



Национальный исследовательский университет «Высшая школа экономики»
Программа дисциплины «Алгоритмы и структуры данных» для направления
01.03.02 «Прикладная математика и информатика» подготовки бакалавра

**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»
направления подготовки 01.03.02 «Прикладная математика и информатика»
бакалавра

Разработчик(и) программы
Объедков С.А., к.т.н., s.obiedkov@hse.ru

Одобрена на заседании департамента анализа данных и искусственного интеллекта
«__» _____ 2018 г.
Руководитель департамента
С.О. Кузнецов _____

Утверждена Академическим советом образовательной программы
«__» _____ 2018 г., № протокола _____

Академический руководитель образовательной программы
А.С. Конушин _____

Москва, 2018

*Настоящая программа не может быть использована другими подразделениями университета и
другими вузами без разрешения подразделения-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 2017 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

Уровни формирования компетенций:

РБ — ресурсная база, в основном теоретические и предметные основы (знания, умения);

СД – способы деятельности, составляющие практическое ядро данной компетенции;

МЦ – мотивационно-ценностная составляющая, отражает степень осознания ценности компетенции человеком и готовность ее использовать

В результате освоения дисциплины студент осваивает 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 |
|--|---------------|----------------------------------|---|---|--|
| Способен вести исследовательскую деятельность, включая анализ проблем, постановку целей и задач, выделение объекта и предмета исследования, выбор способа и методов исследования, а также оценку его каче- | УК-6 | РБ | Способность находить и самостоятельно осваивать нужную информацию из общедоступных источников; умение формулировать и объяснять доказательства теорем в устной и письменной форме; умение оптимально распределять собственное время при работе над поставленными задачами | Лекции, практические занятия, домашние задания | Практические задания, контрольная работа, экзамен |



| 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 |
|---|---------------|----------------------------------|--|---|--|
| ства | | | | | |
| Способен формализовать и алгоритмизировать поставленную задачу | ПК-4 | РБ, СД | Умение формализовать условие задачи, требующей алгоритмического решения; умение разбить задачу на подзадачи; умение проводить анализ корректности и временной сложности алгоритмов | Лекции, практические занятия, домашние задания | Практические задания, контрольная работа, экзамен |
| Способен писать, оформлять, отлаживать и оптимизировать программный код | ПК-5 | СД | Способность программировать, отлаживать и тестировать алгоритмы и структуры данных | Лекции, практические занятия, домашние задания | Практические задания, контрольная работа, экзамен |

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
- Дискретная математика

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:

- Машинное обучение
- Численные методы
- Параллельное и распределённое программирование
- Теория баз данных
- Введение в ООП
- Распределённые системы
- Машинное обучение и майнинг данных
- Комбинаторика, графы и вычислительная логика
- Автоматическая обработка текста



5 Тематический план учебной дисциплины

| № | Название раздела | Всего часов | Аудиторные часы | | | Самостоятельная работа |
|---|--|-------------|-----------------|----------|----------------------|------------------------|
| | | | Лекции | Семинары | Практические занятия | |
| 1 | Алгоритмы типа «разделяй и властвуй» и алгоритмы сортировки | 56 | 12 | | 12 | 32 |
| 2 | Динамическое программирование | 24 | 4 | | 4 | 16 |
| 3 | Основные алгоритмы на графах | 62 | 12 | | 12 | 38 |
| 4 | Жадные алгоритмы | 56 | 12 | | 12 | 32 |
| 5 | Структуры данных | 58 | 16 | | 16 | 26 |
| 6 | Потоки в сетях | 34 | 6 | | 6 | 22 |
| 7 | Конечные автоматы, регулярные выражения, контекстно-свободные грамматики | 52 | 10 | | 10 | 32 |

6 Формы контроля знаний студентов

| Тип контроля | Форма контроля | 1 год | | Параметры |
|------------------|--------------------|-------|--------|--|
| | | 2 | 4 | |
| Текущий (неделя) | Контрольная работа | 7 | 7 | Письменное решение задач, 80 минут. |
| | Домашнее задание | 1 – 8 | 1 – 11 | Решение задач в компьютерной системе и письменное решение задач с последующим устным объяснением. Задания разбиты на части, для каждой из которых указывается срок выполнения. |
| Промежуточный | Экзамен | * | | Решение задач в компьютерной системе, до 3 часов. |
| Итоговый | Экзамен | | * | Письменный или устный экзамен, до 3 часов. |

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 шкале от 0 до 10 баллов включительно.

