

МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра цифровых технологий и машинного обучения

УТВЕРЖДАЮ

Декан физического факультета



И.С. Огнев

«21» мая 2024 г.

Рабочая программа дисциплины
«Алгоритмы и структуры данных на Python»

03.03.03 Радиофизика

Направленность (профиль)
Технологии беспроводной связи

Форма обучения
очная

Программа рассмотрена	Программа одобрена НМК
на заседании кафедры цифровых технологий и машинного обучения	физического факультета
протокол № 8 от «26» апреля 2024 года	протокол № 5 от «30» апреля 2024 года

Ярославль

1. Цели освоения дисциплины

Дисциплина направлена на формирование у студентов фундаментальных знаний и навыков в области программирования и компьютерных наук. Целью курса является развитие умений и навыков решения практических задач с использованием языка Python, что включает в себя:

— Овладение методами реализации алгоритмов и изучение часто применяемых структур данных.

— Понимание специфики и реализации различных структур данных.

— Освоение алгоритмов обработки данных и подходов к их анализу.

— Изучение взаимосвязи между алгоритмами и структурами данных.

В результате прохождения курса студенты смогут не только эффективно программировать на Python, но и развить абстрактное мышление и расширить свой научно-технический кругозор. Курс поможет им овладеть:

— Практическими навыками программирования на примере решения различных задач.

— Основными алгоритмами и структурами данных.

— Методами анализа и оценки эффективности алгоритмов.

— Способностью к абстрактному мышлению и применению научных подходов в решении задач.

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Алгоритмы и структуры данных на Python» относится к элективным дисциплинам Блока 1 и изучается в течение одного семестра. Для успешного освоения данного курса студентам необходимо обладать следующими навыками и компетенциями:

- Умение логично, аргументированно и ясно формулировать свои мысли как устно, так и письменно.
- Способность к саморазвитию, стремление к повышению своей квалификации и мастерства.
- Навыки работы с компьютером для управления и обработки информации.
- Готовность учитывать современные тенденции в области информационных технологий.

Кроме того, для успешного изучения данной дисциплины требуется предварительное освоение курса «Программирование на языке Python», а также успешное овладение смежными курсами по общей физике и математике.

Знания, умения и навыки, приобретенные в рамках курса «Алгоритмы и структуры данных на Python», являются необходимыми для изучения последующих профильных дисциплин, продуктивного выполнения научно-исследовательской работы студентов (НИРС), а также для продолжения обучения в магистратуре.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата.

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Код	Формулировка	Перечень планируемых
-----	--------------	----------------------

компетенции	компетенции	результатов обучения
Общепрофессиональные компетенции		
ОПК-3. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности.	ИД_ОПК-3.2. Демонстрирует умение проектировать решение конкретных задач профессиональной деятельности, выбирая оптимальные способы их решения с использованием современных информационных технологий.	Знать: – базовые структуры данных и алгоритмы для работы с ними. Уметь: – выбирать структуры данных и/или алгоритмы, наиболее подходящие для решения конкретной задачи; – создавать алгоритмы решения математических задач; – выбирать наиболее подходящие вычислительные средства и методы, для решения задач в области профессиональной деятельности.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 2 зачетные единицы, 72 акад. часа.

Очная форма обучения

Содержание учебной программы									
№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания		
1.	Линейные структуры данных	4	2	4				2	Контроль по результатам выполнения лабораторных работ
2.	Рекурсивная обработка	4	2	4		1		2	Контроль по результатам

	иерархических списков								выполнения лабораторных работ
3.	Деревья и леса	4	2	4				2	Контроль по результатам выполнения лабораторных работ
4.	Исчерпывающий поиск	4	2	4				2	Контроль по результатам выполнения лабораторных работ
5.	Быстрый поиск	4	2	4				2	Контроль по результатам выполнения лабораторных работ
6.	Сортировка	4	2	4		1		2	Контроль по результатам выполнения лабораторных работ
7.	Алгоритмы на графах	4	2	5				2	Контроль по результатам выполнения лабораторных работ
8.	NP-полные и труднорешаемые задачи	4	3	5		1		2	Контроль по результатам выполнения лабораторных работ
							0,3	2,7	Зачёт
	Итого 72 часа		17	34		2	0,3	18,7	

Содержание разделов дисциплины:

Раздел 1. *Линейные структуры данных*

Структуры данных и алгоритмы. Стек, очередь и дек как линейные списки (последовательности) с ограниченными наборами операций (доступа). Стек, очередь и дек как абстрактные типы данных: функциональные спецификации и аксиомы. Представление и реализация (непрерывная, ссылочная в связанной памяти и на базе вектора). Примеры алгоритмов, использующих стек, очередь, дек.

