

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

УТВЕРЖДЕНО

**Проректор по учебной работе и
довузовской подготовке**

А.А. Воронов

	Рабочая программа дисциплины (модуля)
по дисциплине:	Алгоритмы
по направлению:	Прикладная математика и информатика
профиль подготовки:	Комбинаторика и цифровая экономика центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" кафедра дискретной математики
курс:	1
квалификация:	магистр

Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 75 час.

Всего часов: 135, всего зач. ед.: 3

Программу составил: А.Б. Дайняк, канд. физ.-мат. наук, доцент, доцент

Программа обсуждена на заседании кафедры дискретной математики 30.01.2025

Аннотация

Алгоритм - это пошаговая процедура, которая определяет набор инструкций, которые должны выполняться в определенном порядке для получения желаемого результата. Алгоритмы обычно создаются независимо от основных языков, то есть алгоритм может быть реализован на нескольких языках программирования. Курс изучает классические алгоритмы сортировки, доступа к данным для различных структур и сложности общих алгоритмов.

1. Цели и задачи

Цель дисциплины

- сформировать представление о разнообразных вычислительных задачах в теории графов и об асимптотических сложностях их решений;
- дать теоретические и практические знания об алгоритмах и структурах данных теории графов с доказательством корректности их работы, о методах оценки сложности алгоритмов.

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, их модификаций и комбинаций, в том числе с помощью амортизационного анализа, выбирать подходящие структуры данных для поставленных задач, реализовывать алгоритмы в обобщенной форме на языке программирования C++.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1 Анализирует проблемную ситуацию как систему, выявляя ее составляющие и связи между ними
	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
	УК-1.3 Разрабатывает стратегию достижения поставленной цели как последовательность шагов, предвидя результат каждого из них и оценивая их влияние на внешнее окружение планируемой деятельности и на взаимоотношения участников этой деятельности
ОПК-2 Способен совершенствовать и реализовывать новые математические методы решения прикладных задач	ОПК-2.1 Имеет представление о современном состоянии математических исследований в рамках тематической области своей профессиональной деятельности
	ОПК-2.2 Способен оценить актуальность и практическую значимость прикладных математических исследований в своей профессиональной области
	ОПК-2.3 Владеет профессиональной терминологией, используемой в современной научно-технической литературе, обладает навыками устного и письменного изложения результатов научной деятельности в рамках профессиональной коммуникации

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.
- Стандартные алгоритмы на графах и используемые структуры данных, подходы к модификации классических алгоритмов.
- Разнообразные классические задачи в теории графов и асимптотические сложности их решений.

уметь:

- формулировать задачи в терминах изученных теорий, выбирать подходящий алгоритм для поставленной задачи;
- разрабатывать комбинации алгоритмов для решения поставленной задачи,
- оценивать сложности алгоритмов, их модификаций и комбинаций, в том числе с помощью амортизационного анализа,
- выбирать подходящие структуры данных для конкретной задачи,
- реализовывать алгоритм в обобщенной форме на языке программирования с++;
- реализовывать стандартные алгоритмы на графах и структуры данных на языке программирования C++.

владеть:

- Методами декомпозиции задач в области информационных технологий и построения единого решения с использованием изученных алгоритмов.
- Методами оценки сложности алгоритмов, их модификаций и комбинаций.

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	Обходы Графа	6	6		15
2	Кратчайшие пути во взвешенном графе	6	6		15
3	Остовные деревья	6	6		15
4	Потоки в сетях	6	6		15
5	Структуры данных с операцией поиска на отрезке	6	6		15
Итого часов		30	30		75
Подготовка к экзамену		0 час.			
Общая трудоёмкость		135 час., 3 зач.ед.			

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

Семестр: 2 (Весенний)

1. Обходы Графа

- Ориентированный граф, псевдограф. Неориентированный граф, псевдограф.
 - Связность в неор. графе, компоненты связности.
- Слабая и сильная связность в ор. графе. Компоненты слабой, сильной связности.
- Обход в глубину. Цвета вершин. Времена входа и выхода. Лемма о белых путях.
 - Проверка связности неориентированного графа.

