

**Санкт-Петербургский филиал федерального государственного
автономного образовательного учреждения высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет Санкт-Петербургская школа экономики и менеджмента
Национального исследовательского университета
«Высшая школа экономики»

Департамент прикладной математики и бизнес-информатики

**Рабочая программа дисциплины
Алгоритмы и структуры данных**

для образовательной программы «Анализ больших данных в бизнесе, экономике и обществе»
направления подготовки 01.04.02«Прикладная математика и информатика»
уровень магистратура

Разработчик(и) программы:

Колганов Р.А., преподаватель, roman.kolganov@gmail.com;

Сироткин А.В., к.ф.-м.н, доцент департамента прикладной математики и бизнес-информатики,
avsirotkin@hse.ru

Согласована менеджером ОП Анализ больших данных в бизнесе, экономике и обществе
«29» августа 2016г.

Е.С. Авдонина _____

Утверждена Академическим руководителем образовательной программы

А.В. Сироткин _____

«30» августа 2016г.

Санкт-Петербург, 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 рабочая программа дисциплины устанавливает минимальные требования к знаниям и умениям студента, а также определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих дисциплину Алгоритмы и структуры данных, учебных ассистентов и студентов направления подготовки 01.04.02 «Прикладная математика и информатика», обучающихся по образовательной программе «Анализ больших данных в бизнесе, экономике и обществе».

Рабочая программа дисциплины разработана в соответствии с:

- Образовательным стандартом НИУ ВШЭ
<http://www.hse.ru/data/2016/11/02/1111123560/01.04.02%20Прикладная%20математика%20и%20информатика.pdf>;
- Образовательной программой «Анализ больших данных в бизнесе, экономике и обществе», направление подготовки 01.04.02 «Прикладная математика и информатика», ;
- Объединенным учебным планом университета по образовательной программ «Анализ больших данных в бизнесе, экономике и обществе».

2 Цели освоения дисциплины

Целью освоения дисциплины «Алгоритмы и структуры данных» являются:

- изучение основных алгоритмов и структур данных, широко применяющихся в современной информатике.
- формирование представления об основных алгоритмах и структурах данных, алгоритмах на графах и строках, динамическом программировании, жадных алгоритмах.
- формирование представления об анализе алгоритмов, теории сложности алгоритмов и способах сравнения алгоритмов и структур данных между собой.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент осваивает следующие компетенции:

Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
<i>Системные компетенции</i>					
Способен рефлексировать (оценивать и перерабатывать) основные научные методы и способы деятельности.	СК-1,	РБ, СД	Способен описывать задачу на математическом языке. Умеет подбирать корректные алгоритмы для решения поставленных задач.	Лекции, семинарские занятия	Работа на семинарах, экзамен
Способен анализировать, верифицировать, оценивать полноту информации в ходе профессиональной деятельности, при необходимости восполнять и	СК-6	РБ, СД	Способен давать оценку сложности поставленной задачи и оценивать вычислительную сложность алгоритмов решения задачи.	Лекции, семинарские занятия	Работа на семинарах



Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
синтезировать недостающую информацию.					
Профессиональные компетенции					
Инструментальные компетенции					
Способен описывать проблемы и ситуации профессиональной деятельности, используя язык и аппарат прикладной математики при решении междисциплинарных проблем.	ПК-14	РБ, СД	Способен формулировать математически-точные описания прикладных задач и проводить их сведение к уже известным.	Семинарские занятия	Работа на семинарах, контрольная работа, экзамен

4 Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к базовой части цикла дисциплин направления 01.04.02. «Прикладная математика и информатика».

Изучение данной дисциплины базируется на следующих дисциплинах обучения в бакалавриате: «Линейная алгебра», «Математический анализ», «Теория вероятностей и математическая статистика».

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

- способен анализировать, верифицировать, оценивать полноту информации в ходе профессиональной деятельности, при необходимости восполнять и синтезировать недостающую информацию;
- способен описывать проблемы и ситуации профессиональной деятельности, используя язык и аппарат прикладной математики при решении междисциплинарных проблем;
- способен создавать междисциплинарные тексты с использованием языка и аппарата прикладной математики;
- способен рефлексировать (оценивать и перерабатывать) основные научные методы и способы деятельности.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин: «Распределенная обработка и анализ больших данных», «Информационный поиск и обработка текстов на естественном языке».

5 Тематический план учебной дисциплины

ОБЪЕМ ДИСЦИПЛИНЫ - 3 зачетные единицы.

