

**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:	Алгоритмы и структуры данных. Продвинутый поток
по направлению:	Прикладная математика и информатика
профиль подготовки:	А1360: Передовые методы искусственного интеллекта Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	1
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

1 (осенний) - Экзамен

2 (весенний) - Экзамен

Аудиторных часов: 120 всего, в том числе:

лекции: 60 час.

семинары: 60 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 90 час.

Подготовка к экзамену: 60 час.

Всего часов: 270, всего зач. ед.: 6

Количество контрольных работ, заданий: 4

Программу составил: И.Д. Степанов, ассистент

Программа обсуждена на заседании кафедры алгоритмов и технологий программирования 05.08.2025

Аннотация

В начале курса вводятся общие математические обозначения и методы анализа сложности, позволяющие работать с асимптотиками и оценивать эффективность алгоритмов. Далее изучаются базовые структуры данных (стек, очередь, вектор) с анализом их времени работы, вводятся кучи (двоичная, биномиальная, фибоначчиева) с обсуждением границ применимости. Рассматриваются деревья поиска (splay, AVL, декартово, B-дерево) с доказательствами корректности, анализом асимптотики и примерами прикладного использования. Изучаются универсальные техники обработки запросов: хэш-таблицы, деревья отрезков, деревья Фенвика, разреженные таблицы.

Отдельное внимание уделяется алгоритмам на графах, включая поиск максимального потока в транспортных сетях (в том числе минимальной стоимости), а также алгоритмам на строках и вычислительной геометрии. Эти разделы демонстрируют, каким образом фундаментальные структуры и алгоритмы применяются для решения прикладных задач анализа и обработки данных.

Учебный процесс построен на сочетании лекций, интерактивных семинаров, практических мастер-классов и проектных заданий. Важной частью дисциплины является выполнение практических кейсов, участие в конкурсах, хакатонах и командных проектах. Привлечение специалистов из промышленных партнёров (Авито, Сбер, Яндекс) позволяет студентам работать с реальными задачами обработки данных и интеграции алгоритмов в прикладные системы искусственного интеллекта.

Освоение дисциплины способствует формированию компетенций по работе с данными: определению требований к наборам данных, оценке их качества, применению инструментов DataOps, а также выбору технологий обработки больших данных и разработке прикладных решений с элементами ИИ.

Курс базируется на знаниях, полученных при изучении дисциплины «Введение в математический анализ», и является предшествующей для дисциплины «Машинное обучение».

Дисциплина входит в состав образовательных программ по треку «Искусственный интеллект» и реализуется на первом курсе.

1. Цели и задачи

Цель дисциплины

Цель дисциплины «Алгоритмы и структуры данных» — формирование у студентов знаний и навыков проектирования, анализа и реализации алгоритмов и структур данных, обоснования их корректности и оценки сложности, а также развитие компетенций в области работы с данными и применения алгоритмов для решения задач искусственного интеллекта и цифровых технологий.

Задачи дисциплины

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

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

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

ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.3 Знает основные требования информационной безопасности
BD-2 Способен определять требования к наборам данных для решения задач машинного обучения проводить разметку и анализ наборов данных оценивать качество данных обеспечивать непрерывную интеграцию данных	BD-2.1 Определяет требования к наборам и качеству данных для решения задач машинного обучения
	BD-2.2 Работает с данными, в том числе собирает данные из разрозненных источников, проверяет данные на корректность
	BD-2.3 Применяет инструменты и практики непрерывной интеграции данных (DataOps)
BD-4 Способен применять различные модели и (или) технологии обработки данных	BD-4.1 Осуществляет выбор технологий обработки больших данных, приемлемых для создания прикладной системы ИИ с заданными требованиями
	BD-4.2 Разрабатывает и отлаживает прикладные решения с элементами ИИ с применением различных технологий обработки данных
	BD-4.3 Тестирует, испытывает и оценивает качество решений с элементами ИИ, реализованных с использованием технологий обработки данных

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

уметь:

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

владеть:

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

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

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

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост.