6.2 Порядок формирования оценок по дисциплине

Текущая оценка за второй модуль рассчитывается следующим образом:

$$O_{\text{текущая}} = 0,4 \cdot O_{\text{к/р1}} + 0,6 \cdot O_{\text{дз1}}$$

Промежуточная оценка за второй модуль рассчитывается следующим образом:

$$O_{\text{промежуточная}} = 0,4 \cdot O_{\text{экзамен1}} + 0,6 \cdot O_{\text{текущая}}$$



Накопленная оценка за четвертый модуль рассчитывается следующим образом:

$$O_{\text{накопленная}} = 0,5 \cdot O_{\text{к/р2}} + 0,5 \cdot O_{\text{дз2}}.$$

Завершающая накопленная оценка рассчитывается следующим образом:

$$O_{\text{завершающая}} = 0,5 \cdot O_{\text{промежуточная}} + 0,5 \cdot O_{\text{накопленная}}.$$

Итоговая оценка за дисциплину рассчитывается следующим образом:

$$O_{\text{итоговая}} = 0,75 \cdot O_{\text{завершающая}} + 0,25 \cdot O_{\text{экзамен2}}.$$

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» и алгоритмы сортировки

Задача сортировки. Сортировка вставками: анализ корректности с использованием инварианта цикла и времени работы. Асимптотические обозначения. Циклическая сортировка.

Стратегия «Разделяй и властвуй». Сортировка слиянием. Доказательство корректности рекурсивных алгоритмов по индукции. Оценка времени работы рекурсивных алгоритмов при помощи рекуррентных соотношений: дерево рекурсии, итерационный метод, основная теорема.

Решение рекуррентных соотношений методом подстановки. Линейный алгоритм поиска k -ой порядковой статистики.

Нижние оценки для сортировки сравнениями. Сортировка подсчетом, блочная сортировка, поразрядная сортировка.

Рандомизированные алгоритмы. Лас-Вегас и Монте-Карло. Краткое введение в теорию вероятностей: вероятностные пространства, события, случайные переменные, математическое ожидание и его линейность, геометрические случайные переменные. Алгоритмы Bogosort и Quicksort.

2. Раздел 2. Динамическое программирование

Основные принципы, примеры алгоритмов: нахождение наименьшей общей подпоследовательности, задача о рюкзаке, поиск максимального независимого множества в дереве.

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 поиск путей в графах. Поиск кратчайших путей во взвешенном графе, алгоритмы Беллмана – Форда, Флойда – Уоршелла.

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

Таблицы с прямой адресацией. Хеш-таблицы. Хеш-функции. Универсальные семейства хеш-функций. Открытая адресация.

Двоичные деревья поиска. Семейства сбалансированных деревьев. Красно-черные деревья.

Амортизационный анализ. Куча Фибоначчи. Система непересекающихся множеств. Суффиксное дерево.

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. Конечные автоматы, регулярные выражения, контекстно-свободные грамматики

Конечные автоматы. Регулярные языки. Замкнутость регулярных языков по объединению. Минимизация конечного автомата. Регулярные выражения. Недетерминированные конечные автоматы. Замкнутость регулярных языков относительно регулярных операций. Построение конечного автомата по регулярному выражению. Эквивалентность регулярных выражений и конечных автоматов. Нерегулярные языки. Лемма о накачке.

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 эффективный алгоритм, решающий эту задачу. 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 содер-



жит 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. Докажите или опровергните: если $f(n) = O(g(n))$ и $g(n) = O(h(n))$, то $f(n) = O(h(n))$.
2. Докажите или опровергните: если $f_1(n) = O(g(n))$ и $f_2(n) = O(g(n))$, то $f_1(n) + f_2(n) = O(g(n))$.
3. *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.
4. Разработайте и реализуйте алгоритм, сортирующий массив из n чисел, каждое из которых равно -1 , 0 или 1 , за время $O(n)$ в худшем случае.
5. Решите рекуррентное соотношение: $T(n) = 8T(n/2) + n^3$.
6. 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.
7. Пусть есть два множества чисел S и T , реализованные в виде связанных 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 элементы (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++.