

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

Факультет Санкт-Петербургская школа  
физико-математических и компьютерных наук

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 программы  
Сироткин А.В., avsirotkin@hse.ru

Согласована начальником ОСУП  
«31» августа 2018г.

О.Н. Колобова \_\_\_\_\_

Утверждена Академическим руководителем образовательной программы  
«31» августа 2018г.

А.В. Сироткин \_\_\_\_\_

Санкт-Петербург, 2018

*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 образовательным результатам, 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, утвержденным ученым советом Национального исследовательского университета «Высшая школа экономики», протокол от 22.12.2017 №13  
<https://www.hse.ru/data/2018/02/09/1162000644/01.04.02%20%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%B0%D1%8F%20%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0%20%D0%B8%20%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0.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 программе «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	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
Способен анализировать, верифицировать, оценивать полноту информации в ходе профессиональной	УК-6	Способность находить и самостоятельно осваивать нужную информацию из документации,	Практические занятия, самостоятельная работа, лекции	Контрольная работа, экзамен



деятельности, при необходимости восполнять и синтезировать недостающую информацию.		литературы, сети Интернет		
Способен обоснованно выбирать и применять в профессиональной деятельности современные компьютерные технологии в соответствии со спецификой решаемых задач, включая операционные системы, сетевые технологии, языки программирования, языки манипулирования данными, электронные библиотеки, пакеты прикладных программ.	ОПК-3	Способен давать оценку сложности поставленной задачи и оценивать вычислительную сложность алгоритмов решения задачи. Способен выбрать оптимальный алгоритм.	Семинарские занятия, самостоятельная работа, лекции	Контрольная работа, аудиторная работа, экзамен
Способен анализировать и воспроизводить смысл междисциплинарных текстов с использованием языка и аппарата прикладной математики и информатики.	ПК-4	Умеет анализировать условия задач по программированию; Способен описать решение задачи используя псевдокод.	Семинарские занятия, самостоятельная работа	Контрольная работа, аудиторная работа, экзамен
Способен осуществлять целенаправленный многокритериальный поиск информации о новейших научных и технологических достижениях в сети Интернет и в других	ПК-7	Умеет производить отбор алгоритмов и программных библиотек, предлагающих решения отдельных частей задачи, с целью наилучшего решения с точки зрения	Семинарские занятия, самостоятельная работа	Контрольная работа, аудиторная работа, экзамен



источниках.		эффективности алгоритмов.		
Способен реализовывать модели и алгоритмы прикладной математики в виде компьютерных программ.	ПК-10	Может создать программу на языке Python для решения поставленной задачи с использованием указанных алгоритмов.	Семинарские занятия, самостоятельная работа	Контрольная работа, аудиторная 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

Курс рассчитан на 32 часа аудиторной нагрузки, из них 16 часов семинарских занятий. Общий 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 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.

Двунаправленный 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.