		лекции	семинары	лаборат. работы	работа
1	Асимптотики, мастер-теорема	7	7		12
2	Линейные структуры данных	8	8		12
3	Сортировки и порядковые статистики	7	7		11
4	Кучи	8	8		10
5	Деревья поиска	6	6		10
6	Дерево отрезков, дерево Фенвика	6	6		10
7	Хэш-таблицы, фильтры Блума	6	6		10
8	Паросочетания, алгоритм Куна	6	6		8
9	Максимальные потоки в сетях	6	6		7
Итого часов		60	60		90
Подготовка к экзамену		60 час.			
Общая трудоёмкость		270 час., 6 зач.ед.			

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

Семестр: 1 (Осенний)

1. Асимптотики, мастер-теорема

Форматы и методы:

Конкурсы индивидуальных проектов: «Оптимизация алгоритма сортировки с $O(n^2)$ до $O(n \log n)$ », «Сравнение рекурсивных и итеративных решений одной задачи», «Визуализация роста функций O -большое».

Групповые проекты: «Каталог алгоритмов с оценками сложности и примерами применения», «Анализ реальных алгоритмов из стандартной библиотеки Python/C++».

Мастер-классы от индустриального партнёра: «Как асимптотика влияет на выбор алгоритмов в Big Data и высоконагруженных системах» (Яндекс).

Практические кейсы от индустриального партнёра: «Оптимизация обработки запросов к поисковой системе» (Яндекс), «Выбор алгоритмов для обработки транзакций в реальном времени» (Сбер).

Хакатоны: «Code Optimization Hack: ускорение рабочего алгоритма в 10 раз».

2. Линейные структуры данных

Форматы и методы:

Лекция: Стек, очередь, дек, их реализация.

Конкурсы индивидуальных проектов: «Реализация дека с динамическим расширением», «Стек с функцией поиска минимума за $O(1)$ ».

Групповые проекты: «Симулятор очередей в распределённой системе», «Реализация планировщика задач с приоритетами».

Мастер-классы от индустриального партнёра: «Очереди в распределённых системах (RabbitMQ, Kafka)» (Сбер/Яндекс.Cloud).

Практические кейсы: «Организация очередей для модерации объявлений» (Авито), «Буферизация транзакций в банковских системах» (Сбер).

Хакатоны: «Task Queue Sprint: создать прототип очереди обработки запросов».

3. Сортировки и порядковые статистики

Форматы и методы:

Лекция: Сравнение сортировок (MergeSort, QuickSort, CountingSort).

Конкурсы индивидуальных проектов: «Реализация детерминированного QuickSort (медиана медиан)», «Сравнение стабильных и нестабильных сортировок».

Групповые проекты: «Модуль сортировок для базы данных», «Система ранжирования заказов/объявлений».

Мастер-классы: «Сортировки в базах данных и индексах (PostgreSQL, Redis)» (Сбер/Яндекс).

Практические кейсы: «Оптимизация выдачи поисковых результатов» (Яндекс), «Сортировка финансовых транзакций по времени и сумме» (Сбер).

Хакатоны: «Sorting Challenge: самая быстрая сортировка на большом массиве данных».

4. Кучи

Форматы и методы:

Лекция: Бинарная, биномиальная, фибоначчиева кучи.

Конкурсы индивидуальных проектов: «Реализация биномиальной и фибоначчиевой кучи», «Куча с поддержкой удаления произвольного элемента».

Групповые проекты: «Применение куч для приоритетного планировщика задач», «Реализация алгоритма Дейкстры с бинарной кучей».

Мастер-классы: «Кучи в системах планирования (Kubernetes, ОС, диспетчеры задач)» (Яндекс.Cloud).

Практические кейсы: «Оптимизация логистических маршрутов» (Авито), «Планирование финансовых транзакций в банке» (Сбер).

Хакатоны: «Heap Hack: реализовать кучу для задач маршрутизации и планирования».

Семестр: 2 (Весенний)

5. Деревья поиска

Форматы и методы:

Лекция: AVL, Splay, Декартово дерево.

Конкурсы индивидуальных проектов: «Балансировка AVL-дерева», «Реализация Splay-дерева».

Групповые проекты: «Реализация B-дерева для СУБД», «Поисковая система на основе сбалансированных деревьев».

Мастер-классы: «Деревья в индексации и поиске (PostgreSQL, Elasticsearch)» (Яндекс).

Практические кейсы: «Построение индекса для поисковой системы» (Яндекс), «Деревья в аналитических базах данных» (Сбер).

Хакатоны: «Index Tree Sprint: построить оптимизированный индекс для большого массива данных».