Раздел 2. *Рекурсивная обработка иерархических списков*

Рекурсивное определение и функциональная спецификация линейных списков. Рекурсивное определение и функциональная спецификация иерархических (нелинейных) списков и S-выражений. Базовые функции (индикаторы, селекторы, конструкторы). Точечная форма записи S-выражений. Записи с вариантами в языках высокого уровня. Представление S-выражений и реализация базовых функций на языках высокого уровня. Элементы функционального программирования и рекурсивная обработка S-выражений на языках высокого уровня. Примеры использования нелинейных списков: дифференцирование символических выражений, действия с полиномами многих переменных.

Раздел 3. *Деревья и леса*

Определение дерева, леса, бинарного дерева. Графическое и текстовое (скобочное) представление леса. Спецификация дерева, леса, бинарного дерева: базовые функции и аксиомы. Естественное соответствие бинарного дерева и леса.

Обходы бинарных деревьев: рекурсивные и не рекурсивные алгоритмы. Обходы дерева или леса.

Представления и реализации бинарных деревьев: ссылочная реализация в связанной памяти, ссылочная реализация ограниченного бинарного дерева на базе вектора.

Прошитые бинарные деревья: представление, обход, включение.

Пример использования бинарных деревьев в задаче упаковки сообщений: префиксные коды и бинарные деревья, метод кодирования Фано-Шеннона, критерий оптимальности кода, алгоритм кодирования (сжатия) информации по Хаффмену (построение дерева, кодирование и декодирование), доказательство оптимальности кода Хаффмена, неравенство Крафта, теорема кодирования в отсутствие шума (энтропийная оценка средней длины кода). Динамическое кодирование по Хаффмену.

Раздел 4. *Исчерпывающий поиск*

Поиск с возвратом (backtracking). Общий алгоритм. Пример: задача о ферзях. Усовершенствования. Оценка сложности выполнения: метод Монте-Карло. Другие способы программирования поиска с возвратом: рекурсия и использование макросов.

Метод ветвей и границ. Общая схема. Задача коммивояжера: решение методом ветвей и границ. Эвристические методы: ближайшего соседа, ближайшего города. Оценки приближения.

Динамическое программирование. Пример (кратчайший путь в слоистой сети) и общая идея. Задача определения порядка умножения цепочки матриц.

Раздел 5. *Быстрый поиск*

Поиск и другие операции над таблицами. Последовательный и бинарный поиск. Бинарные деревья поиска. Случайные бинарные деревья поиска. Подсчет числа структурно различных бинарных деревьев с заданным числом узлов. Среднее время поиска в случайных деревьях. Рандомизированные бинарные деревья поиска (Treaps).

Оптимальные бинарные деревья поиска. Алгоритм построения оптимального дерева. Хорошие бинарные деревья поиска. Сбалансированные по высоте бинарные деревья (АВЛ-деревья). Включение в АВЛ-дерево. Исключение из АВЛ-дерева. Оценка сложности в худшем случае: деревья Фибоначчи. Реализация упорядоченных линейных списков на базе АВЛ-деревьев или рандомизированных деревьев. Операции поиска, вставки и удаления элементов; операции сцепления и расщепления списков.

2-3-деревья. Б-деревья. Метод поиска с использованием функции расстановки (хеширование). Разрешение коллизий: метод внутренних и внешних цепочек, метод открытой адресации. Коэффициент загрузки, оценки сложности. Выбор функции расстановки. Задача поиска подстроки. Алгоритм Кнута-Мориса-Пратта. Алгоритм Боуера-Мура.

