовые структуры данных --- стек, дек, очередь, вектор. Показываются варианты реализации этих структур данных, анализируется сложность. На этой основе реализуются кучи --- бинарная, биномиальная а также деревья поиска AVL, splay и декартово. В рамках парадигмы CRUD реализуются алгоритмы хеш-таблиц, деревьев отрезков, деревьев Фенвика.

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 привитие практических навыков при проектировании и анализе алгоритмов и структур данных, воспитание дисциплины программирования, основанной на разделении задач на более простые с последующим их синтезом, подготовка студентов с практической реализацией алгоритмов и структур 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, ознакомление студентов с сложными алгоритмами, применяемыми в практической деятельности: задачи из теории чисел, поиска данных, сжатия информации, вычислительной геометрии и эвристические алгоритмы решения сложных задач. 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

- научить формулировать задачи в терминах изученных теорий, выбирать подходящий алгоритм для поставленной задачи;
- научить разрабатывать композиции алгоритмов для решения поставленных задач, оценивать сложности алгоритмов, выбирать подходящие структуры данных для поставленных задач, реализовывать алгоритмы на языке программирования С.
- закрепить на практике такие разделы алгоритмики, 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 С и С++.
- изучить такие разделы, как прикладные задачи теории чисел и комбинаторики, поисковые задачи на строках, сжатие текста, вычислительная геометрия и эвристические алгоритмы.

## 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-1.1 Способен анализировать поставленную задачу, намечать пути ее решения

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-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
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

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

В результате освоения дисциплины обучающиеся должны

знать:

- определения ассимптотик в О-нотации;
- способы реализации и применения линейных структур данных;
- алгоритмы быстрой и поразрядной сортировки и их применение;
- методы построения и основные свойства 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 минимальных остовных деревьев;
- алгоритмы нахождения кратчайших путей в графах;
- алгоритмы нахождения потоков в графах;
- алгоритмы нахождения минимальных разрезов и максимальных потоков в графах.
- строго доказывать утверждения о корректности алгоритмов;
- применять необходимую технику для решения алгоритмических задач;
- основы теории чисел в применении к информационным технологиям;
- быстрые алгоритмы работы с длинными числами и матрицами;
- применение алгоритмов теории чисел к задачам криптографии;
- методы комбинаторного поиска;
- Z-функцию и префикс-функцию и их применение;
- структуру данных бор и алгоритм Ахо-Корасик;
- суффиксное дерево и суффиксный автомат;
- алгоритмы сжатия информации: Huffman, LZ77, LZ78, BWT, Arithmetic coding;
- алгоритмы вычислительной геометрии: нахождение выпуклой оболочки, триангуляцию, поиск на плоскости;
- методы приближённого решения NP-сложных задач;

уметь:

- реализовывать на языке программирования Си такие структуры данных, как стек, дек, очередь, вектор, бинарная куча, случайное дерево поиска, сбалансированное дерево поиска.
- применять необходимые алгоритмы для решения конкретных задач;
- определять точки декомпозиции и синтеза в практической реализации алгоритмов;
- применять необходимую технику для решения алгоритмических задач;
- находить правильную организацию хеш-таблиц и верно выбирать хеш-функцию для решения задач, связанных с организацией данных и их поиском;
- выбирать верную декомпозицию задачи динамического программирования, приводящую к минимальному пространству аргументов;
- выбирать правильную организацию структур данных для представления графов для применения в конкретном алгоритме;
- выбирать наиболее эффективные алгоритмы для решения задач на графах;
- реализовывать эффективную работу с большими числами;
- осуществлять методы комбинаторного поиска;
- реализовывать алгоритмы быстрого поиска в строках и преобразовывать информацию для более компактного хранения (сжатие информации);
- реализовывать изученные методы вычислительной геометрии;
- решать NP-сложные задачи приближёнными методами.

владеть:

- разнообразными деревьями поиска и методикой выбора наиболее предпочтительного в каждой конкретной ситуации;
- методами доказательства корректности утверждений об алгоритмах;
- методами декомпозиции задач на более простые;
- методами синтеза решения сложных задач из простых.
- 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.
- техникой выбора необходимых алгоритмов для решения задач теории чисел;
- техникой быстрого поиска информации в текстовых данных;
- умением обратимо преобразовывать информацию в более компактное представление;
- методами аналитической геометрии для решения задач вычислительной геометрии;
- умением находить приближённые решения сложных задач эвристическими методами.

