

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

УТВЕРЖДЕНО

**Директор высшей школы
программной инженерии
А.В. Малеев**

| | |
|----------------------------|---|
| | Рабочая программа дисциплины (модуля) |
| по дисциплине: | Алгоритмы и структуры данных |
| по направлению: | Программная инженерия |
| профиль подготовки: | Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс |
| курс: | 1 |
| квалификация: | бакалавр |

Семестры, формы промежуточной аттестации:

2 (весенний) - Экзамен

3 (осенний) - Экзамен

Аудиторных часов: 120 всего, в том числе:

лекции: 60 час.

семинары: 0 час.

лабораторные занятия: 60 час.

Самостоятельная работа: 108 час.

Подготовка к экзамену: 60 час.

Всего часов: 288, всего зач. ед.: 8

Количество контрольных работ, заданий: 6

Программу составил: И.Д. Степанов, ассистент

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 28.04.2023

Аннотация

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

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

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

1. Цели и задачи

Цель дисциплины

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

Задачи дисциплины

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

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

| Код и наименование компетенции | Индикаторы достижения компетенции |
|---|--|
| ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов | ОПК-6.1 Знает алгоритмы решения типовых задач, области и способы их применения |
| | ОПК-6.2 Умеет применять языки программирования для решения прикладных задач |
| ПК-2 Способен формализовать и алгоритмизировать поставленную задачу | ПК-2.1 Способен формализовать и алгоритмизировать поставленную задачу |
| | ПК-2.2 Владеет методами и приемами формализации и алгоритмизации задач |

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. работа |
| 1 | Асимптотики, мастер-теорема | 4 | | 4 | 7 |
| 2 | Линейные структуры данных | 4 | | 4 | 8 |
| 3 | Сортировки и порядковые статистики | 4 | | 4 | 8 |
| 4 | Кучи | 4 | | 4 | 8 |
| 5 | Деревья поиска | 4 | | 4 | 7 |
| 6 | Дерево отрезков, дерево Фенвика | 4 | | 4 | 8 |
| 7 | Хэш-таблицы, фильтры Блума | 6 | | 6 | 8 |
| 8 | Паросочетания, алгоритм Куна | 4 | | 4 | 8 |
| 9 | Максимальные потоки в сетях | 6 | | 6 | 8 |
| 10 | Простейшие строковые алгоритмы | 4 | | 4 | 8 |
| 11 | Строковые суффиксные структуры | 4 | | 4 | 8 |
| 12 | Геометрические примитивы | 4 | | 4 | 7 |
| 13 | Выпуклая оболочка | 4 | | 4 | 7 |
| 14 | Рандомизированные алгоритмы и комбинаторные игры | 4 | | 4 | 8 |
| Итого часов | | 60 | | 60 | 108 |
| Подготовка к экзамену | | 60 час. | | | |
| Общая трудоёмкость | | 288 час., 8 зач.ед. | | | |

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

Семестр: 2 (Весенний)

1. Асимптотики, мастер-теорема

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

2. Линейные структуры данных

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

3. Сортировки и порядковые статистики

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

4. Кучи

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

5. Деревья поиска

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

6. Дерево отрезков, дерево Фенвика

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

7. Хэш-таблицы, фильтры Блума

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

Семестр: 3 (Осенний)

8. Паросочетания, алгоритм Куна

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

9. Максимальные потоки в сетях

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

10. Простейшие строковые алгоритмы

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

11. Строковые суффиксные структуры

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

12. Геометрические примитивы

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

13. Выпуклая оболочка

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

14. Рандомизированные алгоритмы и комбинаторные игры

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

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

учебная аудитория с доской.

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

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-кон테стов 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++.