6. Дерево отрезков, дерево Фенвика

Форматы и методы:

Лекция: RMQ, отложенные операции.

Конкурсы индивидуальных проектов: «Реализация двумерного дерева отрезков», «Динамическое дерево Фенвика».

Групповые проекты: «Система аналитики запросов к массивам данных», «Применение деревьев отрезков в статистике реального времени».

Мастер-классы: «Деревья отрезков и Фенвика в аналитике больших данных» (Сбер/Яндекс).

Практические кейсы: «Агрегация показов рекламы в реальном времени» (Авито), «Операции над временными рядами в банке» (Сбер).

Хакатоны: «Segment Tree Challenge: быстрые запросы к массиву данных в реальном времени».

7. Хэш-таблицы, фильтры Блума

Форматы и методы:

Лекция: Коллизии, двойное хэширование.

Конкурсы индивидуальных проектов: «Реализация открытой адресации и двойного хэширования», «Фильтр Блума для проверки орфографии».

Групповые проекты: «Хэш-индексация для базы данных», «Прототип кэш-системы».

Мастер-классы: «Хэш-таблицы в распределённых системах (Cassandra, Redis)» (Яндекс).

Практические кейсы: «Дедупликация объявлений» (Авито), «Оптимизация кэширования запросов в транзакционной системе» (Сбер).

Хакатоны: «Bloom Hack: реализовать быстрый фильтр для миллионов запросов».

8. Паросочетания, алгоритм Куна

Форматы и методы:

Конкурсы индивидуальных проектов: «Реализация алгоритма Куна», «Решение задачи о назначениях».

Групповые проекты: «Приложение для оптимизации подбора персонала», «Алгоритмы подбора заказов к исполнителям».

Мастер-классы: «Паросочетания и маршрутизация (Uber, Яндекс.Такси)» (Яндекс).

Практические кейсы: «Оптимизация распределения курьеров по заказам» (Яндекс/Авито), «Подбор персонала в HR-системах» (Сбер).

Хакатоны: «Matching Hack: оптимизация назначений в реальном времени»

9. Максимальные потоки в сетях

Форматы и методы:

Лекция: Теорема Форда-Фалкерсона.

Конкурсы индивидуальных проектов: «Реализация алгоритма Диница», «Сравнение алгоритмов Форда-Фалкерсона и Диница».

Групповые проекты: «Симулятор транспортной сети», «Оптимизация потоков в телекоммуникациях».

Мастер-классы: «Максимальные потоки в сетях: от логистики до телекома» (Сбер/Яндекс).

Практические кейсы: «Оптимизация потоков данных в распределённой системе» (Яндекс), «Моделирование потоков транзакций в банке» (Сбер).

Хакатоны: «Network Flow Sprint: оптимизация маршрутов доставки».

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

Для проведения занятий по дисциплине используется учебная аудитория, оснащённая компьютерной техникой с необходимым программным обеспечением и мультимедийным оборудованием (проектор, звуковая система).

Разбор кейсов, групповые обсуждения, проектные практикумы и мастер-классы реализуются посредством облачных SaaS-сервисов с бесплатным или условно-бесплатным доступом, а также при поддержке ресурсной базы индустриальных партнёров (Авито, Сбер, Яндекс).

Привлечение специалистов из индустрии осуществляется посредством программного обеспечения для организации видеоконференций и дистанционного взаимодействия, что позволяет проводить лекции, мастер-классы и консультации ИТ-практиков.

Для выполнения практических заданий и проектной работы используются современные инструменты разработки (Python 3.x, C++, PyCharm, VS Code, Jupyter Notebook), библиотеки для алгоритмов и структур данных, системы визуализации и профилирования, а также системы контроля версий (GitHub, GitLab).

Опционально, по согласованию с индустриальными партнёрами, студенты могут проходить обучение и выполнять проектные задания на площадках партнёров в дистанционном формате, с использованием специализированных облачных сервисов, корпоративных платформ совместной разработки и хранилищ данных. Такой формат обеспечивает доступ к актуальной индустриальной инфраструктуре и позволяет студентам закрепить навыки работы с инструментами, применяемыми в реальной профессиональной среде.

6. Перечень рекомендуемой литературы

Основная литература

1. Введение в теорию алгоритмов и структур данных, Электронная версия печатной публикации / М. А. Бабенко, М. В. Левин. — Москва, МЦНМО, 2016

Дополнительная литература

Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. — М.: Вильямс, 2022.

