

**Федеральное государственное автономное образовательное учреждение
высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет компьютерных наук
Департамент анализа данных и искусственного интеллекта

Рабочая программа дисциплины «Алгоритмы и структуры данных»

для образовательной программы Прикладная математика и информатика
для направления 01.03.02 «Прикладная математика и информатика»
уровень – бакалавр

Разработчик программы:
Объедков С.А., к.т.н., доцент, sergei.obj@gmail.com

Одобрена на заседании департамента анализа данных и искусственного интеллекта
«__»_____ 2016 г.
Руководитель департамента анализа данных и искусственного интеллекта
С.О. Кузнецов _____

Утверждена Академическим советом образовательной программы
«__»_____ 2016 г., № протокола _____

Академический руководитель образовательной программы
А.С. Конушин _____

Москва, 2016

Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения кафедры-разработчика программы.



1 Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов направления подготовки/ специальности 01.03.02 «Прикладная математика и информатика», изучающих дисциплину «Алгоритмы и структуры данных».

Программа разработана в соответствии с:

- Образовательным стандартом федерального государственного автономного образовательного учреждения высшего профессионального образования «Национальный исследовательский университет «Высшая школа экономики»;
- Образовательной программой «Прикладная математика и информатика»;
- Рабочим учебным планом университета по направлению 01.03.02 «Прикладная математика и информатика» подготовки бакалавра, утвержденным в 2015 г.

2 Цели освоения дисциплины

Целями освоения дисциплины «Алгоритмы и структуры данных» являются ознакомление студентов с основными принципами проектирования и анализа алгоритмов и структур данных, развитие навыков обоснования корректности алгоритмов, их практической реализации, теоретической и экспериментальной оценки их временной сложности.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- Знать о наиболее важных алгоритмах и структурах данных и основных принципах их проектирования и анализа;
- Уметь обосновывать корректность алгоритмов, проводить теоретическую и экспериментальную оценки их временной сложности;
- Иметь навыки реализации алгоритмов на языках Python и C++.

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ФГОС/ НИУ	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции
Способность к анализу и синтезу на основе системного подхода	ОНК-1	Умение проводить анализ корректности и временной сложности алгоритмов	Лекции, практические занятия, домашние задания
Способность перейти от проблемной ситуации к проблемам, задачам и лежащим в их основе противоречиям	ОНК-2	Умение формализовать условие задачи, требующей алгоритмического решения; умение разбить задачу на подзадачи	Лекции, практические занятия, домашние задания
Способность приобретать новые знания с использованием научной методологии и современных образовательных и информационных технологий	ОНК-6	Способность находить и самостоятельно осваивать нужную информацию из общедоступных источников	Лекции, практические занятия, домашние задания
Умение работать на ком-	ИК-2	Способность программировать,	Лекции, практические заня-



Компетенция	Код по ФГОС/НИУ	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции
пьютере, навыки использования основных классов прикладного программного обеспечения, работы в компьютерных сетях, составления баз данных		отлаживать и тестировать алгоритмы и структуры данных	тия, домашние задания
Владение культурой мышления и публичного выступления, умение аргументировано и ясно строить устную и письменную речь	СЛК-5	Умение формулировать и объяснять доказательства теорем в устной и письменной форме	Лекции, практические занятия, домашние задания
Способность использовать в научной и познавательной деятельности, а также в социальной сфере профессиональные навыки работы с информационными и компьютерными технологиями	СЛК-10	Умение корректно пользоваться легальными источниками, размещенными в сети Интернет	Практические занятия, домашние задания
Способность составлять и контролировать план выполняемой работы, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы	ПК-11	Умение оптимально распределять собственное время при работе над поставленными задачами	Практические занятия, домашние задания

4 Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к циклу математических и естественнонаучных дисциплин, является базовой для подготовки бакалавра по направлению 01.03.02 «Прикладная математика и информатика».