Раздел 6. *Сортировка*

Задача сортировки (внешней и внутренней). Сортировка вставками, обменами, выбором. Быстрая сортировка Хоара. Процедура разделения. Рекурсивный и не рекурсивный алгоритмы быстрой сортировки. Анализ сложности. Оптимизация программы (неполная сортировка). Пирамидальная сортировка (HeapSorting): турнирная сортировка, построение пирамиды и полное упорядочение. Анализ сложности алгоритма.

Распределяющая (поразрядная) сортировка. Сравнение алгоритмов и программ внутренней сортировки. Нижняя граница сложности задачи сортировки. Оптимальная сортировка. Внешняя сортировка. Простое слияние. Естественное слияние. Задача поиска медианы: алгоритм Хоара, линейный алгоритм. Анализ сложности.

Раздел 7. *Алгоритмы на графах*

Графы: определения и примеры. Упорядоченный граф. Представления графов: матрица инцидентности, матрица смежности, список пар, структура смежности (списки инцидентности). Преобразования представлений. Остовные деревья графа. Минимальное остовное дерево. Теорема "о минимальном ребре". Жадный алгоритм (Краскал). Алгоритм "ближайшего соседа" (Прим, Дейкстра). Поиск в графе: алгоритм пометок. Поиск в ширину. Поиск в глубину. Связные компоненты. Алгоритм сложности $O(m \cdot \log n)$ построения минимального остова. Построение и свойства остовных деревьев при поиске в глубину и в ширину. Поиск в глубину и топологическая сортировка. Нахождение компонент двусвязности: точки сочленения графа и их свойства в глубинном остовном дереве. Алгоритм нахождения компонент двусвязности. Сильная связность. Поиск в глубину в орграфе. Алгоритм нахождения сильно связных компонент. Клики. Алгоритм порождения клик графа. Кратчайшие пути в графе. Кратчайшие пути от фиксированной вершины. Случай неотрицательных весов: алгоритм Дейкстры. Алгоритм Форда-Беллмана. Кратчайшие пути в бесконтурном графе. Кратчайшие пути между всеми парами вершин. Матрица смежности, матрица достижимости и транзитивное замыкание отношения, алгоритм Уоршалла. Алгоритм Флойда-Уоршалла вычисления расстояний между всеми парами вершин, одновременное построение путей.

Раздел 8. *NP-полные и труднорешаемые задачи*

Массовая и индивидуальная задачи. Сложность алгоритма и кодирование входных и выходных данных. Полиномиальные алгоритмы и класс P. Недетерминированные алгоритмы и класс NP. Различные формы постановки задач комбинаторной оптимизации: оптимизационная, вычислительная, форма распознавания. Примеры. Полиномиальная преобразуемость задач. NP-трудные и NP-полные задачи. Задача о выполнимости булева выражения, представленного в конъюнктивной нормальной форме. Доказательство NP-полноты задачи о выполнимости. Преобразуемость задачи о выполнимости в задачу о 3-выполнимости. Полиномиальность задачи о 2-выполнимости. Задача о клике графа. Преобразуемость задачи о 3-выполнимости в задачу о клике. Задача о многопроцессорном расписании (МПП). Преобразуемость задачи о клике в задачу о МПП. Задача о 0-1-рюкзаке и криптография. Практическое решение NP-полных задач.

5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и

организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция с элементами лекции-беседы – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Элементы лекции-беседы обеспечивают контакт преподавателя с аудиторией, что позволяет привлекать внимание студентов к наиболее важным темам дисциплины, активно вовлекать их в учебный процесс, контролировать темп изложения учебного материала в зависимости от уровня его восприятия.

При чтении лекций используется технология проблемного обучения (последовательное и целенаправленное выдвижение перед студентом познавательных задач, разрешая которые студенты активно усваивают знания). Курс построен на принципах системного подхода к отбору программного материала и определению последовательности его изучения студентами, что предусматривает глубокое изучение предмета.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

Консультации – вид учебных занятий, являющийся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты при освоении материала дисциплины, преподаватель отвечает на вопросы студентов, которые возникают у них в процессе самостоятельной работы.

Контроль качества усвоения программного материала проводится с учетом работы студентов на лекциях и лабораторных занятиях.

Самостоятельная работа студентов заключается в проработке конспекта лекций, изучении рекомендованной литературы, выполнении домашних и индивидуальных заданий.