Седжвик Р., Уэйн К. Алгоритмы на Java / C++ / Python (последние издания). — СПб.: Питер, 2021.

Добрая Н. Структуры данных и алгоритмы в Python. — СПб.: Питер, 2020.

Скiena С. Алгоритмы. Руководство по разработке. — М.: Вильямс, 2020.

Тарян Р. Алгоритмы на графах. — М.: Мир, 2019.

Кнут Д. Искусство программирования. Том 1–3. — М.: Вильямс, 2018.

Литература по треку «Искусственный интеллект»:

Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. — М.: ДМК Пресс, 2020.

Чоллет Ф. Глубокое обучение на Python. — СПб.: Питер, 2019.

Рассел С., Норвиг П. Искусственный интеллект: современный подход. — М.: Вильямс, 2021.

Кристофер Бишоп. Распознавание образов и машинное обучение. — М.: ДМК Пресс, 2020.

Научные публикации (пример):

Postnicov V. et al. Evaluation of three-point correlation functions from structural images on CPU and GPU architectures: Accounting for anisotropy effects // Physical Review E. — 2024. — Т. 110. — №. 4. — С. 045306. DOI: 10.1103/PhysRevE.110.045306

Akhtyamov P. et al. GPU-accelerated Kendall distance computation for large or sparse data // GigaScience. — 2024. — Т. 13. — С. giae088. DOI: 10.1093/gigascience/giae088

Kozhemyachenko A. et al. Modification of the grid-characteristic method on chimera meshes for 3D problems of railway non-destructive testing // Lobachevskii Journal of Mathematics. — 2023. — Т. 44. — №. 1. — С. 376–386. DOI: 10.1134/S1995080223010262

Ковалев Д., Безносиков А., Бородич Е. и др. Оптимальное градиентное скольжение и его применение к распределенной оптимизации // arXiv:2205.15136 [math.OC], 2022. URL: <https://arxiv.org/abs/2205.15136>

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

Официальная документация Python: <https://docs.python.org/3/>
Стандартные библиотеки C++ (STL): <https://en.cppreference.com/w/>
Visualgo (визуализация алгоритмов): <https://visualgo.net/>
Project Euler (задачи по алгоритмам): <https://projecteuler.net/>
Документация по библиотекам для анализа данных и ИИ:
NumPy: <https://numpy.org/doc/>
pandas: <https://pandas.pydata.org/docs/>
scikit-learn: <https://scikit-learn.org/stable/>
PyTorch: <https://pytorch.org/docs/>
TensorFlow: https://www.tensorflow.org/api_docs

Ресурсы промышленных партнёров: открытые API, технические блоги и образовательные материалы.

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

Для реализации дисциплины используются:

языковые средства и среды разработки (Python 3.x, C++, PyCharm, VS Code, Jupyter Notebook);
библиотеки и пакеты для реализации и визуализации алгоритмов (NumPy, pandas, matplotlib, graphviz, networkx);
системы управления данными и версиями (SQLite, PostgreSQL, Git, GitHub/GitLab);
облачные и SaaS-сервисы (Google Colab, Kaggle, Яндекс.Cloud, СберCloud) для выполнения проектных заданий и дистанционных практик;
SaaS-сервисы совместной работы (Miro, Notion, Trello, Slack или их российские аналоги);
программное обеспечение для организации видеоконференций (Zoom, MS Teams, Google Meet) для дистанционного участия специалистов промышленных партнёров в мастер-классах и консультациях;
системы дистанционного обучения (Moodle, Canvas, Blackboard);
открытые API и тестовые данные промышленных партнёров для выполнения практических кейсов.

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

Освоение дисциплины требует регулярного выполнения практических заданий и самостоятельной работы с материалами лекций и семинаров.

Особое внимание следует уделять алгоритмам и структурам данных с последующей их реализацией на языке программирования (Python или C++), а также проверке корректности и анализу временной и пространственной сложности.

Для закрепления теоретического материала рекомендуется активно участвовать в конкурсах индивидуальных и групповых проектов, решении практических кейсов и хакатонах.

При выполнении проектных заданий необходимо использовать системы контроля версий (GitHub/GitLab), облачные сервисы (Google Colab, Яндекс.Cloud, СберCloud) и совместные SaaS-платформы (Miro, Notion, Trello, Slack или российские аналоги).