Изучение данной дисциплины базируется на следующих дисциплинах:

- «Основы и методология программирования»

Для освоения учебной дисциплины, студенты должны владеть следующими знаниями и компетенциями:

- Умение формализовать условие задачи по программированию
- Умение применять язык программирования в новых ситуациях
- Способность записать и выполнить программу на компьютере на требуемых языках программирования
- Умение логически анализировать информацию, умение пользоваться документацией

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

- Технологии программирования
- Машинное обучение
- Вычислительные методы
- Алгоритмы для больших данных



- Теория вычислений
- Алгоритмы и сложность
- Логические методы в информатике
- Комбинаторные методы в информатике
- Модели вычислений
- Вероятностные алгоритмы и протоколы
- Теоретическая информатика
- Компьютерная лингвистика
- Майнинг данных
- Методы обработки текстов на естественном языке
- Распределенные алгоритмы

5 Тематический план учебной дисциплины

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
1	Алгоритмы типа «разделяй и властвуй»	52	12		12	28
2	Динамическое программирование	27	6		6	15
3	Основные алгоритмы на графах	52	12		12	28
4	Жадные алгоритмы	51	12		12	27
5	Структуры данных	68	16		16	36
6	Потоки в сетях	42	10		10	22
7	Труднорешаемые задачи и методы их решения	50	12		12	26

6 Формы контроля знаний студентов

Тип контроля	Форма контроля	1 год		Параметры
		3	4	
Текущий (неделя)	Контрольная работа		9	Решение задач в компьютерной системе, от 80 до 180 минут.
	Домашнее задание	1 – 3, 4 – 7, 8 – 10	1 – 4, 5 – 7	Решение задач в компьютерной системе и письменное решение задач с последующим устным объяснением. Задания разбиты на части, для каждой из которых указывается срок выполнения.
Итоговый	Экзамен		*	Устный экзамен, 3 часа.

6.1 Критерии оценки знаний, навыков

При прохождении контроля студент должен продемонстрировать владение различными техниками проектирования и анализа алгоритмов, навыки программирования и тестирования алгоритмов и структур данных, а также умение формулировать и доказывать математические утверждения.



Оценки по всем формам текущего контроля выставляются по 10-ти балльной шкале.

6.2 Порядок формирования оценок по дисциплине

Преподаватель оценивает работу студентов на практических занятиях: правильность решения теоретических задач, корректность и эффективность компьютерных программ, написанных студентом. Оценки за работу на семинарских и практических занятиях преподаватель выставляет в рабочую ведомость. Накопленная оценка по 10-ти балльной шкале за работу на семинарских и практических занятиях определяется перед промежуточным контролем — $O_{аудиторная}$.

Накопленная оценка за текущий контроль рассчитывается следующим образом:

$$O_{накопленная} = 0,2 \cdot O_{к/р} + 0,12 \cdot O_{дз1} + 0,12 \cdot O_{дз2} + 0,12 \cdot O_{дз3} + 0,12 \cdot O_{дз4} + 0,12 \cdot O_{дз5} + 0,2 \cdot O_{аудиторная}.$$

Способ округления накопленной оценки текущего контроля: на усмотрение преподавателя до ближайшего сверху или снизу целочисленного значения.

Результирующая оценка за дисциплину рассчитывается следующим образом:

$$O_{результ} = 0,7 \cdot O_{накопл} + 0,3 \cdot O_{экз}.$$

Способ округления результирующей оценки по учебной дисциплине: на усмотрение преподавателя до ближайшего сверху или снизу целочисленного значения.

В соответствии с пунктом 2.4 Приложения 7 «Порядок применения дисциплинарных взысканий при нарушениях академических норм в написании письменных учебных работ в НИУ ВШЭ» к Правилам внутреннего распорядка Национального исследовательского университета «Высшая школа экономики» оценка за работу в рамках текущего или промежуточного контроля обнуляется при обнаружении нарушений студентом академических норм (списывания, двойной сдачи, плагиата, подлога, запрещенной условиями выполнения задания совместной работы), характер которых ставит под сомнение самостоятельность выполнения работы или одного из ее основных разделов. Данная работа считается не выполненной.

7 Содержание дисциплины

1. Раздел 1. Алгоритмы типа «разделяй и властвуй»

Алгоритмы сортировки, основанные на принципе «разделяй и властвуй» (сортировка слиянием, быстрая сортировка), и другие (сортировка вставкой). Двоичное дерево поиска и сортировка при помощи такого дерева; связь с алгоритмом быстрой сортировки.

Анализ алгоритмов в терминах O -обозначений. Рекуррентные соотношения и их решение. Основная теорема.

Примеры алгоритмов, основанных на принципе «разделяй и властвуй»: выбор порядковой статистики, быстрое возведение в степень, умножение целых чисел.

