

**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:	Введение в программирование и алгоритмы
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс
курс:	1
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 1 (осенний) - Экзамен

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 0 час.

лабораторные занятия: 30 час.

Самостоятельная работа: 54 час.

Подготовка к экзамену: 30 час.

Всего часов: 144, всего зач. ед.: 4

Количество контрольных работ, заданий: 2

Программу составил: И.Д. Степанов, ассистент

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 17.06.2021

Аннотация

Курс представляет собой классическую часть большого курса “Алгоритмы и структуры данных”. 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 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

- 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;
- 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 естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
	ОПК-5.3 Владеет навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем

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.

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;
- 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	10
2	Линейные структуры данных	4		4	8
3	Сортировки и порядковые статистики	6		6	11
4	Кучи	4		4	6
5	Деревья поиска	4		4	5
6	Дерево отрезков, дерево Фенвика	4		4	6
7	Хэш-таблицы, фильтры Блума	4		4	8
Итого часов		30		30	54
Подготовка к экзамену		30 час.			
Общая трудоёмкость		144 час., 4 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.

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.

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

Учебная аудитория с доской и проектором

6. Перечень рекомендуемой литературы

Основная литература

1. Алгоритмы: построение и анализ [Текст] : [учебник для вузов] / Т. Кормен [и др.] ; [пер. с англ. И. В. Красикова и др.] .— 3-е изд. — М. : Вильямс, 2014 .— 1328 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

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
Семестр, формы промежуточной аттестации: 1 (осенний) - Экзамен	
Разработчик:	И.Д. Степанов, ассистент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ОПК-1.1 Способен анализировать поставленную задачу, намечать пути ее решения
	ОПК-1.2 Способен строить математические модели, производить количественные расчеты и оценки
	ОПК-1.3 Способен определять границы применимости полученных результатов
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1 Владеет на практике методологией составления научно-технических отчетов (проектов)
	ОПК-5.2 Способен оценивать концепции и атрибуты качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роль людей, процессов, методов, инструментов и технологий обеспечения качества
	ОПК-5.3 Владеет навыками инсталляции программного и аппаратного обеспечения информационных и автоматизированных систем

2. Показатели оценивания компетенций

В результате изучения дисциплины «Введение в программирование и алгоритмы» 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.

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;
- приемами сведения общих задач к более конкретным и простым.

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

1. Пусть $c > 0$ -- константа. Докажите, что решение рекуррентности $T(n) = T(n/3) + T(2n/3) + cn$ 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 $O(k \log k)$.

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

Вопросы:

1. Нижняя оценка на число сравнений в сортировке, основанной на сравнениях.
2. Удаление в куче (по указателю и по значению).
3. Фибоначчиева куча: операций decreaseKey. Асимптотика и корректность.
4. 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.
5. 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.
6. Предложите метод хранения минимального значения в очереди с помощью структуры deque. Если к очереди поступило n запросов, время работы программы должно составлять $O(n)$.
7. 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.
8. Лемма о корректности siftDown и siftUp.
9. Двумерное дерево отрезков снизу: прибавление в точке и сумма в прямоугольнике.
10. Фильтры Блума: асимптотика и корректность.

Билет №1

1. Фибоначчиева куча: операций decreaseKey. Асимптотика и корректность.
2. Лемма о корректности siftDown и siftUp.

Билет №2

1. Предложите метод хранения минимального значения в очереди с помощью структуры deque. Если к очереди поступило n запросов, время работы программы должно составлять $O(n)$.
2. Нижняя оценка на число сравнений в сортировке, основанной на сравнениях.

Билет №3

1. Двумерное дерево отрезков снизу: прибавление в точке и сумма в прямоугольнике.
2. Удаление в куче (по указателю и по значению).

Критерии оценивания

Оценка Баллы Критерии

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.