№	Название раздела	Всего часов	Аудиторные часы		Самостоятельная работа
			Лекции	Семинары	



1	Введение. Числовые алгоритмы. Рекуррентные соотношения. Вычислительная сложность.	10	2		8
2	Алгоритмы сортировки.	16	2	2	12
3	Декомпозиция графов. Пути в графах.	24	4	4	16
4	Жадные алгоритмы. Динамическое программирование.	24	4	4	16
5	Структуры данных: список, массив, стек, очередь, хеш-таблица, очередь с приоритетами.	22	2	4	16
6	Алгоритмы на строках: поиск подстроки.	18	2	2	14
ИТОГО		114	16	16	82

6 Формы контроля знаний студентов

Тип контроля	Форма контроля	1 модуль								Параметры
		Сентябрь				Октябрь				
Текущий	Работа на семинарах				*				*	Предлагаются фрагменты кода, которые необходимо закончить или исправить
	Контрольная работа								*	Решение задач на компьютере 80 минут
Итоговый	Экзамен					*				Экзамен в форме решения задач на компьютере - 120 минут

7 Критерии оценки знаний, навыков

Критерии оценки работы на семинарах:

Правильный выбор алгоритмов и структур данных для решения поставленных задач.

В течении занятий студентам предлагаются фрагменты кода, в которых отсутствуют ключевые моменты решения задачи. Студенты должны восполнить отсутствующие фрагменты кода.

Общее количество заданий может составлять от 8 до 15.

Оценка знаний студентов за работу на семинарах ($O_{\text{ауд_сем}}$) вычисляется как

$$O_{\text{ауд_сем}} = 10 * \Pi / 3,$$

где Π - количество правильно решенных заданий, а 3 - общее число заданий выданных на семинарах.

$O_{\text{ауд_сем}}$ округляется по правилам арифметики. Максимальное значение, учитываемое за работу на семинарских занятиях при расчете накопленной оценки по 10-балльной системе – **10 баллов**.

Критерии оценки контрольной работы:

Прохождение созданной программой тестов. Правильность выбора использованных алгоритмов. Дается 5 задач. Каждая задача оценивается по двум критериям, оптимальность и корректность выбора алгоритма, и прохождение формальных тестов. За соответствие каждому из двух

критериев начисляется один балл. Итоговый балл за контрольную, вычисляется как сумма баллов за все задачи.

Максимальное значение, учитываемое за контрольную работу при расчете накопленной оценки по 10-балльной системе – **10 баллов**.

Критерии оценки за экзамен:

Экзамен ($O_{\text{экз}}$) проводится в виде решения задач на компьютере и содержит 5 задач. Задача считается решенной полностью и за нее начисляется 2 балла, если она проходит все специально подготовленные тесты. По окончании экзамена, задачи не проходящие все тесты, оцениваются преподавателем. Если задача содержит незначительные ошибки, то за нее начисляется 1 балл.

Оценка за экзамен выставляется как сумма баллов за каждую из 5 задач.

Максимальная оценка за экзамен – **10 баллов**.

8 Содержание дисциплины

Тема № 1. Введение. Числовые алгоритмы. Рекуррентные соотношения. Вычислительная сложность.

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

Тема № 2. Алгоритмы сортировки.

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.

Тема № 3. Декомпозиция графов. Пути в графах.

Представление графов в виде списков смежности и матрицы смежности. Обход графа в глубину и ширину. Связность в ориентированных и неориентированных графах. Двухнаправленный поиск путей в графах. Поиск кратчайших путей во взвешенном графе, алгоритмы Беллмана – Форда, Флойда – Уоршелла.

Тема № 4. Жадные алгоритмы. Динамическое программирование.

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

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

Тема № 5. Структуры данных: список, массив, стек, очередь, хеш-таблица, очередь с приоритетами.

Линейные структуры данных. Амортизационный анализ. Двоичные и биномиальные кучи. Система непересекающихся множеств. Хеш-таблицы.

Тема № 6. Алгоритмы на строках: поиск подстроки.

Варианты задачи поиска подстроки в строке. Использование Z -функции для предобработки и поиска подстроки. Понятие об алгоритме поиска реального времени. Алгоритм Кнута-Морриса-Пратта, префикс-функция. Алгоритм построения префикс-функции. Линейность времени его работы.

9 Образовательные технологии

Для проведения занятий со студентами используются:

- презентационные технологии;
- компьютерные технологии.

9.1 Методические рекомендации преподавателю

Особых рекомендаций для преподавателя нет.