- Поиск цикла в неориентированном и ориентированном графе.
- Топологическая сортировка.
- Нахождение компонент сильной связности. Алгоритм Косарайю. Алгоритм Тарьяна.
- Компоненты реберной двусвязности. Мосты. Поиск мостов.
- Компоненты вершинной двусвязности. Точки сочленения. Поиск точек сочленения.
- Волновой алгоритм. Обход в ширину (применение очереди в волновом алгоритме).
- Критерий существования Эйлера пути и цикла в ориентированном и неориентированном графе. Поиск эйлера пути и цикла.

2. Кратчайшие пути во взвешенном графе

- Алгоритм Дейкстры.
- Цвета вершин. Дерево кратчайших путей.
- Потенциалы. Условие применимости алгоритма Дейкстры для измененных длин ребер. Потенциал $\pi(v) = \rho(v, t)$.
- Алгоритм A*. Условие монотонности на эвристику. Примеры эвристик.
- Двусторонний алгоритм Дейкстры.
- Алгоритм Форда-Беллмана.
- Хранение в матрице: Dvk равно длине кратчайшего пути до вершины v за ровно k ребер (не более k ребер). Доказательство корректности. Оценка времени работы.
- Восстановление пути.
- Детектирование цикла отрицательного веса. Поиск самого цикла.
- Нахождение кратчайших путей с учетом циклов отрицательного веса.
- Алгоритм Флойда. Доказательство. Восстановление пути.
- Нахождение цикла отрицательного веса.
- Алгоритм Джонсона. Добавление фиктивного корня и фиктивных ребер для запуска алгоритма Форда-Беллмана.

3. Остовные деревья

- Остовное дерево. Построение с помощью обхода в глубину и в ширину.
- Определение минимального остовного дерева.
- Теорема о разрезе. Доказательство.
- Алгоритм Прима. Аналогия с алгоритмом Дейкстры.
- Доказательство с помощью теоремы о разрезе. Оценка времени работы для различных реализаций очереди с приоритетом: бинарная куча, Фибоначчиева куча (последнее без доказательства).
- Алгоритм Крускала. Доказательство. Оценка времени работы.
- Система непересекающихся множеств. Эвристика потенциалов с доказательством оценки времени работы.
- Эвристика сжатия пути без доказательства.
- Алгоритм Борувки. Доказательство. Оценка времени работы.
- Приближение решения задачи коммивояжера с помощью минимального остовного дерева.

4. Поток в сетях

- Определение сети. Определение потока.
- Физический смысл. Аналогия с законами Кирхгофа.
- Определение разреза. Понятия потока через разрез.
- Доказательство факта, что поток через любой разрез одинаковый.
- Понятие остаточной сети. Понятие дополняющего пути.
- Необходимость отсутствия дополняющего пути для максимальности потока.
- Теорема Форда-Фалкерсона.
- Алгоритм Форда-Фалкерсона. Поиск минимального разреза.
- Пример целочисленной сети, в котором алгоритм работает долго.

- Алгоритм Эдмондса-Карпа.
- Доказательство, что кратчайшее расстояние в остаточной сети не уменьшается.
- Общая оценка времени работы алгоритма Эдмондса-Карпа.
- Слоистая сеть. Алгоритм Диница.
- Паросочетание. Максимальное паросочетание. Наибольшее паросочетание. Совершенное паросочетание.
- Задача поиска наибольшего паросочетания. Примеры реальных задач.
- Чередующий путь. Лемма Берга.
- Наибольшее паросочетание в двудольном графе. Оценка.

5. Структуры данных с операцией поиска на отрезке

- RSQ и RMQ.
- Sparse-table.
- Дерево отрезков.
- Обработка запросов от листьев.
- Обработка запросов от корня.
- Изменение значения в массиве, обновление дерева отрезков.
- Множественные операции.
- Дерево Фенвика.
- LCA. Метод двоичного подъема.
- Сведение LCA к задаче RMQ.
- Сведение RMQ к задаче LCA.
- Декартово дерево по неявному ключу.
- Множественные операции в декартовом дереве по неявному ключу.