В процессе обучения используются следующие технологии электронного обучения и дистанционные образовательные технологии:

Электронная версия учебного курса «Алгоритмы и структуры данных на Python» в котором:

- представлены задания для самостоятельной работы по темам дисциплины;
- представлены тексты лекций по темам дисциплины;
- представлен список учебной литературы, рекомендуемой для освоения дисциплины;

6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине

В процессе осуществления образовательного процесса по дисциплине используются:

- редактор программного кода Sublime Text 4;
- язык программирования Python 3;
- система контроля версий git;
- мессенджер Telegram;
- ОС Linux.

для формирования материалов для текущего контроля успеваемости и проведения промежуточной аттестации, для формирования методических материалов по дисциплине:

- программы AsciiDoctor;
- Microsoft Visual Studio Code.

7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

В процессе осуществления образовательного процесса по дисциплине используются:

Автоматизированная библиотечно-информационная система «БУКИ-NEXT»
http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php

8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет» (при необходимости), рекомендуемых для освоения дисциплины

а) Основная литература:

1. Алгоритмы: построение и анализ. / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн; пер. с англ. И. В. Красикова, Н. А. Ореховой, В. Н. Романова - 2-е изд. М.: Вильямс, 2012. 1290 с.
2. Ахо Альфред В. Структуры данных и алгоритмы: Учеб. пособие для вузов. / А. В. Ахо, Дж. Э. Хопкрофт, Дж. Д. Ульман. Пер. с англ. - М.: Вильямс, 2000. 382 с.

б) Дополнительная литература:

1. Вирт Н. Алгоритмы и структуры данных: Пер.с англ. / Н.Вирт - 2-е изд.,испр. - СПб.: Невский Диалект, 2001. 352с.

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

- учебные аудитории для проведения занятий лекционного типа;
- учебные аудитории для проведения практических работ, в том числе с использованием компьютерной техники;
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду ЯрГУ.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических работ – списочному составу группы обучающихся.

Авторы:

Доцент кафедры ЦТиМО, к.т.н.

И.В.Апальков

**Приложение №1 к рабочей программе дисциплины
«Алгоритмы и структуры данных на Python»**

**Фонд оценочных средств
для проведения текущего контроля успеваемости
и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания или иные материалы, используемые в процессе
текущего контроля успеваемости**

Текущий контроль успеваемости основан на анализе выполнения самостоятельных заданий.

Задания для самостоятельной работы

Задания по теме «Линейные структуры данных»

Разработать класс, реализующий стек.
Разработать класс, реализующий очередь.
Разработать класс, реализующий дек.

Задания по теме «Рекурсивная обработка иерархических списков»

Разработать класс, реализующий линейный список.
Разработать класс, реализующий иерархический (нелинейный) список.
Реализовать итератор для линейного списка.
Реализовать итератор для нелинейного списка.

Задания по теме «Деревья и леса»

Разработать класс, реализующий бинарное дерево.
Разработать класс, реализующий лес.
Разработать класс, реализующий обход бинарных деревьев на базе рекурсивных алгоритмов.
Разработать класс, реализующий обход бинарных деревьев на базе нерекурсивных алгоритмов.
Разработать класс, реализующий ограниченное бинарное дерево на базе вектора.

Задания по теме «Исчерпывающий поиск»

Разработать класс, реализующий алгоритм поиска с возвратом (backtracking).
Разработать класс, реализующий поиск с возвратом при использовании макросредств.
Разработать класс, реализующий методом ветвей и границ.
Разработать класс, реализующий поиск кратчайшего пути в слоистой сети.

Задания по теме «Быстрый поиск»

Разработать класс, реализующий последовательный поиск.
Разработать класс, реализующий бинарный поиск.

Разработать класс, реализующий поиск с помощью рандомизированных бинарных деревьев поиска (Treaps).

Разработать класс, реализующий упорядоченные линейные списки на базе AVL-деревьев или рандомизированных деревьев.

Разработать операции поиска, вставки и удаления элементов; операции сцепления и расщепления списков.

Реализация решения задачи поиска подстроки (алгоритм Кнута-Мориса-Пратта или алгоритм Боуера-Мура).