Для подготовки к занятиям рекомендуется обращаться к официальной документации языков и библиотек, а также к справочным материалам промышленных партнёров.

Участие в мастер-классах и консультациях специалистов индустрии является обязательным элементом освоения дисциплины, так как обеспечивает практическую связь теоретических знаний с реальными задачами ИИ и цифровых технологий.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладная математика и информатика
профиль подготовки:	АІ360: Передовые методы искусственного интеллекта Физтех-школа Прикладной Математики и Информатики кафедра алгоритмов и технологий программирования
курс:	<u>1</u>
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

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

Разработчик: И.Д. Степанов, ассистент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять фундаментальные знания, полученные в области физико-математических и (или) естественных наук и использовать их в профессиональной деятельности	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-2 Способен использовать современные информационные технологии и программные средства при решении задач профессиональной деятельности, соблюдая требования информационной безопасности	ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области
	ОПК-2.1 Способен применять современные вычислительную технику и сервисы сети Интернет в области (сфере) профессиональной деятельности
	ОПК-2.3 Знает основные требования информационной безопасности
BD-2 Способен определять требования к наборам данных для решения задач машинного обучения проводить разметку и анализ наборов данных оценивать качество данных обеспечивать непрерывную интеграцию данных	BD-2.1 Определяет требования к наборам и качеству данных для решения задач машинного обучения
	BD-2.2 Работает с данными, в том числе собирает данные из разрозненных источников, проверяет данные на корректность
	BD-2.3 Применяет инструменты и практики непрерывной интеграции данных (DataOps)
BD-4 Способен применять различные модели и (или) технологии обработки данных	BD-4.1 Осуществляет выбор технологий обработки больших данных, приемлемых для создания прикладной системы ИИ с заданными требованиями
	BD-4.2 Разрабатывает и отлаживает прикладные решения с элементами ИИ с применением различных технологий обработки данных
	BD-4.3 Тестирует, испытывает и оценивает качество решений с элементами ИИ, реализованных с использованием технологий обработки данных

2. Показатели оценивания компетенций

В результате изучения дисциплины «Алгоритмы и структуры данных. Продвинутый поток (ФПМИ)» обучающийся должен:

знать:

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

уметь:

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

владеть:

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

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

1 семестр:

1. По данному числу n найдите все пары целых положительных чисел (a, b) , такие что $a \leq b \leq n$, и $a \mid b$. Оцените асимптотическое поведение числа таких пар.
2. Число 0 записано в n -разрядной двоичной системе. К нему $2^n - 1$ раз прибавляется единица. Будем считать, что время, необходимое на прибавление единицы, равно количеству единиц в двоичной записи числа, которые становятся нулями. Оцените среднюю сложность всех таких операций. Какие операции являются самыми дешёвыми, а какие -- самыми дорогими?
3. Предложите метод хранения минимального значения в очереди с помощью структуры `deque`. Если к очереди поступило n запросов, время работы программы должно составлять $O(n)$.
4. Напомним, что процедура `Partition(A, x)` переупорядочивает элементы массива A так, что сначала идут все элементы, не превосходящие x , в некотором порядке, а затем -- все элементы, большие x . Покажите, как реализовать `Partition(A, x)` с привлечением $O(1)$ дополнительной памяти.
5. Нижняя оценка на число сравнений в сортировке, основанной на сравнениях.
6. Лемма о корректности `siftDown` и `siftUp`.
7. Удаление в куче (по указателю и по значению).
8. Двумерное дерево отрезков снизу: прибавление в точке и сумма в прямоугольнике.
9. Фибоначчиева куча: операций `decreaseKey`. Асимптотика и корректность.
10. Фильтры Блума: асимптотика и корректность.