9.2 Методические указания студентам по освоению дисциплины

Самостоятельная работа студента предусматривает выполнение теоретических заданий, направленных на овладение техникой построения алгоритмов поиска и сортировки, а также практических заданий по программной реализации этих алгоритмов.

При реализации алгоритмов рекомендуется обращать внимание на крайние значения параметров и тестировать работоспособность алгоритмов при таких параметрах.

9.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

- Свободно доступный курс "Алгоритмы: теория и практика. Методы" <https://stepik.org/course/Алгоритмы-теория-и-практика-Методы-217>.
- Сайт с задачами по программированию и автоматической проверкой задач: <http://codeforces.com/>

10 Оценочные средства для текущего контроля и аттестации студента

10.1 Оценочные средства для оценки качества освоения дисциплины в ходе текущего контроля

Примеры задания для оценки работы на семинарах:

1. Завершите функцию возвращающую список всех простых делителей числа.

```
def factors(n):  
    result = []  
    ...  
    return result
```

2. Завершите функцию возвращающую остаток от деления y -вой степени числа x на z .

```
def powmod(x, y, z):  
    ...
```

3. Завершите функцию находящую n -ый по величине элемент без сортировки списка.

```
import random  
# lst is a list of numbers, n is a number  
# returns sorted(lst)[n] without sorting lst  
# there are no big tests, but please implement O(n) algorithm  
# just replace None's and NotImplemented's with appropriate code  
def nth_element(lst, n):  
    if len(lst) == 1:  
        return NotImplemented  
    pivot = random.choice(lst)  
    less = None  
    equal = None  
    greater = None  
    if len(less) > n:  
        return NotImplemented  
    elif len(less) + len(equal) > n:  
        return NotImplemented  
    else:  
        return NotImplemented
```

3. Завершите функцию проверяющую что s представляет собой корректную скобочную структуру.

```
# s is a string containing symbols '(', ')', '[', ']'
# funcuions returns True, if s is correct brackets sequence,
False otherwise
# use stack!
def check_brackets(s):
    stack = []
    pairs = {')' : '(', ']' : '['}
    return NotImplemented
```

Пример задания контрольной работы:

1. Опишите, используя псевдокод, алгоритмы сортировки q-sort и сортировки вставкой. Укажите их сложность в среднем и в худшем случае.
2. Опишите, используя псевдокод, алгоритм кодирования заданной строки по Хаффману, и реализуйте заданный алгоритм на любом языке программирования.
3. Реализуйте алгоритм решающий задачу непрерывного рюкзака.
4. Закончите реализацию функции возвращающей наибольший общий делитель двух неотрицательных чисел:

```
def gcd(a, b):
    print('enter recursive call, a =', a, 'b =', b)
    ...
```

5. Реализуйте алгоритм сортировки слиянием.

10.2 Примеры заданий итогового контроля

1. Вам дан лабиринт, представленный матрицей (список списков), каждая клетка лабиринта это или 0 или 1. Вы можете перемещаться между соседними клетками, только если они обе 0. Ваша задача написать функцию, которая по размеру лабиринта N и квадратной матрице из 0 и 1 найдет кратчайший путь из точки (0,0) в точку (N-1, N-1). Путь всегда существует.

Например:

```
path_trough_maze(2, [[0,0],[0,0]]) должно вернуть 2.
path_trough_maze(5,
```

```
[[0,0,0,0,0],[1,1,1,1,0],[0,0,0,0,0],[0,1,1,1,1],[0,0,0,0,0]]) должно вернуть 16.
```

2. Напишите функцию, которая для заданного числа $N \leq 1000$ выводит N-ное простое число. Например, `primes(2)` должно возвращать 3-ку, а `primes(138)` -- 787.

3. Лягушка прыгает по липким листочкам выстроенным в ряд. Она может прыгнуть на один, два или три листа вперед. Чтобы прыгнуть на i листов вперед, лягушке нужно потратить i^2 энергии. Кроме этого, для каждого листа указана его липкость, то есть количество энергии которое нужно потратить, чтобы спрыгнуть с этого листа. Чтобы прыгнуть на 2 листа вперед с листа с липкостью 2 нужно потратить $2^2+2=6$ единиц энергии.

Ваша задача для заданного списка липкостей вычислить минимальную энергию необходимую лягушке, чтобы допрыгать с первого до последнего листа.

Например:

```
minimal_frog_energy([1,1,1,1,1,1,1,1]) должно возвращать 14, а
minimal_frog_energy([1,8,5,1]) должно возвращать 10.
```

4. У вас есть одна аудитория и много профессоров. Каждый профессор хочет прочитать лекцию, при этом i-тый профессор хочет начать лекцию в s_i минут, а закончить в e_i минут. Вы составляете расписание, при этом если аудиторию занял профессор i, то вы можете поставить лекцию профессора j только если $e_i < s_j$.

