

**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)
по дисциплине:	Разработка и анализ алгоритмов
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	1
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

2 (весенний) - Экзамен

3 (осенний) - Экзамен

4 (весенний) - Экзамен

Аудиторных часов: 210 всего, в том числе:

лекции: 90 час.

семинары: 0 час.

лабораторные занятия: 120 час.

Самостоятельная работа: 285 час.

Подготовка к экзамену: 90 час.

Всего часов: 585, всего зач. ед.: 13

Количество контрольных работ, заданий: 9

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

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 22.07.2021

Аннотация

Курс представляет собой первую часть большого курса по разработке и анализу алгоритмов. Она существенно опирается на математический аппарат, полученный студентами на курсе «Математические основания алгоритмов и сложность вычислений». Семестр посвящён изучению структур данных, необходимых для разнообразных алгоритмов. Простейшие структуры стек, очередь, вектор анализируются на предмет эффективности и времени выполнения. Вводятся кучи (двоичная, биномиальная и фибоначиева), 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.

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 первичное ознакомление студентов с основными принципами проектирования и анализа алгоритмов и структур данных, обучение навыкам 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;
- изучить такие разделы, как хеш-таблицы, динамическое программирование и графы;
- изучить такие разделы, как прикладные задачи теории чисел и комбинаторики, поисковые задачи на строках, сжатие текста, вычислительная геометрия и эвристические алгоритмы.
- 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-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
	УК-2.2 Проектирует решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений
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 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 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:

- 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;
 - 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 алгоритмом 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	Асимптотики, мастер-теорема.	2		4	9
2	Линейные алгоритмы.	2		4	9
3	Линейные структуры данных.	2		4	9
4	Сортировки и порядковые статистики.	4		8	13
5	Задача поиска.	2		4	9
6	Деревья и кучи.	4		8	12
7	Деревья поиска.	6		12	18
8	Дерево отрезков, дерево Фенвика.	4		8	13
9	Хэш-таблицы, фильтры Блума.	4		8	13
10	Хеш-функции и хеш-таблицы.	2		2	6
11	Задача динамического программирования.	2		2	6
12	Выбор декомпозиции.	2		2	6
13	Многомерные варианты.	2		2	6
14	Графы, их организация.	2		2	6
15	Отношение сильной связности.	2		2	6
16	Мосты и точки сочленения.	2		2	6
17	Поиск кратчайшего расстояния в графах.	2		2	6
18	Алгоритм Флойда-Уоршалла.	2		2	6
19	Поиск минимальных основных деревьев.	2		2	6
20	Паросочетания в произвольном графе.	2		2	6
21	Потоки в графах.	2		2	6
22	Слоистая сеть.	2		2	6
23	Деревья.	2		2	6

24	Задача RMQ.	2		2	6
25	Теория чисел.	5		5	15
26	Строки. Z-функция.	5		5	15
27	Структура данных бор.	5		5	15
28	Алгоритмы сжатия текста.	5		5	15
29	Вычислительная геометрия.	5		5	15
30	Приближённое решение неполиномиальных алгоритмов.	5		5	15
Итого часов		90		120	285
Подготовка к экзамену		90 час.			
Общая трудоёмкость		585 час., 13 зач.ед.			

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

Семестр: 2 (Весенний)

1. Асимптотики, мастер-теорема.

Обозначения в O-нотации: о-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. 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

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, радикас-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 пятёрок.

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, 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: 3 (Осенний)

10. Хеш-функции и хеш-таблицы.

Хеширование во внешней памяти. Применение хеширования для задач поиск дублирующейся информации. Сооккоо хеш-таблицы.

11. Задача динамического программирования.

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

12. Выбор декомпозиции.

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

13. Многомерные варианты.

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

14. Графы, их организация.

Обход графов. Алгоритмы BFS и DFS. Лемма о белых путях. Топологическая сортировка.

15. Отношение сильной связности.

Компоненты связности. Алгоритмы Косарайю и Тарджана. Конденсация графа.

16. Мосты и точки сочленения.

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

17. Поиск кратчайшего расстояния в графах.

Алгоритм Дейкстры. Реализации алгоритма Дейкстры. Алгоритм A^* . Допустимые и монотонные эвристики. Корректность и сложность.