## 7 Оценочные средства

### 7.1. Формы контроля знаний студентов

Тип контроля	Форма контроля	2 модуль						Параметры
		Ноябрь			Декабрь			



Текущий	Аудиторная работа, 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

## 7.2. Критерии и шкалы оценки, примеры заданий

### 7.2.1. Текущий контроль

#### Работа на семинарах

Примеры заданий для 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  $u$ -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
```



### Критерии оценивания 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

$$O_{\text{ауд\_сем}} = 10 * \frac{\Pi}{Z},$$

где  $\Pi$  - количество правильно решенных заданий, а  $Z$  - 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**.

### Контрольная работа

Контрольная работа представляет собой набор заданий. В конце контрольной работы результаты сдаются в виде кода или псевдокода с комментариями и ответами на поставленные вопросы.

Пример задания на контрольную работу:

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.

### Критерии оценивания и шкала оценки контрольной работы

Каждый из пунктов задания домашней работы оценивается в два бала с шагом 0.5, если не указана индивидуальная стоимость задания. Полный балл (2) выставляется, если задание полностью выполнено и не содержит ошибок. 1.5 балла выставляется за задание, если в нем есть недочеты, влияющие на точность результатов, но не являющиеся существенными. 1 балл выставляется, если сделана основная часть задания, но оно не доведено до конца. 0.5 балла выставляется, если намечен правильный путь решения, но при этом могут быть ошибки в коде, не позволившие доделать задание. 0 баллов выставляется при отсутствии задания или при наличии грубых ошибок.

Итоговый балл выставляется путем суммирования баллов за отдельные задания и округлением по математическим правилам.

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

### 7.2. Итоговый контроль

Итоговый контроль проходит в форме решения задач на компьютере продолжительностью 120 минут.

Пример задания на письменный экзамен.

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  $f(10) > 0$ .

### 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**.

### 7.3. 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:

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

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

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

В формулу для  $O_{\text{результ}}$  подставляются значения  $O_{\text{накопленная}}$  и  $O_{\text{экзамен}}$ , 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».

## 8 Образовательные технологии

Основными образовательными технологиями являются: интерактивные лекции, работа в группах на практических занятиях и семинарах.

## 9 Учебно-методическое и информационное обеспечение дисциплины

### 9.1 Основная литература

1. Stephens, Rod. 2013. Essential Algorithms : A Practical Approach to Computer Algorithms (1). Somerset, US: Wiley.

### 9.2 Дополнительная литература

1. Hetland M. L. Python Algorithms: mastering basic algorithms in the Python Language. – Apress, 2010..
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to Algorithms. The MIT Press, Cambridge, second edition, 2001.
3. R. Motwani, P. Raghavan. Randomized Algorithms. Cambridge University Press, 1995.

## 10 Рекомендации для самостоятельной работы студентов.

Самостоятельная работа может рассматриваться как организационная форма обучения - система педагогических условий, обеспечивающих управление учебной деятельностью по освоению знаний и умений в области учебной деятельности без посторонней помощи. Студенту нужно четко понимать, что самостоятельная работа – не просто обязательное, а необходимое условие для получения знаний по дисциплине и развитию компетенций, необходимых в будущей профессиональной деятельности.

Самостоятельная работа проводится с целью:

- систематизации и закрепления полученных на лекциях теоретических знаний;

- углубления и расширения теоретических знаний;
- формирования умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развития познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирования самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- формирования практических (общеучебных и профессиональных) умений и навыков;
- развития исследовательских умений;
- получения навыков эффективной самостоятельной профессиональной (практической и научно-теоретической) деятельности.

В учебном процессе выделяют два вида самостоятельной работы:

- аудиторная;
- внеаудиторная.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа - планируемая учебная работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Самостоятельная работа, не предусмотренная программой учебной дисциплины, раскрывающей и конкретизирующей ее содержание, осуществляется студентом инициативно, с целью реализации собственных учебных и научных интересов.

Для более эффективного выполнения самостоятельной работы по дисциплине преподаватель рекомендует источники для работы, характеризует наиболее рациональную методику самостоятельной работы, демонстрирует ранее выполненные студентами работы и т. п.

Виды заданий для внеаудиторной самостоятельной работы, их содержание и характер могут иметь вариативный и дифференцированный характер, учитывать индивидуальные особенности студента.

Самостоятельная работа может осуществляться индивидуально или группами студентов online и на занятиях в зависимости от цели, объема, конкретной тематики самостоятельной работы, уровня сложности.

Контроль результатов внеаудиторной самостоятельной работы осуществляется в пределах времени, отведенного на обязательные учебные занятия по дисциплине на практических занятиях.

Для представления результатов самостоятельной работы рекомендуется:

Составить план выступления, в котором отразить тему, самостоятельный характер проделанной работы, главные выводы и/или предложения, их краткое обоснование и практическое значение – с тем, чтобы в течение 3 – 5 минут представить достоинства выполненного самостоятельно задания.

Подготовить иллюстративный материал в виде презентации для использования во время представления результатов самостоятельной работы в аудитории. Конкретный вариант наглядного представления результатов определяется форматом аудиторного занятия и задания преподавателя.

В качестве дополнительной самостоятельной подготовки студентам рекомендуется использовать сайт <https://www.kaggle.com>, для участия в соревнованиях по построению моделей, а так же для изучения разбора лучших решений, для завершенных соревнований.

## **11 Материально-техническое обеспечение дисциплины и информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения информационных справочных систем (при необходимости).**

Для проведения всех занятий используется проектор и компьютер для проекции слайдов.

Практические занятия проводятся в компьютерном классе. Для самостоятельной работы необходимо следующее ПО: Python

## **12 Особенности организации обучения для лиц с ограниченными возможностями здоровья**

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться следующие варианты восприятия учебной информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

1) *для лиц с нарушениями зрения:* в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

2) *для лиц с нарушениями слуха:* в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

3) *для лиц с нарушениями опорно-двигательного аппарата:* в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.