Имея список пожеланий профессоров, вычислите сколько максимум лекций можно будет поставить в одну имеющуюся аудиторию.

При реализации, пользоваться встроенными сортировками НЕЛЬЗЯ.

Напишите функцию `max_lectures`, которая по заданным спискам `e` и `s` вычисляет максимальное возможное число лекций.

Например, `max_lectures([1, 2, 3], [1, 2, 3])` должно вернуть 3, а `max_lectures([3, 2, 1], [4, 3, 2])`, должно вернуть 2.

5. Напишите функцию которая для заданной функции $f(x)$ находит решение уравнения $f(x) = 0$, если известно, что $f(-10) < 0$, а $f(10) > 0$.

11 Порядок формирования оценок по дисциплине

Накопленная оценка по дисциплине рассчитывается с помощью взвешенной суммы оценок за отдельные формы текущего контроля знаний следующим образом:

$$O_{\text{накопленная}} = 0,5O_{\text{ауд_сем}} + 0,5O_{\text{кр}}, \text{ где}$$

$O_{\text{ауд_сем}}$ - оценка знаний студента за работу **на семинарах**;

$O_{\text{кр}}$ - оценка знаний студента за **контрольную работу**;

Результирующая оценка по дисциплине (которая идет в диплом) рассчитывается следующим образом:

$$O_{\text{результ}} = 0,5O_{\text{накопленная}} + 0,5O_{\text{экза}}, \text{ где}$$

$O_{\text{накопленная}}$ - накопленная оценка по дисциплине;

$O_{\text{экза}}$ - оценка за экзамен.

В формулу для $O_{\text{результ}}$ подставляются значения $O_{\text{накопленная}}$ и $O_{\text{экза}}$, округленные до ближайшего целого значения. $O_{\text{результ}}$ округляется до ближайшего целого значения.

По усмотрению ведущего преподавателя, если это не противоречит действующим документам на момент экзамена, при получении накопленной оценки 8 баллов и более, студент может быть освобожден от экзамена. В таком случае, с согласия студента, ему выставляется результирующая оценка, равная накопленной.

Студент не получает возможность пересдать низкие результаты за домашнюю работу и/или работу на семинарских или контрольную работу, а также при пропуске соответствующих им учебных часов.

При получении неудовлетворительной оценки $O_{\text{результ}}$ (значение после округления менее 4 баллов) выставляется оценка «НЕУДОВЛЕТВОРИТЕЛЬНО».

12 Учебно-методическое и информационное обеспечение дисциплины

12.1 Основная литература

Stephens R. Essential Algorithms : A Practical Approach to Computer Algorithms [Electronic Resource] / Rod Stephens. - Somerset, US: Wiley, 2013.- 625 p. - Authorized access: <http://site.ebrary.com/lib/hselibrary> (Online Digital Library "Ebrary").

12.2 Дополнительная литература

1. Hetland M. L. Python Algorithms: mastering basic algorithms in the Python Language [Electronic Resource] / Magnus Lie Hetland. – Apress, 2010. – 337 p. - Authorized access:



<http://library.books24x7.com/toc.aspx?bookid=37959#&&assetid=37959&view=toc> (Online Digital Library "Books24x7").

2. Cormen T. H. Introduction to Algorithms [Electronic Resource] / Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. - The MIT Press, 2009. – 1313p. - Authorized access: <http://site.ebrary.com/lib/hselibrary/detail.action?docID=10397652> (Online Digital Library "Ebrary").
3. Motwani R., Raghavan P. Randomized Algorithms / [Electronic Resource] / Rajeev Motwani, Prabhakar Raghavan Cambridge University Press, 1995.- 456 p/ - Authorized access: <http://library.books24x7.com/toc.aspx?bookid=3503> (Online Digital Library "Books24x7").

12.3 Справочники, словари, энциклопедии

12.4 Ресурсы информационно-телекоммуникационной сети «Интернет»

<http://repl.it>

12.5 Программные средства

Для проведения занятий требуется компьютерный класс с установленным Python.

12.6 Информационные справочные системы

12.7 Дистанционная поддержка дисциплины

Не предусмотрена

13 Материально-техническое обеспечение дисциплины

При проведении лекционных и семинарских занятий преподавателем может использоваться компьютер и мультимедийный проектор. Практические занятия выполняются в компьютерном классе.