2 семестр:

1. Сведите задачу поиска максимального паросочетания в двудольном графе к поиску максимального потока в некоторой сети. Сравните время работы алгоритмов Форда--Фалкерсона и Эдмондса--Карпа. Можно ли проделать то же в произвольном (необязательно двудольном) графе?
2. На фабрику поступило n заказов, реализация i -го из которых принесёт ей прибыль в a_i рублей. Каждый заказ для производства требует некоторого набора инструментов. Пусть всего зависимостей “заказ--инструмент” ровно k . Инструменты можно переиспользовать, то есть задействовать в нескольких заказах. Однако в данный момент на фабрике вообще нет инструментов, так что покупка j -го из них обойдётся в b_j рублей. Конечно, от приёма каких-то заказов или покупки каких-то инструментов фабрика может отказаться. Какие заказы следует реализовать для максимизации прибыли (с учётом затрат на инструменты)?
3. Задан полный взвешенный двудольный граф, в обеих долях которого находится по n вершин. Предложите алгоритм поиска самого дешёвого совершенного паросочетания.
4. С помощью суффиксного автомата найдите количество различных подстрок строки s за время, линейное от её длины.
5. Пусть из n объектов только один обладает хорошим свойством. Смоделируем выбор с возвращением: на каждом шаге выбирается случайный равновероятный объект. Если он не обладает хорошим свойством, предмет кладётся обратно в кучу. В противном случае игра останавливается. Определите математическое ожидание числа шагов до конца игры.
6. Поиск минимального вершинного покрытия с доказательством корректности.
7. Теоремы Карзанова. Леммы для первой теоремы (б/д), вывод из них самой теоремы. Вторая теорема -- б/д. Единичные сети и алгоритм Хопкрофта--Карпа.
8. Поиск окружности минимального радиуса, покрывающей данное множество точек.
9. Зет-функция: определение, линейный алгоритм построения и доказательство корректности. Поиск всех вхождений шаблона s в текст t .
10. Алгоритм Ахо--Корасик: поиск всех вхождений шаблона s (в котором допускается не более k вхождений) в текст t .

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

1 семестр:

1. Динамический массив.
2. Амортизационный анализ.
3. Учетная оценка времени добавления элемента в динамический массив (с удвоением заполненного буфера).
4. Связные списки.
5. Стек, очередь, дек и их реализации.
6. Быстрая сортировка (QuickSort).
7. Поиск порядковой статистики методом “Разделяй и властвуй” (QuickSelect).
8. Сортировка слиянием (MergeSort).
9. Поразрядные сортировки.
10. Двоичная куча и сортировка кучей (HeapSort).
11. Слияние k отсортированных массивов с помощью кучи.
12. Хеш-таблица, полиномиальная хеш-функция.
13. Динамическое программирование: общая идея, линейная динамика, матричная, динамика на отрезках, на масках.
14. RMQ. Sparse table. Дерево отрезков.
15. LCA: сведение к RMQ и метод двоичного подъёма.
16. Двоичное дерево поиска.

Примеры билетов 1 семестр:

Билет №1:

1. Сортировка слиянием (MergeSort).
2. Хеш-таблица, полиномиальная хеш-функция.

Билет №2:

1. Двоичная куча и сортировка кучей (HeapSort).
2. RMQ. Sparse table. Дерево отрезков.

2 семестр:

1. Обходы в глубину и в ширину.
2. Поиск ключа, наивные вставка и удаление ключа.
3. АВЛ-дерево.
4. Красно-чёрное дерево.
5. Определение, описание алгоритма insert.
6. Декартово дерево.
7. Декартово дерево по неявному ключу.
8. Минимальное остовное дерево: алгоритмы Прима и Крускала.
9. Максимальные потоки в сети.
10. Методы: Форда-Фалкерсона; Эдмондса-Карпа (б/д).
11. Обход графа в глубину, ширину.
12. Поиск кратчайших путей в графе: алгоритмы Дейкстры, Форда-Беллмана, Флойда-Уоршелла.
13. Поиск сильно-связных компонент в графе. Задача 2SAT.
14. Мосты и точки сочленения в графе.
15. Нахождение подстроки в строке: префикс-функция, алгоритм Кнута-Морриса-Пратта.

Примеры билетов 2 семестр:

Билет №1:

1. Максимальные потоки в сети.
2. Поиск сильно-связных компонент в графе. Задача 2SAT.

Билет №2:

1. Обход графа в глубину, ширину.
2. Поиск кратчайших путей в графе: алгоритмы Дейкстры, Форда-Беллмана, Флойда-Уоршелла.

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 экзамена обучающемуся предоставляется 30 минут на подготовку. Опрос обучающегося по билету не должен превышать двух астрономических часов.

Во время проведения экзамена обучающиеся могут пользоваться программой дисциплины.