#### 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	Рекурсия и асимптотики.		5		10
2	Линейные и переработанные алгоритмы.		5		10
3	Сортировки и порядковые статистики.		5		10
4	Задача поиска. Поиск с сужением зоны. Деревья AVL, splay, декартово. Кучи.		5		10
5	Дерево отрезков и дерево Фенвика.		5		10
6	Хеш-функции, хеш-таблицы.		5		10
7	Хеш-функции и хеш-таблицы.		4		10

8	Динамическое программирование.		6		8
9	Представление графов и их обход.		4		8
10	Поиск кратчайших путей в графах.		4		8
11	Поиск минимальных остовных деревьев.		4		8
12	Паросочетания и потоки в графах.		4		8
13	Деревья.		4		10
14	Теория чисел		6		12
15	Быстрые алгоритмы на строках.		6		12
16	Алгоритмы сжатия текста.		6		12
17	Вычислительная геометрия.		6		12
18	Приближённое решение NP-сложных задач.		6		12
Итого часов			90		180
Подготовка к экзамену		0 час.			
Общая трудоёмкость		270 час., 6 зач.ед.			

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

Семестр: 2 (Весенний)

##### 1. Рекурсия и асимптотики.

Рекуррентные последовательности. Рекурсивные алгоритмы. Применение мастер-теоремы для определения сложности алгоритма Карацубы. Решение задач на рекурсию.

##### 2. Линейные и переработанные алгоритмы.

Нахождение жадных алгоритмов. Определение критерия корректности жадного алгоритма. Решение задач на жадные алгоритмы. Переборные алгоритмы - итеративные и рекурсивные. Перебор с возвратом (backtracking). Отсечение при переборе. Решение задач на переборные алгоритмы. Построение конечных автоматов и решение задач на конечные автоматы.

##### 3. Сортировки и порядковые статистики.

Реализации базовых структур данных. Реализация очередей на основе двух стеков. Связные списки и их реализация. Исследование алгоритмов квадратичных сортировок, сортировки Шелла и comb-сортировки. Реализация поиска порядковой статистики и быстрой сортировки в нескольких вариантах. Сравнение производительности вариантов. Реализация сортировок подсчётом и радикс-сортировок. Решение задач на сортировку.

##### 4. Задача поиска. Поиск с сужением зоны. Деревья AVL, splay, декартово. Кучи.

Ускорение линейного поиска методом граничного элемента. Реализация бинарного и тернарного поиска. Быстрый поиск экстремума в унимодальном массиве. Структура данных skip-list и её реализация. Решение задач на поиск. Реализация двоичной кучи и операций над ними. Построение деревьев поиска. Реализация операции поворота. Реализация сбалансированных деревьев на примере AVL. Использование декартовых деревьев. Реализация абстракции массива с логарифмической сложностью операций вставки и удаления на основе декартова дерева. Реализация методов работы со splay-деревьями.

##### 5. Дерево отрезков и дерево Фенвика.

Построение дерева отрезков. Реализация обновления информации снизу вверх и сверху вниз. Определение необходимости применения отложенных операций. Реализация двумерного, динамического и персистентного деревьев отрезков. Построение деревьев Фенвика. Изучение специфических побитовых алгоритмов. Сложные варианты деревьев Фенвика.

#### 6. Хеш-функции, хеш-таблицы.

Изучение различных хеш-функций. Исследование свойств хеш-функций - лавинности, равномерности. Построение семейства хеш-функций. Построение хеш-таблиц с открытой, закрытой адресацией и с рехешированием.

### Семестр: 3 (Осенний)

#### 7. Хеш-функции и хеш-таблицы.

Избранные вопросы реализации хеш-функций и хеш-таблиц. Cookoo-хеширование. Персистентные хеш-таблицы.

#### 8. Динамическое программирование.

Решение задачи динамического программирования. Прямой и обратный порядки решения. Выбор декомпозиции. Восстановление ответа. Многомерные случаи и контуры.

#### 9. Представление графов и их обход.

Исследование влияния организации графов на производительность. Обходы BFS и DFS. Нахождение компонентов связности, мостов и точек сочленения. Решение задач на компоненты связности, мосты и точки сочленения.

#### 10. Поиск кратчайших путей в графах.

Поиск кратчайшего расстояния в графах. Реализация алгоритмов Дейкстры, A\*, Флойда-Уоршалла, Форда-Беллмана. Поиск отрицательных циклов. Алгоритм Джонсона. Построение матрицы транзитивного замыкания. Возведение матрицы в степень.

#### 11. Поиск минимальных остовных деревьев.

Поиск минимальных основных деревьев. Алгоритм Прима и его реализация. Реализация структуры DSU и алгоритмов Краскала и Борушки. Решение задач на DSU и поиск остовных деревьев.