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

Программа курса в разделе «самостоятельная работа» обозначает минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- проработку учебного материала (по конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- выполнение домашних заданий.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

| | |
|----------------------------|---|
| по направлению: | Программная инженерия |
| профиль подготовки: | Разработка программно-информационных систем высшая школа программной инженерии МФТИ - Яндекс высшая школа программной инженерии |
| курс: | 1 |
| квалификация: | бакалавр |

Семестры, формы промежуточной аттестации:

2 (весенний) - Экзамен

3 (осенний) - Экзамен

| | |
|---------------------|--------------------------|
| Разработчик: | И.Д. Степанов, ассистент |
|---------------------|--------------------------|

1. Компетенции, формируемые в процессе изучения дисциплины

| Код и наименование компетенции | Индикаторы достижения компетенции |
|---|--|
| ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов | ОПК-6.1 Знает алгоритмы решения типовых задач, области и способы их применения |
| | ОПК-6.2 Умеет применять языки программирования для решения прикладных задач |
| ПК-2 Способен формализовать и алгоритмизировать поставленную задачу | ПК-2.1 Способен формализовать и алгоритмизировать поставленную задачу |
| | ПК-2.2 Владеет методами и приемами формализации и алгоритмизации задач |

2. Показатели оценивания компетенций

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

знать:

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

уметь:

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

владеть:

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

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

Для 2 семестра

1. Пусть $c > 0$ -- константа. Докажите, что решение рекурренты $T(n) = T(n/3) + T(2n/3) + cn$ ведёт себя как $\Omega(n \log n)$.
2. Разработайте стек, который умеет прибавлять ко всем хранящимся значениям произвольную поправку x за $O(1)$. Иными словами, нужно реализовать операцию увеличения всех чисел в стеке на x .
3. В $2n - 1$ ящиках лежат яблоки и апельсины. Требуется выбрать n ящиков так, что в них окажется не менее половины всех яблок и не менее половины всех апельсинов. Докажите, что такой выбор всегда существует.
4. Двоичная куча с минимумом в корне на n элементах расположена в памяти в виде массива. По числу k найдите минимальные k элементов в куче за $O(k \log k)$.

Для 3 семестра

1. Пусть G -- ациклический ориентированный граф. Предложите алгоритм поиска наименьшего количества вершинно непересекающихся путей, которые покрывают все вершины G . Асимптотика: $O(nm)$, где n -- число вершин в G , а m -- число его рёбер.

2. Модифицируйте алгоритм Куна так, чтобы его время работы составляло $O(am)$, где a -- размер максимального паросочетания, а m -- число рёбер графа.
3. Докажите теорему о декомпозиции потока: если в сети G течёт некий поток f , то его можно разбить на несколько путей из s в t , а также на несколько (замкнутых) циклов.
4. Как реализовать алгоритм Ахо--Корасик на динамически расширяющемся множестве строк? Используйте идею разложения n по степеням двойки.
5. Предложите алгоритм нахождения пересечения двух выпуклых многоугольников за $O(nm)$, если n -- это количество сторон в первом из них, а m -- во втором.

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

1. По данному числу n найдите все пары целых положительных чисел (a, b) , такие что $a \leq b \leq n$, и $a \mid b$. Оцените асимптотическое поведение числа таких пар.
2. Число 0 записано в n -разрядной двоичной системе. К нему $2^n - 1$ раз прибавляется единица. Будем считать, что время, необходимое на прибавление единицы, равно количеству единиц в двоичной записи числа, которые становятся нулями. Оцените среднюю сложность всех таких операций. Какие операции являются самыми дешёвыми, а какие -- самыми дорогими?
3. Предложите метод хранения минимального значения в очереди с помощью структуры deque. Если к очереди поступило n запросов, время работы программы должно составлять $O(n)$.
4. Напомним, что процедура $\text{Partition}(A, x)$ переупорядочивает элементы массива A так, что сначала идут все элементы, не превосходящие x , в некотором порядке, а затем -- все элементы, большие x . Покажите, как реализовать $\text{Partition}(A, x)$ с привлечением $O(1)$ дополнительной памяти.
5. 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?
6. 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)?
7. Задан полный взвешенный двудольный граф, в обеих долях которого находится по n вершин. Предложите алгоритм поиска самого дешёвого совершенного паросочетания.
8. С помощью суффиксного автомата найдите количество различных подстрок строки s за время, линейное от её длины.
9. 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.
10. Поток минимальной стоимости. Корректность алгоритма, асимптотика. Применение алгоритмов Форда-Беллмана и Дейкстры (с потенциалами).

Примеры экзаменационных билетов:

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

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

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

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

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

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