**Программа учебной дисциплины «Алгоритмы и структуры данных»**

Утверждена

Академическим советом ООП

Протокол № от «\_\_»\_\_\_\_\_20\_\_ г.

|  |  |
| --- | --- |
| Автор  | Объедков Сергей Александрович |
| Число кредитов  | 9 |
| Контактная работа (час.)  | 160 |
| Самостоятельная работа (час.)  | 182 |
| Курс  | 1 |
| Формат изучения дисциплины | без использования онлайн курса |

1. **ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

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.

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;
	+ Уметь формализовать условие задачи, требующей алгоритмического решения, разбить задачу на подзадачи;
	+ Иметь навыки реализации алгоритмов на языках Python и C++.

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

# Содержание УЧЕБНОЙ ДИСЦИПЛИНЫ

*Тема 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.

*Тема 2.* Динамическое программирование

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

*Тема 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. Двунаправленный поиск 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.

*Тема 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.

*Тема 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.

*Тема 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.

*Тема 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.

# ОЦЕНИВАНИЕ

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.

В курсе предусмотрена одна контрольная работа (решение задач в компьютерной системе, от 80 до 180 минут), два домашних задания (решение задач в компьютерной системе и письменное решение задач с последующим устным объяснением; задания разбиты на части, для каждой из которых указывается срок выполнения) и устный экзамен (3 часа).

Преподаватель оценивает работу студентов на практических занятиях: правильность решения теоретических задач, корректность и эффективность компьютерных программ, написанных студентом. Накопленная оценка по 10-ти балльной шкале за работу на семинарских и практических занятиях определяется перед промежуточным контролем — *Оаудиторная*.

Накопленная оценка за текущий контроль рассчитывается следующим образом:

*Онакопленная*= 0,2*·Ок/р +* 0,12*·Одз1 +* 0,12*·Одз2 +* 0,12*·Одз3 +* 0,12*·Одз4 +* 0,12*·Одз5* + 0,2*·Оаудиторная*.

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

Результирующая оценка за дисциплину рассчитывается следующим образом:

*Орезульт =* 0,7*·Онакопл +* 0,3*·Оэкз*.

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.

# ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ

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 2*n* 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.

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*) = Ω(*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*) = 8*T*(*n*/2) + *n*3.
		7. В некоторой программе осуществляется n последовательных вызовов операции *f*. Известно, что эта последовательность вызовов занимает время Θ(*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 элементы (но, возможно, в разном порядке). В идеале, ожидаемое (среднее) время работы алгоритма — O(min {|*S*|, |*T*|}).
1. **РЕСУРСЫ**
	1. **Основная литература**

Кормен, Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест, К. Штайн; Пер. с англ. под ред. И. В. Красикова. – 2-е изд. – М.; СПб.; Киев: Вильямс, 2005 (или более поздние издания). – 1290 с.

* 1. **Дополнительная литература**

Страуструп, Б. Язык программирования C++ / Б. Страуструп; Пер. с англ. под ред. Н. Н. Мартынова. – Спец. изд. – М.: БИНОМ, 2012. – 1135 с.

* 1. **Программное обеспечение**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
| 1. |  Microsoft Windows 7 Professional RUSMicrosoft Windows 10Microsoft Windows 8.1 Professional RUS | *Из внутренней сети университета (договор)* |
| 2. | Microsoft Office Professional Plus 2010 | *Из внутренней сети университета (договор)* |
| 3. | Интерпретатор языка Python 3.x  | *Свободно распространяемое программное обеспечение* |
| 4. | Компилятор языка C++11 (g++ 4.8) | *Свободно распространяемое программное обеспечение* |

* 1. **Профессиональные базы данных, информационные справочные системы, интернет-ресурсы (электронные образовательные ресурсы)**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
| 1. | Книги ebrary | *Из внутренней сети университета (договор)* |
| 2. | Books24x7  | *Из внутренней сети университета (договор)* |

* 1. **Материально-техническое обеспечение дисциплины**

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);

мультимедийный проектор с дистанционным управлением.

Учебные аудитории для лабораторных и самостоятельных занятий по дисциплине оснащены ­­­­­­­­­­­­­­­­­­­­­­­­ ПЭВМ с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде НИУ ВШЭ.