18. Алгоритм Флойда-Уоршалла.

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

19. Поиск минимальных основных деревьев.

Алгоритм Прима. Лемма о безопасном ребре. Система непересекающихся множеств. Алгоритм Краскала. Алгоритм Борувки.

20. Паросочетания в произвольном графе.

Двудольные графы. Понятие увеличивающего пути. Теорема Бержа. Алгоритм поиска максимального паросочетания в двудольном графе. 2SAT.

21. 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.

22. Слоистая сеть.

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

23. Деревья.

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

24. Задача RMQ.

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

Семестр: 4 (Весенний)

25. Теория чисел.

Модульная арифметика. Быстрое возведение в степень. Схема Горнера. Решето Эратосфена. Алгоритм Евклида. Диофантовы уравнения. Побитовое нахождение НОД. Порядок элемента по модулю. Определение простоты числа. Факторизация. Алгоритмы арифметики длинных чисел. Нахождение первообразного корня. Прямое и обратное дискретное преобразование Фурье. Быстрое умножение многочленов. Быстрое умножение матриц. Алгоритм Штрассена. Комбинаторный поиск. Генерирование всех перестановок. Алгоритм Диффи-Хеллмана. Алгоритм RSA.

26. Строки. Z-функция.

Префикс-функция. Преобразование Z-функции в префикс-функцию и обратно. Алгоритм Кнута-Мориса-Пратта. Rolling-hash. Алгоритм Карпа-Рабина.

27. Структура данных бор.

Алгоритм Ахо-Корасик. Сопоставление строк по образцу. Вхождение образца в строку. Подсчёт числа вхождений. Суффиксное дерево и суффиксный автомат.

28. Алгоритмы сжатия текста.

Алгоритм Хаффмена. Алгоритм LZ78. Алгоритм LZ77 с вариациями (LZSS). Преобразование Burrows-Wheeler. Арифметическое кодирование. Контекстное кодирование.

29. Вычислительная геометрия.

Представление геометрических примитивов. Скалярное и векторное произведения. Пересечение двух прямых. Пересечение прямой и окружности. Пересечение двух окружностей. Пересечение двух отрезков. Площади многоугольников. Метод шнуровки. Нахождение принадлежности точки выпуклому многоугольнику. Нахождение принадлежности точки произвольному многоугольнику. Метод пересекающей прямой. Выпуклая оболочка. Алгоритмы Джарвиса, Грэхема, Эндрю. Триангуляция выпуклой оболочкой. Нахождение диаметра множества точек. Метод вращающихся калиперов. Нахождение кратчайшего расстояния между точками множества: жадный алгоритм и алгоритм разделяй и властвуй.

30. Приближённое решение неполиномиальных алгоритмов.

Перебор с отсечением. Альфа-бета алгоритм. Задача о вершинном покрытии графа. Имитация отжига. Обратные задачи. Генетические алгоритмы. Метод роя частиц. Метод дифференциальной эволюции.

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. Вазирани ; 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. «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)

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)

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

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-2 Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений	УК-2.1 Формулирует совокупность взаимосвязанных задач в рамках поставленной цели работы, обеспечивающих ее достижение. Определяет ожидаемые результаты решения поставленных задач
	УК-2.2 Проектирует решение конкретной задачи проекта, выбирая оптимальный способ ее решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений
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 «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 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:

- 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;
- 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

для 2-го семестра

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.

для 3-го семестра

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.

для 4-го семестра

1. Найдите произведение данных многочленов [многочлены] с использованием дискретного преобразования Фурье.
2. Определите последовательность токенов в алгоритме LZ77 для следующего текста: [дан фрагмент текста]
3. Приведите пример множества точек, для которого жадный алгоритм нахождения минимального расстояния между точками сходится медленно.

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

Пример экзаменационных вопросов для 2-го семестра

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 партииции 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. B- и B+-деревья. Организация. Применение. Алгоритмы CRUD над B-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.
25. Докажите, что невозможно создать алгоритм сортировки сравнениями, алгоритмическая сложность которого была бы меньше $O(N \log N)$.
26. Сформулируйте и докажите теорему об универсальном семействе хеш-функций.
27. Докажите, что средняя высота случайного дерева поиска есть $O(N)$.

Пример экзаменационного билета.