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

- проектор с возможностью подключения через HDMI и / или VGA);
- доска с мелом или доска с фломастерами;
- компьютерный класс оборудован ПК.

6.Перечень рекомендуемой литературы

Основная литература

1. Функциональное программирование [Текст] : [учеб. пособие для вузов] / А. Филд, П. Харрисон ; пер. с англ. М. В. Горбатовой [и др.] ; под ред. В. А. Горбатова .— М. : Мир, 1993 .— 638 с.
2. Структуры данных и проектирование программ [Текст] / Р. Круз ; пер. 3-го англ. изд. К. Г. Финогенова - М.БИНОМ. Лаб. знаний,2008

Дополнительная литература

1. Функциональное программирование. Применение и реализация [Текст] : [учеб. пособие для вузов] / П. Хендерсон ; пер. с англ. Л. Т. Петровой под ред. А. П. Ершова .— М. : Мир, 1983 .— 352 с.

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

1. SimonPeytonJones (editor), Haskell 98 LanguageandLibraries (TheRevisedReport)
2. JohnHughes, IntroductiontoProgramminginHaskell, www.cs.chalmers.se/~rjmh
3. PaulHudak, JohnPeterson, Joseph H. Fasel ,A GentleIntroductiontoHaskell 98
4. CordeliaHall, JohnHugs, ThelittleHaskeller
5. PhilipWadler, Monadsforfunctionalprogramming, DepartmentofComputingScience, UniversityofGlasgow

6. AllAboutMonads , <http://www.nomaware.com/monads/html/>
7. EmeryBerger , FP + OOP = Haskell, DepartmentofComputerScience, TheUniversityofTexasatAustin
8. RexPage, TwoDozenShortLessonsinHaskell, a participatorytextbookonfunctionalprogramming, SchoolofComputerScience, UniversityofOklahoma
9. DamirMedak, GerhardNavratil, Haskell-Tutorial, InstituteforGeoinformationTechnicalUniversityVienna
10. Главная страница языка Haskell, <http://haskell.org>.

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

ПО для разработки и отладки программ на языке программирования C++.

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

Студент, изучающий дисциплину, должен, с одной стороны, овладеть общим концептуальным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике. В результате изучения дисциплины студент должен знать основные определения, понятия, аксиомы.

Успешное развитие курса требует интенсивной самостоятельной работы студента. Программа курса предусматривает минимальное время, необходимое студенту для работы над темой. Самостоятельная работа включает в себя:

- чтение и ведение записей рекомендуемой литературы;
- изучение учебного материала (по конспектам лекций, учебной и научной литературе), подготовка ответов на вопросы, предназначенные для самостоятельного изучения, подтверждение индивидуальных высказываний, свойств;
- подготовка к экзамену.

Управление и контроль самостоятельной работы студента осуществляется в форме индивидуальных консультаций.

Важно получить понимание изучаемого материала, а не его механическое запоминание. Если трудно изучать отдельные темы, вопросы, вам следует проконсультироваться с лектором.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Прикладная математика и информатика
профиль подготовки:	Комбинаторика и цифровая экономика центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" кафедра дискретной математики
курс:	<u>1</u>
квалификация:	магистр
Семестр, формы промежуточной аттестации: 2 (весенний) - Дифференцированный зачет	
Разработчик:	А.Б. Дайняк, канд. физ.-мат. наук, доцент, доцент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1 Анализирует проблемную ситуацию как систему, выявляя ее составляющие и связи между ними
	УК-1.2 Осуществляет поиск вариантов решения поставленной проблемной ситуации на основе доступных источников информации
	УК-1.3 Разрабатывает стратегию достижения поставленной цели как последовательность шагов, предвидя результат каждого из них и оценивая их влияние на внешнее окружение планируемой деятельности и на взаимоотношения участников этой деятельности
ОПК-2 Способен совершенствовать и реализовывать новые математические методы решения прикладных задач	ОПК-2.1 Имеет представление о современном состоянии математических исследований в рамках тематической области своей профессиональной деятельности
	ОПК-2.2 Способен оценить актуальность и практическую значимость прикладных математических исследований в своей профессиональной области
	ОПК-2.3 Владеет профессиональной терминологией, используемой в современной научно-технической литературе, обладает навыками устного и письменного изложения результатов научной деятельности в рамках профессиональной коммуникации