2. Раздел 2. Динамическое программирование

Основные принципы, примеры алгоритмов: составление расписания для взвешенных интервалов, выравнивание текста по ширине, выравнивание последовательностей.

3. Раздел 3. Основные алгоритмы на графах

Представление графов в виде списков смежности и матрицы смежности. Обход графа в глубину и ширину. Связность в ориентированных и неориентированных графах. Двухнаправленный по-

иск путей в графах. Поиск кратчайших путей во взвешенном графе, алгоритмы Беллмана – Форда, Флойда – Уоршелла. Приложения алгоритмов поиска путей: задача о выполнимости 2-КНФ и др.

4. Раздел 4. Жадные алгоритмы.

Основные принципы, примеры алгоритмов. Поиск кратчайших путей в графе при помощи алгоритма Дейкстры. Минимальные остовные деревья: алгоритмы Прима и Крускала. Применение алгоритма Крускала для кластеризации. Жадные алгоритмы на матроиде.

5. Раздел 5. Структуры данных

Линейные структуры данных. Амортизационный анализ. Двоичные и биномиальные кучи. Система непересекающихся множеств. Хеш-таблицы. Сбалансированные деревья поиска, красно-черные деревья. Списки с пропусками. Самоорганизующиеся списки и конкурентный анализ онлайн-алгоритмов.

6. Раздел 6. Потоки в сетях.

Определение потока, циркуляции. Задача о максимальном потоке. Алгоритм Форда – Фалкерсона. Максимальный поток и минимальный разрез. Максимальное паросочетание в двудольном графе. Совершенное паросочетание с минимальным весом во взвешенном двудольном графе.

7. Раздел 7. Труднорешаемые задачи и методы их решения.

Понятие NP-полноты, примеры NP-полных задач. Рандомизированные и приближенные алгоритмы. Эффективное перечисление последовательностей. Полиномиальная задержка, кумулятивная задержка, сложность алгоритма относительно размера входа и выхода.

8 Образовательные технологии

В преподавании дисциплины сочетаются лекции, практические занятия в компьютерном классе и домашние задания, состоящие из теоретических задач и задач на программирование.

9 Оценочные средства для текущего контроля и аттестации студента

9.1 Тематика заданий текущего контроля

Примерные задания для домашней работы:

1. Оцените сложность данного алгоритма относительно размера входа.
2. Приведите рекуррентное соотношение, описывающее сложность данного алгоритма и перепишите это соотношение в замкнутой форме.
3. Перед вами данные об изменении курса доллара за n дней. Обозначим за $r(i)$ курс доллара к рублю в i -ый день. Вы хотите понять, в какой из этих n дней стоило купить доллары, а в какой — продать, чтобы максимизировать прибыль в рублях (в предположении, что в первый день у вас были только рубли). Разработайте и реализуйте алгоритм, работающий в рамках стратегии «разделяй и властвуй» и решающий эту задачу за время $O(n \log n)$. Если прибыль получить невозможно, алгоритм должен сообщить об этом.
4. Перед вами ряд из n карт, на каждой из которых написано некоторое число. Все числа различны, а карты лежат числом вниз. Требуется найти карту, на которой написано число, меньшее, чем числа на всех соседних картах. Разрешается перевернуть $O(\log n)$ карт. Как решить задачу?
5. Назовем два пути в графе *непересекающимися по ребрам*, если они не содержат одинаковых ребер (но могут содержать одинаковые вершины). Требуется выяснить, есть ли в заданном ориентированном графе k путей, непересекающихся по ребрам, из вершины s в вершину t . Разработайте и реализуйте эффективный алгоритм решения этой задачи.



