

**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:	Практикум по алгоритмам и структурам данных
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
курс:	1
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

2 (весенний) - Дифференцированный зачет

3 (осенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 0 час.

семинары: 60 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 84 час.

Всего часов: 144, всего зач. ед.: 4

Программу составил: А.В. Созыкин, канд. техн. наук, доцент

Программа обсуждена на заседании высшей школы программной инженерии 28.04.2023

Аннотация

Практикум по алгоритмам и структурам данных дополняет курс по алгоритмам и структурам данных, и предназначен для овладения практическими навыками реализации структур данных на языке C++ и алгоритмов для работы с ними.

В рамках прохождения практикума будут изучены реализации алгоритмов в области поиска максимального потока в транспортных сетях, базовых и продвинутых алгоритмов на строках, и алгоритмов вычислительной геометрии.

1. Цели и задачи

Цель дисциплины

- дисциплина направлена на формирование, закрепление и развитие практических навыков и компетенций по профилю образовательной программы. Практикум проводится с целью дать теоретические и практические знания об алгоритмах и структурах данных, теории графов, с доказательством корректности их работы, о методах оценки сложности алгоритмов.

Задачи дисциплины

Перед студентами ставятся следующие задачи:

- овладение навыками выбора подходящих алгоритмов для решения прикладных задач
- овладение навыками оптимизации вычислительной сложности при решении прикладных задач
- овладение навыками тестирования и доказательства корректности работы программ
- овладение навыками реализации алгоритмов 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-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.1 Знает алгоритмы решения типовых задач, области и способы их применения
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:

- алгоритмы на графах и структуры данных, связанные с ними;
- оценки сложности стандартных алгоритмов;
- методы теоретического и экспериментального исследования в области, соответствующей профилю образовательной программы;
- принципы организации проведения экспериментов и испытаний.

уметь:

- провести обзор имеющегося материала для решения поставленной задачи;
- использовать выбранный метод или сочетать различные методы в решении поставленной задачи;
- применять современные методы сбора и обработки данных при прохождении практики;
- строить деятельность на основе выполнения технологических требований и нормативов, придерживаться правовых и этических норм, принятых в профессиональной деятельности;
- реализовывать стандартные алгоритмы на графах и структуры данных на языке программирования C++;
- оформлять и представлять результаты выполненной работы.

владеть:

- навыками анализа научной и технической информации в области, соответствующей профилю образовательной программы;
- методами декомпозиции задач в области информационных технологий и построения единого решения с использованием изученных алгоритмов;
- 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	Подготовительный этап		7		10
2	Обзор и анализ решения типовых задач		7		10
3	Решение задач, защита решений и подготовка отчета		8		11
4	Решение контрольной задачи и анализ результата		8		11
5	Решение задач, защита решений и подготовка отчета		15		21
6	Решение контрольной задачи и анализ результата		15		21
Итого часов			60		84
Подготовка к экзамену		0 час.			
Общая трудоёмкость		144 час., 4 зач.ед.			

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

Семестр: 2 (Весенний)

1. Подготовительный этап

Постановка учебных задач. Ознакомление с правилами сдачи работы и критериями оценивания. Вводные занятия по используемым технологиям.

2. Обзор и анализ решения типовых задач

Разбор типовых задач по темам:

- линейные структуры данных;
- сортировки и порядковые статистики;
- кучи;
- деревья поиска.

Решение типовых задач.

3. Решение задач, защита решений и подготовка отчета

Решение задач и их устная защита.

Итеративный процесс улучшения решений, которые засчитываются руководителем практики только после успешного прохождения процедуры рецензирования и устранения всех замечаний.

Подготовка итогового отчета о выполненных работах.

4. Решение контрольной задачи и анализ результата

Прохождение рубежного контроля в виде решения типовых задач за ограниченное время.

Семестр: 3 (Осенний)

5. Решение задач, защита решений и подготовка отчета

Решение задач и их устная защита.

Итеративный процесс улучшения решений, которые засчитываются руководителем практики только после успешного прохождения процедуры рецензирования и устранения всех замечаний.

Подготовка итогового отчета о выполненных работах.

6. Решение контрольной задачи и анализ результата

Прохождение рубежного контроля в виде решения типовых задач за ограниченное время.

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

Для проведения дисциплины необходимы: рабочее место для самостоятельной работы, содержащее персональный компьютер, с доступом к сети Интернет и электронной образовательной среде МФТИ, учебные аудитории для проведения групповых и индивидуальных консультаций, проведения промежуточной аттестации (по месту прохождения практики).

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 предусмотрено

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

Программное обеспечение: набор компиляторов GNU GCC, либо среда Microsoft Visual Studio.

