

**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:	Построение и анализ алгоритмов в программировании
по направлению:	Информатика и вычислительная техника
профиль подготовки:	Физтех-школа Прикладной Математики и Информатики кафедра когнитивных технологий
курс:	3
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет

Аудиторных часов: 30 всего, в том числе:

лекции: 15 час.

семинары: 15 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 15 час.

Всего часов: 45, всего зач. ед.: 1

Количество контрольных работ, заданий: 1

Программу составил: В.Л. Арлазаров, д-р физ.-мат. наук, (на удаление) член-корреспондент
государственной академии наук

Программа обсуждена на заседании кафедры когнитивных технологий 04.03.2020

Аннотация

В рамках дисциплины слушатели знакомятся с классическими алгоритмами, учатся применять алгоритмы и типовые структуры данных, а также вырабатывать на их основе структуры, адекватные решаемым задачам. Целью курса является выработка у студентов понимания важности постоянного внимания к эффективности алгоритмов, используемых при программировании, познакомить их с методами, которые могут использоваться для достижения эффективности. Студенты овладевают методами оценки эффективности используемых решений.

1. Цели и задачи

Цель дисциплины

Выработать у студентов понимание важности постоянного внимания к эффективности алгоритмов, используемых при программировании и познакомить их с методами, которые могут использоваться для достижения эффективности.

Задачи дисциплины

- Овладение студентами методами оценки эффективности, овладение типовыми алгоритмами, используемыми в задачах обработки информации, выработка умения применять типовые структуры данных и вырабатывать на их основе структуры, адекватные решаемым задачам.
- Подготовка к участию в научных семинарах, научно-технических конференциях и симпозиумах, составлению научных обзоров, рефератов и библиографии по тематике исследований.
- Подготовка к оказанию консалтинговых услуг по данной тематике.
- Подготовка к участию в международных проектах по тематике дисциплины.
- Подготовка к участию в разработке корпоративной политики и мероприятиях в области повышения социальной ответственности бизнеса перед обществом, включая разработку и реализацию решений, направленных на поддержку социально-значимых проектов.
- Совершенствование и расширение общенаучной базы. Повышение уровня общекультурного и нравственного совершенствования своей личности.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны

знать:

Фундаментальные концепции и профессиональные результаты, системные методологии в профессиональной области; современное состояние и принципиальные возможности языков и систем программирования, основные методы быстрой сортировки, методы организации динамически изменяемых справочных систем, методы решения оптимизационных задач на графах.

уметь:

Использовать новые знания и применять их в профессиональной деятельности; использовать современные теории, методы, системы и средства прикладной математики и информационных технологий для решения научно-исследовательских и прикладных задач, профессиональной области; современное состояние и принципиальные возможности языков и систем программирования, основные методы быстрой сортировки, методы организации динамически изменяемых справочных систем, методы решения оптимизационных задач на графах.

владеть:

Основами методологии научного познания и системного подхода при изучении различных уровней организации материи, информации, пространства и времени.

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	Сортировки.	1	1		
2	Типовые структуры данных.	2	1		
3	Методы поддержания справочных данных.	2	1		
4	Динамическое программирование	2	1		
5	Жадные алгоритмы.	2	1		
6	Алгоритмы на графах.	2	4		
7	Методы перебора.	2	3		
8	Теоретико-числовые алгоритмы.	2	3		15
Итого часов		15	15		15
Подготовка к экзамену		0 час.			
Общая трудоёмкость		45 час., 1 зач.ед.			

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

Семестр: 6 (Весенний)

1. Сортировки.

Сортировка пирамидой (Определение пирамиды, «почти пирамиды», лемма о преобразовании почти пирамиды в пирамиду. Начальное построение пирамиды. Преобразование пирамиды в упорядоченный массив. Оценки).

Сортировка слиянием (Слияние упорядоченных массивов. Рекурсивная программа слияния. Оценки быстродействия и памяти).

Сортировки за линейное время (Сортировка подсчётом. Цифровая сортировка. Сортировка «вычерпыванием»).

2. Типовые структуры данных.

Элементарные структуры данных (стеки, очереди. Очереди с приоритетами, односторонние и двухсторонние списки. Связанные списки. Поиск в списке.)

Деревья. Определение дерева. Степени ветвления. Реализация с помощью списков или «гнездовых» структур. Обходы дерева. Частично упорядоченные множества. Двоичные деревья поиска.

3. Методы поддержания справочных данных.

Хэш таблицы. (прямая адресация. Организация хэш таблицы. Функции расстановки. Алгоритмы разрешения коллизий – списки, использование памяти таблицы. Открытая адресация. Организация хэш таблиц на внешней памяти).

АВЛ деревья. (Определение сбалансированного дерева. Оценки глубины. Добавление нового элемента и балансировка дерева).

Б-деревья. (Определение Б-дерева. Операция добавления. Деления блоков. Удаление элемента. Уплотнение Б-дерева).