6. Дед Мороз готовит подарки к Новому году. У Деда Мороза есть заводной автомобиль, книжка–раскраска, кукла Таня, кулек конфет и еще $n - 4$ различных подарка. Каждый из них имеется в неограниченном количестве, так что проблем вроде бы быть не должно. Однако Деду Морозу кажется, что кулек конфет и заводной автомобиль — неравноценные подарки, поэтому к кульку конфет он, скорее всего, добавит книжку–раскраску или что-то еще. Дед Мороз уже составил k подарочных наборов, каждый из которых содержит не более одного подарка каждого типа (т.е. в одном наборе — не более одной куклы Тани). Принципы Деда Мороза не позволяют ему дарить одному ребенку явно больше, чем другому, поэтому каждый из любых двух наборов содержит какой-то подарок, которого нет в другом наборе. Дед Мороз пытается составить еще один — $(k + 1)$ -ый — подарочный набор, согласующийся с его принципами, но у него не получается. Предложите алгоритм, определяющий, существует ли такой $(k + 1)$ -ый набор (для заданных k наборов и n типов подарков), приведите оценку времени работы предложенного алгоритма и реализуйте его.
7. Пусть $X[1..n]$ и $Y[1..n]$ — два отсортированных массива, каждый из которых содержит по n элементов. Разработайте и реализуйте алгоритм, в котором поиск медианы всех $2n$ элементов, содержащихся в массивах X и Y , выполнялся бы за время $O(\log n)$.
8. Даны два массива по n элементов в каждом: i -ый элемент первого массива — ставка i -го игрока на тотализаторе, i -ый элемент второго массива — выигрыш i -го игрока. Все ставки различны, все выигрыши положительны. Обозначим за W сумму всех выигрышей. Требуется найти ставки каждого игрока x , для которого верно следующее: суммарный выигрыш игроков, поставивших меньше игрока x , не превышает $W/2$, и суммарный выигрыш игроков, поставивших больше, не превышает $W/2$. Разработайте и запрограммируйте эффективный алгоритм решения задачи и приведите оценку его времени работы.

9.2 Вопросы для оценки качества освоения дисциплины

Примерный перечень вопросов к экзамену по всему курсу:

1. Нарисуйте любое красно-черное дерево с ключами 1, 2, 3, 4, 5 и 6. Покажите, как изменится это красно-черное дерево при добавлении к нему ключа 7.
2. Докажите или опровергните: если $f(n) = O(g(n))$ и $g(n) = O(h(n))$, то $f(n) = O(h(n))$.
3. Докажите или опровергните: если $f_1(n) = O(g(n))$ и $f_2(n) = O(g(n))$, то $f_1(n) + f_2(n) = O(g(n))$.
4. *Вершинным покрытием* неориентированного графа называется подмножество его вершин, включающее хотя бы одну из двух вершин каждого ребра. *Независимым множеством* неориентированного графа называется подмножество его вершин, в котором никакие две вершины не связаны ребром. Допустим, у вас есть алгоритм A , вычисляющий вершинное покрытие минимального размера за время $O(f(n))$, где $f(n) = \Omega(n)$. Предложите алгоритм B , вычисляющий независимое множество максимального размера за время $O(f(n))$. Поясните ответ.
5. Разработайте и реализуйте алгоритм, сортирующий массив из n чисел, каждое из которых равно -1, 0 или 1, за время $O(n)$ в худшем случае.
6. Решите рекуррентное соотношение: $T(n) = 8T(n/2) + n^3$.
7. В некоторой программе осуществляется n последовательных вызовов операции f . Известно, что эта последовательность вызовов занимает время $\Theta(n \log n)$ в худшем случае. Каким может быть максимальное время выполнения (в терминах Θ) одной операции f из этой последовательности? Объясните ответ.
8. Пусть есть два множества чисел S и T , реализованные в виде связных списков. Все элементы в каждом из списков различны. Длины списков неизвестны, доступ к элементам списков по индексу невозможен. Любая операция над одним числом или двумя числами занимает константное время. Предложите эффективный алгоритм, определяющий, верно ли, что $S = T$, т.е. что списки, реализующие множества S и T , содержат одинаковые эле-



менты (но, возможно, в разном порядке). В идеале, ожидаемое (среднее) время работы алгоритма — $O(\min \{|S|, |T|\})$.

10 Учебно-методическое и информационное обеспечение дисциплины

10.1 Базовый учебник

Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К. Алгоритмы: построение и анализ. — 2-е издание: Пер. с англ. — М.: Вильямс, 2007.

10.2 Основная литература

Дасгупта С., Пападимитриу Х., Вазирани У. Алгоритмы. — М.: МНЦМО, 2014.

10.3 Дополнительная литература

Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. — М.: Мир, 1982.

Лутц М. Изучаем Python. СПб.: Символ-Плюс, 2009.

Kleinberg J., Tardos E. (2005). Algorithm Design. Addison-Wesley.

Stroustrup, B. (2013). The C++ Programming Language (4th edition). Addison-Wesley.

10.4 Программные средства

Для успешного освоения дисциплины, студент использует следующие программные средства:

- интерпретатор языка Python 3.x;
- компилятор языка C++11 (g++ 4.8)