1. Алгоритм нахождения k -й порядковой статистики. Оценка сложности при рандомизированном выборе ведущего элемента.
2. Организация хеш-таблиц с открытой адресацией. Амортизированная сложность операций поиска, вставки, удаления.

перечень вопросов для 3-го семестра

- 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.

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

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.

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

1. 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.
2. На заданной клетчатой доске часть клеток недоступна. Определите наибольшее количество непересекающихся домино, которые можно разместить на данной доске.
3. Во взвешенном неориентированном графе найдите такое минимальное число k такое, что в любом минимальном остовном дереве содержится хотя бы одно из этих рёбер.

Пример экзаменационных вопросов.

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 2×2 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.

пример экзаменационного билета

1. 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.
2. На заданной клетчатой доске часть клеток недоступна. Определите наибольшее количество непересекающихся домино, которые можно разместить на данной доске.
3. Во взвешенном неориентированном графе найдите такое минимальное число k такое, что в любом минимальном остовном дереве содержится хотя бы одно из этих рёбер.

примеры экзаменационных вопросов для 4-го семестра

- 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.

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

1. Найдите произведение данных многочленов [многочлены] с использованием дискретного преобразования Фурье.
2. Определите последовательность токенов в алгоритме LZ77 для следующего текста: [дан фрагмент текста]
3. Приведите пример множества точек, для которого жадный алгоритм нахождения минимального расстояния между точками сходится медленно.

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

1. Построить Z-функцию для данной строки.
2. На столе находятся блины - круги с заданными центром и радиусом. Определите, сколько блинов будет разрезано лучом, проведённым из заданной точки в другую заданную.
3. На плоскости заданы множество P точек. Определить для заданных точек ближайшие к ним из множества P. Установите сложность предподсчёта, если он нужен, и сложность поиска.

Пример экзаменационных вопросов.

1. Алгоритм побитового нахождения НОД. Доказательство корректности и определение сложности.
2. Дискретное преобразование Фурье. Алгоритмы и применение.
3. Алгоритмы деления длинных чисел.
4. Алгоритм Штрассена. Доказательство его корректности и определение сложности.
5. Вычисление функции Эйлера для заданного числа.
6. Алгоритм Диффи-Хеллмана и его применения.
7. Алгоритм RSA и его применения.
8. Z-функция. Свойства и вычисление за $O(N)$. Применение.
9. Префикс-функция. Свойства и вычисление за $O(N)$. Применение. Алгоритм Кнута-Мориса-Пракса.
10. Структура данных бор. Алгоритм Ахо-Корасик. Применение.
11. Суффиксное дерево и суффиксный автомат.
12. Алгоритм Хаффмена. Кодирование таблицы частот.
13. Алгоритм LZ78. Варианты алгоритма. Борьба с переполнением.
14. Алгоритм LZ77. Алгоритмы быстрого поиска подстрок. Варианты.
15. Арифметическое кодирование. Применение в контекстном сжатии.
16. Вычисление площади ориентированного многоугольника. Формула Пика. Алгоритм шнуровки.
17. Определение принадлежности точки многоугольнику: выпуклому и невыпуклому.
18. Нахождение выпуклой оболочки. Алгоритм Джарвиса.
19. Нахождение выпуклой оболочки. Алгоритм Грэхема.
20. Алгоритм триангуляции множества точек на плоскости.
21. Определение диаметра множества точек на плоскости.
22. Нахождение самых близких точек на плоскости: два варианта поиска.
23. Перебор с отсечением. Альфа-бета алгоритм.
24. Приближённое решение NP-сложных. Имитация отжига.
25. Приближённое решение NP-сложных задач. Генетические алгоритмы.

пример экзаменационного билета

1. Построить Z-функцию для данной строки.
2. На столе находятся блины - круги с заданными центром и радиусом. Определите, сколько блинов будет разрезано лучом, проведённым из заданной точки в другую заданную.
3. На плоскости заданы множество P точек. Определить для заданных точек ближайшие к ним из множества P. Установите сложность предподсчёта, если он нужен, и сложность поиска.

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

При проведении устного экзамена обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося по билету на устном экзамене не должен превышать двух астрономических часов.

Во время проведения экзамена обучающиеся могут пользоваться программой дисциплины.