

**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>      | Алгоритмы и структуры данных. Продвинутый поток  |
| <b>по направлению:</b>     | Прикладная математика и информатика  |
| <b>профиль подготовки:</b> | Математика   |
|                            | Физтех-школа Прикладной Математики и Информатики<br>кафедра алгоритмов и технологий программирования |
| <b>курс:</b>               | 1  |
| <b>квалификация:</b>       | 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 12.02.2024

## 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-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   | ОПК-2.2 Знает и умеет применять численные математические методы и прикладное программное обеспечение для решения научных задач в профессиональной области |

|   |   |
|---|---|
| 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-2.3 Знает основные требования информационной безопасности   |

### 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        | 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

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.

## Семестр: 2 (Весенний)

### 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.

### 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).

## 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)

Не используются

## 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)

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>     | Прикладная математика и информатика  |
| <b>профиль подготовки:</b> | Математика<br>Физтех-школа Прикладной Математики и Информатики<br>кафедра алгоритмов и технологий программирования |
| <b>курс:</b>               | <u>1</u>   |
| <b>квалификация:</b>       | 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   |

## 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  $\text{Partition}(A, x)$  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. Нижняя оценка на число сравнений в сортировке, основанной на сравнениях.
6. Лемма о корректности  $\text{siftDown}$  и  $\text{siftUp}$ .
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  $\text{decreaseKey}$ . 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  $k$  вопросов) 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. AVL-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.