4. Динамическое программирование

Принцип оптимальности. (определение, условие монотонности оценок. Прямой и обратный ход. Алгоритм Витерби).

Монотонность и порядок решения подзадач. (Оценки и их ограничения, запоминания решений, «отсечения»).

Типовые размеры. (Задача о расстановке скобок при перемножении матриц. Разбиение последовательности на фрагменты. Поиск «почти оптимальных» решений).

5. Жадные алгоритмы.

Локально-оптимальный выбор. (принцип поиска оптимальной подзадачи. Непротиворечивость оптимальному решению. Ограничения локально-оптимального выбора).

Примеры задач, решаемых жадными алгоритмами. (задача о выборе заявок, коды Хаффмана, расписание с равной длительностью).

Матроиды. (Определение. Взвешенный матроид. Матричный матроид. Достаточные условия применимости жадного алгоритма).

6. Алгоритмы на графах.

Представление графа. (Матрицы смежности. Списки рёбер и их порядок. Нагруженные рёбра. Пути в графе и их идентификация. Сильно связанные компоненты).

Алгоритмы обхода графа. (Обход в ширину, обход в глубину, топологическая сортировка).

Поиск путей в графе. (Поиск кратчайшего расстояния между вершинами. Алгоритм Дейкстры. Алгоритм Беллмана).

Задача о максимальном потоке. (Потоки в сетях. Алгоритм Флойда-Уоршола. Алгоритм Джонсона).

7. Методы перебора.

Порождение подмножеств. (Все подмножества. Подмножества с заданными свойствами. Рекурсивный выбор подмножеств).

обход дерева перебора. (Методы фиксации состояния. Оценки состояния. Метод ветвей и границ).

Деревья игры. (Дерево с оценками. Теорема Цермело. Перенос оценок. Метод граней и оценок Брудно).

8. Теоретико-числовые алгоритмы.

Элементы теории чисел. (Делимость. Сравнение по модулю. Теорема о наибольшем общем делителе. Единственность разложения на простые множители).

Китайская теорема об остатках. (Алгоритм Эвклида. Оценка быстродействия. Решение линейных диофантовых уравнений).

криптосистема RSA с открытыми и закрытыми ключами. (Теорема Эйлера. Малая теорема Ферма. Построение открытого и закрытого ключа. Поиск простых чисел).

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

Учебная аудитория, оснащенная медиапроектором и экраном.

6.Перечень рекомендуемой литературы

Основная литература

1. Алгоритмы : построение и анализ [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест .— М. : МЦНМО, 2001 .— 960 с.
2. Алгоритмы: вводный курс [Текст] /Томас Х. Кормен; пер. с англ. %dAlgorithms Unlocked. М., ООО "И. Д. Вильямс", 2017

Дополнительная литература

1. Искусство программирования для ЭВМ [Текст] : [в 7 т.]. Т. 1 : Основные алгоритмы : [учеб. пособие для вузов] / Д. Кнут ; пер. с англ. Г. П. Бабенко, Ю. М. Баяковского ; под ред. К. И. Бабенко, В. С. Штаркмана .— М. : Мир, 1976 .— 735 с.
2. Искусство программирования для ЭВМ [Текст] : в 7 т. Т. 2 : Получисленные алгоритмы : [учеб. пособие для вузов] / Д. Кнут ; пер. с англ. Г. П. Бабенко [и др.] ; под ред. К. И. Бабенко, В. С. Штаркмана .— М. : Мир, 1977 .— 724 с.

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

<http://elibrary.ru/defaultx.asp> Научная электронная библиотека;
<http://www.twirpx.com> Все для студента

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

На занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

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

Во время проведения дифференцированного зачета обучающиеся могут пользоваться программой дисциплины, а также литературой из списка рекомендуемой литературы.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению: Информатика и вычислительная техника

профиль подготовки: Физтех-школа Прикладной Математики и Информатики
кафедра когнитивных технологий

курс: 3

квалификация: бакалавр

Семестр, формы промежуточной аттестации: 6 (весенний) - Дифференцированный зачет

Разработчик: В.Л. Арлазаров, д-р физ.-мат. наук, (на удаление) член-корреспондент
государственной академии наук

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.2 Находит, критически анализирует и выбирает информацию, необходимую для решения поставленной задачи
	УК-1.5 Определяет и оценивает практические последствия возможных вариантов решения задачи
ПК-1 Способен ставить, формализовывать и решать задачи, в том числе разрабатывать и исследовать математические модели изучаемых явлений и процессов, системно анализировать научные проблемы, получать новые научные результаты	ПК-1.1 Способен находить, анализировать и обобщать информацию об актуальных результатах исследований в рамках тематической области своей профессиональной деятельности

2. Показатели оценивания компетенций

В результате изучения дисциплины «Построение и анализ алгоритмов в программировании» обучающийся должен:

знать:

Фундаментальные концепции и профессиональные результаты, системные методологии в профессиональной области; современное состояние и принципиальные возможности языков и систем программирования, основные методы быстрой сортировки, методы организации динамически изменяемых справочных систем, методы решения оптимизационных задач на графах.