Задания по теме «Сортировка»

Разработать класс, реализующий сортировку вставками.

Разработать класс, реализующий сортировку обменов.

Разработать класс, реализующий сортировку выбором.

Разработать класс, реализующий быструю сортировку Хоара.

Реализовать рекурсивный и не рекурсивный алгоритмы быстрой сортировки.

Реализовать пирамидальную сортировку (HeapSorting).

Провести сравнение алгоритмов и программ внутренней сортировки.

Задания по теме «Алгоритмы на графах»

Разработать класс, реализующий упорядоченный граф.

Разработать класс, реализующий построение минимального остовного дерева.

Разработать класс, реализующий поиск в ширину.

Разработать класс, реализующий поиск в глубину.

Разработать класс, реализующий поиск в глубину и топологическую сортировку.

Разработать класс, реализующий алгоритм нахождения компонент двусвязности.

Разработать класс, реализующий поиск в глубину в орграфе.

Разработать класс, реализующий нахождение кратчайшего пути от фиксированной вершины (алгоритм Дейкстры, алгоритм Форда-Беллмана).

Задания по теме «NP-полные и труднорешаемые задачи»

Показать сложность алгоритма и кодирование входных и выходных данных.

Показать различные формы постановки задач комбинаторной оптимизации: оптимизационная, вычислительная, форма распознавания.

Разработать класс, реализующий полиномиальные преобразуемые задачи.

Разработать класс, реализующий решение задачи о выполнимости булева выражения, представленного в конъюнктивной нормальной форме.

Выполнить доказательство NP-полноты задачи о выполнимости.

Разработать класс, реализующий решение задачи о клике графа.

Разработать класс, реализующий решение задачи о многопроцессорном расписании.

Критерии оценивания ответов на вопросы при защите работ

Показатели	На «Зачтено»	На «Не зачтено»
Формулы, команды, вычисления	Применяются необходимые в условиях поставленной задачи команды без синтаксических ошибок; вычисления выполнены в полном объеме и представлены в	Используются неприменимые в условиях поставленной задачи команды; в базовых выражениях допущены ошибки; вычисления отсутствуют или выполнены

Показатели	На «Зачтено»	На «Не зачтено»
	соответствующем заданию формате.	некорректно.
Графики	Построенная зависимость имеет верный вид, по осям указаны аргумент и имя функции со своими единицами, есть шкалы на осях, нанесены контрольные метки, соответствующие заданию.	Вид зависимостей неверный, соответствующее заданию оформление графиков отсутствует.
Объяснения (ответы на смысловые вопросы)	Даны развёрнутые, корректные ответы на все вопросы; правильно интерпретируются результаты вычислений	Объяснение отсутствует.

Шкала оценивания:

0 баллов – «не зачтено»;

1 балл – «зачтено»;

Суммируются баллы за каждую работу.

Допуск к зачёту осуществляется по количеству набранных баллов:

менее 60% от максимально возможного количества баллов – к зачёту не допускается;

60-80% от максимально возможного количества баллов – помимо основного зачётного задания выполняются дополнительные задания по пропущенным темам;

80-99% от максимально возможного количества баллов – основное зачётное задание;

100% от максимально возможного количества баллов – проставляется зачёт автоматом.

2. Список вопросов и (или) заданий для проведения промежуточной аттестации

Список заданий к зачету

Зачет выставляется по результатам выполнения самостоятельных работ.

**Приложение №2 к рабочей программе дисциплины
«Алгоритмы и структуры
данных на Python»**

Методические указания для студентов по освоению дисциплины

Основной формой усвоения учебного материала по дисциплине **«Алгоритмы и структуры данных на Python»** является практическая работа студента, причем в достаточно большом объеме. По всем темам предусмотрены задания самостоятельной работы, на которых происходит закрепление изученного материала и отработка навыков работы с компьютером и операционной системой.

Освоить вопросы дисциплины **«Алгоритмы и структуры данных на Python»** самостоятельно студенту достаточно сложно. Посещение всех предусмотренных занятий является совершенно необходимым. Без упорных и регулярных самостоятельных занятий в течении 2 семестров сдать зачеты практически невозможно