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

В преподавании дисциплины сочетаются практические занятия в учебном классе и домашние задания, состоящие из теоретических задач.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии
курс:	<u>1</u>
квалификация:	бакалавр

Семестры, формы промежуточной аттестации:

- 2 (весенний) - Дифференцированный зачет
- 3 (осенний) - Дифференцированный зачет

Разработчик: А.В. Созыкин, канд. техн. наук, доцент

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-6.2 Умеет применять языки программирования для решения прикладных задач
	ОПК-6.1 Знает алгоритмы решения типовых задач, области и способы их применения
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 «Практикум по алгоритмам и структурам данных» обучающийся должен:

знать:

- алгоритмы на графах и структуры данных, связанные с ними;
- оценки сложности стандартных алгоритмов;
- методы теоретического и экспериментального исследования в области, соответствующей профилю образовательной программы;
- принципы организации проведения экспериментов и испытаний.

уметь:

- провести обзор имеющегося материала для решения поставленной задачи;
- использовать выбранный метод или сочетать различные методы в решении поставленной задачи;
- применять современные методы сбора и обработки данных при прохождении практики;
- строить деятельность на основе выполнения технологических требований и нормативов, придерживаться правовых и этических норм, принятых в профессиональной деятельности;
- реализовывать стандартные алгоритмы на графах и структуры данных на языке программирования C++;
- оформлять и представлять результаты выполненной работы.

владеть:

- навыками анализа научной и технической информации в области, соответствующей профилю образовательной программы;
- методами декомпозиции задач в области информационных технологий и построения единого решения с использованием изученных алгоритмов;
- приёмами сведения общих задач к более конкретным и простым.

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

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлого занятия.

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

Вопросы для 2го семестра

1. Написать программу поиска префикс-функции по заданной Z-функции.
2. Реализовать поиск в двух неубывающих последовательностях по одному элементу с минимальным модулем их разности.
3. Реализовать структуру данных "дек" (deque).

4. Реализовать структуру данных "куча" и реализовать поиск минимального и минимального значения в ней.
5. Реализовать поиск минимума для каждой заданной подматрицы в исходной матрице.
6. Реализовать аналог класса `std::string`.
7. Реализовать классы для работы с длинной арифметикой и рациональными числами.
8. Реализовать умножение двух длинных чисел.
9. Решить задачу о поиске троек из единичек в строке, состоящей из значений 0 и 1.
10. Реализовать перевод чисел длины 100000 из одной системы счисления в другую.
11. Реализовать деление многочленов с остатком.
12. Реализовать проверку, является ли корень числа длины 1000000 целым.
13. Реализовать иерархию классов для работы с геометрическими фигурами на плоскости.
14. Изучить алгоритм `radix` и реализовать перевод числа длины 1000 из 10-чной системы в бинарную.
15. Реализовать шаблонный класс `residue` (остаток).
16. Реализовать класс для работы с матрицами.
17. Реализовать с помощью шаблонов структуру `deque`.
18. Реализовать программу для проверки, сколько из заданных точек лежат внутри выпуклого многоугольника.
19. Реализовать построение выпуклой оболочки множества точек.
20. Написать программу вычисления площади объединения двух треугольников.
21. Реализовать построение суффиксного дерева.
22. Решить задачу определения принадлежности точек заданным отрезкам.
23. Реализовать алгоритм поиска подстрок с максимальным произведением длины на число вхождений.

Вопросы для 3го семестра

1. Реализовать алгоритм поиска подстроки в строке.
2. Реализовать алгоритм поиска суммы вхождений строк в бор.
3. Реализовать префикс-функции.
4. Реализовать Z-функцию.
5. Реализовать алгоритм поиска количества повторов пар букв в s-строке.
6. Решить задачу о выборе футболистов с максимальной суммарной эффективностью.
7. Решить задачу о поиске общего числа обготов при старте с интервалом в 1 минуту и с разной скоростью.
8. Реализовать алгоритм Блюма.
9. Доказать эффективность алгоритма `timsort`.
10. Решить задачу об оптимальном совмещении ДНК.
11. Реализовать выполнение двух быстрых преобразований Фурье.
12. Реализовать программу для поиска оптимального сопоставления двух двумерных таблиц.
13. Доказать теорему о регулярных графах.
14. Написать программу распределения заказов по минимальному числу машин.
15. Реализовать алгоритм поиска k-ых общих подстрок для двух строк.

Критерии оценивания

отлично

- 10 Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач. Решения задач оформлены в едином удобочитаемом стиле.
- 9 Полностью и вовремя решены все задачи без ошибок. Продемонстрирован грамотный подход к решению задач
- 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.