#### 12. Паросочетания и потоки в графах.

Двудольные графы и паросочетания. Реализация алгоритма Куна. Потоки в графах. Реализация алгоритмов Форда-Фалкерсона и Эдмондса-Карпа. Слоистая сеть. Реализация алгоритма Диница. Решение задач на паросочетания и потоки.

#### 13. Деревья.

Поиск диаметра дерева. Реализация алгоритмов определения изоморфизма графов. Варианты решения задачи LCA - обходом Эйлера и алгоритмом Фарах-Колтона-Бендера. Варианты решения задачи RMQ - за  $O(N \log N)$  предподсёта и за  $O(N)$  предподсчёта. Решение задач на деревья.

### Семестр: 4 (Весенний)

#### 14. Теория чисел

Решение задач на модульную арифметику, быстрое возведение в степень. Получение простых чисел алгоритмом решета Эратосфена. Решение задач на диофантовы уравнения. Применение малой теоремы Ферма и теоремы Эйлера для нахождения обратного элемента группы вычетов. Алгоритм Рабина для определения простоты числа. Разложение на простые множители с помощью алгоритмов ро-Полларда и Ферма. реализация быстрых алгоритмов умножения и деления длинных чисел. Реализация ДПФ. Реализация алгоритмов Диффи-Хеллмана и RSA.

#### 15. Быстрые алгоритмы на строках.

Представление строк. Построение Z- и префикс-функций и решение задач с их использованием. Быстрый поиск с использованием алгоритма Кнута-Морриса-Пратта. Использование rolling-hash для реализации алгоритма Карпа-Рабина и решение задач на rolling-hash. Построение бора для реализации алгоритма Ахо-Корасик. Построение суффиксных деревьев и суффиксных автоматов.

#### 16. Алгоритмы сжатия текста.

Реализация побитового ввода/вывода. Построение дерева Хаффмена и его хранение. Реализация алгоритма Хаффмена. Алгоритмы LZ78 и LZW. Алгоритм скользящего окна и его реализация с помощью хеш-таблиц. Интервальное и арифметическое кодирование. Контекстное кодирование. Задачи на наиболее сильное сжатие заданного текста.

#### 17. Вычислительная геометрия.

Реализация примитивов аналитической геометрии. Задачи на взаимное расположение геометрических объектов - точек, отрезков, лучей, прямых, многоугольников. Вычисление площадей многоугольников. Нахождение принадлежности точки многоугольникам - выпуклому и невыпуклому. Построение выпуклой оболочки алгоритмами Джарвиса, Грэхема, Эндрю. Триангуляция выпуклой оболочкой. Нахождение диаметра множества точек методом вращающихся калиперов. Нахождение наиболее близких точек множества на плоскости - два способа.

#### 18. Приближённое решение NP-сложных задач.

Точное и приближённое решение неполиномиальных алгоритмов. Перебор с отсечением. Альфа-бета алгоритм. Задача о вершинном покрытии графа. Имитация отжига. Обратные задачи. Реализация генетических алгоритмов методом роя частиц и методом дифференциальной эволюции. Задачи на наилучшее решение NP-трудной проблемы за ограниченное время.

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

Учебная аудитория с доской.

### **6. Перечень рекомендуемой литературы**

#### Основная литература

1. Алгоритмы : построение и анализ [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест .— М. : МЦНМО, 2001 .— 960 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. Вазириани ; пер. с англ. А. А. Куликова ; под ред. А. Шеня .— М. : МЦНМО, 2014 .— 320 с.

#### Дополнительная литература

Не предусмотрено

## **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 C; рекомендуется gcc;  
средства построения проектов; рекомендуются make и cmake;  
средства отладки и профилирования; рекомендуются valgrind и gprof.

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

В преподавании дисциплины сочетаются практические занятия в компьютерном классе и домашние задания, состоящие из теоретических задач и задач на программирование с автоматической проверкой корректности.