уметь:

Использовать новые знания и применять их в профессиональной деятельности; использовать современные теории, методы, системы и средства прикладной математики и информационных технологий для решения научно-исследовательских и прикладных задач, профессиональной области; современное состояние и принципиальные возможности языков и систем программирования, основные методы быстрой сортировки, методы организации динамически изменяемых справочных систем, методы решения оптимизационных задач на графах.

владеть:

Основами методологии научного познания и системного подхода при изучении различных уровней организации материи, информации, пространства и времени.

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

Перечень вопросов для промежуточного контроля:

1. Определение пирамиды. Начальное построение пирамиды.
2. Слияние упорядоченных массивов.
3. Очереди с приоритетами, односторонние и двухсторонние списки. Связанные списки. Поиск в списке.
4. Реализация с помощью списков или «гнездовых» структур.
5. Организация хэш таблицы. Организация хэш таблиц на внешней памяти.
6. Алгоритм Витерби.
7. Принцип поиска оптимальной подзадачи.
8. Примеры задач решаемых жадными алгоритмами.
9. Алгоритмы на графах. Представление графа.
10. Алгоритм Дейкстры. Алгоритм Беллмана.
11. Алгоритм Флойда-Уоршола. Алгоритм Джонсона.
12. Подмножества с заданными свойствами. Рекурсивный выбор подмножеств.
13. Элементы теории чисел.
14. Алгоритм Эвклида.

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

1. Дать определение пирамиды.
2. Какая связь между подчиненными элементами пирамиды и номерами элементов массива?
3. Что такое «почти пирамида» и как она преобразуется в пирамиду?
4. Как создать пирамиду из произвольного массива?
5. Как превратить пирамиду в упорядоченный массив?
6. Сколько действий требуется для слияния двух упорядоченных массивов из m и n элементов?
7. Алгоритм сортировки слиянием.
8. Каково минимальное число действий, гарантирующее упорядочивание массива путём сравнения элементов.
9. Что такое устойчивая сортировка?
10. Пусть элементы массива целые числа. Как упорядочить за время линейное относительно m и n ?
11. Алгоритм цифровой сортировки.
12. Что такое хэширование и хэш-таблица?
13. Функции расстановки, основанные на умножении и делении.
14. Метод разрешения коллизий в хэш-таблицах с помощью цепочек и оценка времени добавления и поиска.
15. Что такое открытая адресация?
16. Что такое сбалансированное (АВЛ) дерево? Какова зависимость глубины и количества элементов АВЛ дерева?
17. Два случая балансировки АВЛ дерева при добавлении элемента.
18. Задача о расстановке скобок при перемножении матриц.
19. Алгоритм разбиения последовательности на цепочки слов.
20. Метод динамического программирования. Запоминание подзадач.
21. Алгоритм Витерби.
22. Отсечение в динамическом программировании.
23. Задача о выборе заявок.
24. Коды Хоффмена.
25. Достаточные условия применимости жадного алгоритма.
26. Что такое матрица смежности ориентированного графа?
27. Что такое сильно связанные компоненты графа.
28. Алгоритм нахождения кратчайшего расстояния между вершинами.
29. Рекурсивный набор.
30. Дерево игры двух противников. Метод граней и оценок Брудно.
31. Что такое наибольший делитель? Алгоритм Эвклида.
32. Малая теорема Ферма.
33. Что такое криптографическая система RSA с открытыми и закрытыми ключами.
34. Китайская теорема об остатках.

Задание:

1. Решение уравнения в целых числах. Задача о максимальном потоке в сети.

Критерии оценивания

отлично (10) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

отлично (9) - выставляется студенту, показавшему свободное оперирование знаниями учебной программы дисциплины, выполнение заданий творческого характера.

отлично (8) - выставляется студенту, показавшему владение программным учебным материалом с наличием несущественных ошибок в действиях, самостоятельно исправляемых учащимся.

хорошо (7) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускается в ответе или в решении задач некоторые неточности.

хорошо (6) - выставляется студенту если он осознает воспроизведение программного учебного материала, в том числе и различной степени сложности, с несущественными ошибками, затруднения в применении отдельных навыков.

хорошо (5) - выставляется студенту если теоретическое содержание освоено не полностью, некоторые практические навыки сформированы недостаточно, в некоторых случаях были допущены ошибки.

удовлетворительно (4) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации.

удовлетворительно (3) - выставляется студенту в случае большого количества недочетов и неправильных ответов, а также пассивной работе в ходе занятий, многие учебные задания не выполнены.

неудовлетворительно (2) - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

неудовлетворительно (1) - выставляется студенту, который не освоил теоретическое и практическое содержание курса, все выполненные учебные задания содержат грубые ошибки.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированный зачет проводится по итогам текущей успеваемости и сдачи заданий путем организации специального опроса, проводимого в устной форме.