2. Показатели оценивания компетенций

В результате изучения дисциплины «Алгоритмы» обучающийся должен:

знать:

- Алгоритмы на графах и структуры данных, связанные с ними,
- Оценки сложности стандартных алгоритмов.
- Стандартные алгоритмы на графах и используемые структуры данных, подходы к модификации классических алгоритмов.
- Разнообразные классические задачи в теории графов и асимптотические сложности их решений.

уметь:

- формулировать задачи в терминах изученных теорий, выбирать подходящий алгоритм для поставленной задачи;
- разрабатывать комбинации алгоритмов для решения поставленной задачи,
- оценивать сложности алгоритмов, их модификаций и комбинаций, в том числе с помощью амортизационного анализа,
- выбирать подходящие структуры данных для конкретной задачи,
- реализовывать алгоритм в обобщенной форме на языке программирования c++;
- реализовывать стандартные алгоритмы на графах и структуры данных на языке программирования C++.

владеть:

- Методами декомпозиции задач в области информационных технологий и построения единого решения с использованием изученных алгоритмов.
- Методами оценки сложности алгоритмов, их модификаций и комбинаций.

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

не предусмотрено

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

1. Какое исключение. Что термин пытается с ресурсами.
2. Опишите формат CSV.
3. Что такое дженерик? Когда стоит использовать?
4. Расскажите о работе GarbageCollector.
5. Перечислите методы класса Object.
6. Расскажите подробнее о методе общедоступного конечного нативного класса getClass ().
7. Расскажите подробнее о методе public native int hashCode ().
8. Расскажите подробнее об общедоступном методе логического равенства (Object obj).
9. Расскажите подробнее о защищенном собственном методе Object clone (), который генерирует исключение CloneNotSupportedException.
10. Расскажите подробнее об общедоступном методе String toString ().
11. Расскажите подробнее о том, что метод Throwable выбрасывает защищенный метод void finalize ().
12. Расширьте понятие хэш-функций и хеш-функций.
13. Опишите основные идеи Model-View-Controller.

пример экзаменационного билета:

1. Подробно расскажите о том, что метод finalize () защищает Throwable.
2. Что такое дженерик? Когда стоит использовать?

Критерии оценивания

отлично

- 10 всестороннее, систематизированное, глубокое знание учебного плана дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;
- 9 систематическое, глубокое знание учебного плана дисциплины и умение уверенно применять их на практике при решении конкретных задач, правильно обосновывать принятые решения;
- 8 глубокое знание учебного плана дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

хорошо

- 7 твердо знает материал, правильно и по существу излагает его, знает, как применять полученные знания на практике, но допускает некоторые неточности в ответе или решении проблем;
- 6 знает материал, правильно представляет его, знает, как применить полученные знания на практике, но допускает некоторые неточности в ответе или при решении проблем;
- 5 знает основной материал, правильно представляет его, знает, как применять полученные знания на практике, но допускает неточность в ответе или при решении проблем;

удовлетворительно

- 4 фрагментированный, фрагментированный характер знаний, недостаточно правильная формулировка основных понятий, нарушение логической последовательности в представлении программного материала, но в то же время он владеет основными разделами учебного плана, необходимыми для дальнейшего обучения, и может применять полученные знания в стандартной ситуации;
- 3 характер знаний достаточен для дальнейшего обучения и может применять полученные знания на модели в стандартной ситуации;

неудовлетворительный

- 2 не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировке основных понятий дисциплины и не знает, как правильно использовать полученные знания при решении типичных практических задач.

1 не знает формулировки основных понятий дисциплины и не умеет использовать полученные знания при решении типичных практических задач.

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

Время дифференцированного зачета - 2 академических часа.

Во время дифференцированного зачета студенты могут использовать программу дисциплины.