**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Информатика и вычислительная техника
<b>профиль подготовки:</b>	Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
<b>курс:</b>	1
<b>квалификация:</b>	бакалавр
Семестры, формы промежуточной аттестации:	
	2 (весенний) - Дифференцированный зачет
	3 (осенний) - Дифференцированный зачет
	4 (весенний) - Дифференцированный зачет
<b>Разработчик:</b>	С.Л. Бабичев, канд. физ.-мат. наук, доцент

## 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-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
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-4 Способен осуществлять сбор и обработку научно-технической и (или) технологической информации для решения фундаментальных и прикладных задач	ОПК-4.1 Владеет методами научного поиска и интеллектуального анализа информации при решении задач профессиональной деятельности
	ОПК-4.2 Знает основные источники научно-технической и (или) технологической информации в области профессиональной деятельности
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

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Практикум по разработке и анализу алгоритмов» обучающийся должен:

**знать:**

- определения ассимптотик в О-нотации;
- способы реализации и применения линейных структур данных;
- алгоритмы быстрой и поразрядной сортировки и их применение;
- методы построения и основные свойства 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 минимальных остовных деревьев;
- алгоритмы нахождения кратчайших путей в графах;
- алгоритмы нахождения потоков в графах;
- алгоритмы нахождения минимальных разрезов и максимальных потоков в графах.
- строго доказывать утверждения о корректности алгоритмов;
- применять необходимую технику для решения алгоритмических задач;
- основы теории чисел в применении к информационным технологиям;
- быстрые алгоритмы работы с длинными числами и матрицами;
- применение алгоритмов теории чисел к задачам криптографии;
- методы комбинаторного поиска;
- Z-функцию и префикс-функцию и их применение;
- структуру данных бор и алгоритм Ахо-Корасик;
- суффиксное дерево и суффиксный автомат;
- алгоритмы сжатия информации: Huffman, LZ77, LZ78, BWT, Arithmetic coding;
- алгоритмы вычислительной геометрии: нахождение выпуклой оболочки, триангуляцию, поиск на плоскости;
- методы приближённого решения NP-сложных задач;

#### **уметь:**

- реализовывать на языке программирования Си такие структуры данных, как стек, дек, очередь, вектор, бинарная куча, случайное дерево поиска, сбалансированное дерево поиска.
- применять необходимые алгоритмы для решения конкретных задач;
- определять точки декомпозиции и синтеза в практической реализации алгоритмов;
- применять необходимую технику для решения алгоритмических задач;
- находить правильную организацию хеш-таблиц и верно выбирать хеш-функцию для решения задач, связанных с организацией данных и их поиском;
- выбирать верную декомпозицию задачи динамического программирования, приводящую к минимальному пространству аргументов;
- выбирать правильную организацию структур данных для представления графов для применения в конкретном алгоритме;
- выбирать наиболее эффективные алгоритмы для решения задач на графах;
- реализовывать эффективную работу с большими числами;
- осуществлять методы комбинаторного поиска;
- реализовывать алгоритмы быстрого поиска в строках и преобразовывать информацию для более компактного хранения (сжатие информации);
- реализовывать изученные методы вычислительной геометрии;
- решать NP-сложные задачи приближёнными методами.

#### **владеть:**

- разнообразными деревьями поиска и методикой выбора наиболее предпочтительного в каждой конкретной ситуации;
- методами доказательства корректности утверждений об алгоритмах;
- методами декомпозиции задач на более простые;
- методами синтеза решения сложных задач из простых.
- 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.
- техникой выбора необходимых алгоритмов для решения задач теории чисел;
- техникой быстрого поиска информации в текстовых данных;
- умением обратимо преобразовывать информацию в более компактное представление;
- методами аналитической геометрии для решения задач вычислительной геометрии;
- умением находить приближённые решения сложных задач эвристическими методами.

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

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлого занятия.

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

2 семестр

1. Реализуйте персистентный стек. Определите сложность всех операций над ним.
2. Напишите программу, реализующую radix-сортировку 32-битных беззнаковых чисел, которая должна уложиться в заданный лимит времени.
3. Реализуйте хеш-таблицу с рехешированием при условии использования строк как ключей.

3 семестр

1. Написать программу, реализующую минимальное вершинное покрытие графа.
2. Напишите рекурсивный алгоритм, определяющий за линейное время содержится ли бинарное дерево совершенное паросочетание.
3. Дан граф  $G=(V,E)$ , в котором отмечены две вершины  $v_1$  и  $v_2$ . Постройте эффективный алгоритм поиска кратчайших путей между всеми парами вершин, проходящих через хотя бы одну из вершин  $v_1$  или  $v_2$ .

4 семестр

1. Напишите простую реализацию алгоритма Диффи-Хеллмана, не использующую арифметику длинных чисел.
2. Для заданной строки покажите её преобразование по алгоритму BWT.
3. Реализуйте задачу о вершинном покрытии графа с помощью какого-либо неполиномиального алгоритма.

### Критерии 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.

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

Оценка выставляется по совокупности измерений знаний по домашним работам, по контрольным работам и по активности студентов на семинарах.

Для проведения контрольной работы выделяется один академический час. Работа оформляется письменно на листах бумаги. При проведении контрольной работы допускается использовать данную программу.