**Программа учебной дисциплины «Алгоритмы и структуры данных – 2»**

Утверждена

Академическим советом ООП

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

|  |  |
| --- | --- |
| Автор | Объедков Сергей Александрович |
| Число кредитов | 4 |
| Контактная работа (час.) | 64 |
| Самостоятельная работа (час.) | 88 |
| Курс | 2 |
| Формат изучения дисциплины | без использования онлайн курса |

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

Целями освоения дисциплины «Алгоритмы и структуры данных – 2» являются ознакомление студентов с основами алгоритмической теории сложности, приближенными и вероятностными методами решения труднорешаемых задач, в том числе задач, возникающих в анализе данных.

В результате освоения дисциплины студент:

умеет проводить анализ корректности и временной сложности алгоритмов; распознает класс сложности задач;

способен программировать, отлаживать и тестировать алгоритмы и структуры данных;

умеет формализовать условие задачи, требующей алгоритмического решения, сформулировать и объяснить доказательство теоремы в устной и письменной форме;

умеет разбить задачу на подзадачи, эффективно реализовать программные компоненты для отдельных подзадач и связать их воедино.

Изучение данной дисциплины базируется на следующих дисциплинах:

* + Дискретная математика
  + Основы и методология программирования
  + Алгоритмы и структуры данных

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

* + Способен учиться, приобретать новые знания, умения, в том числе в области, отличной от профессиональной;
  + Способен критически оценивать и переосмыслять накопленный опыт (собственный и чужой), рефлексировать профессиональную и социальную деятельность;
  + Способен провести теоретическую и экспериментальную оценку математического метода, алгоритма, модели данных;
  + Способен использовать объектно-ориентированный подход для разработки архитектуры программного модуля или системы;
  + Знает о наиболее важных алгоритмах и структурах данных и основных принципах их проектирования и анализа;
  + Умеет обосновывать корректность алгоритмов, проводить теоретическую и экспериментальную оценки их временной сложности;
  + Имеет навыки реализации алгоритмов на языках Python и C++.

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

* + Машинное обучение – 1
  + Методы оптимизации
  + Информационный поиск
  + Машинное обучение на больших данных
  + Дополнительные главы теории алгоритмов
  + Сложность вычислений и логика в теоретической информатике – 1
  + Основные методы анализа данных
  + Машинное обучение и майнинг данных

# Содержание УЧЕБНОЙ ДИСЦИПЛИНЫ

*Тема 1.* Основы теории вычислительной сложности

Машины Тьюринга. Классы P, NP, coNP, примеры задач из этих классов. NP-полнота. Полиномиальные сведения. Теорема Кука – Левина. NP-полные задачи: 3-SAT, Not-All-Equal-3- SAT, задачи о клике, вершинном покрытии, независимом множестве, гамильтоновом пути, максимальном разрезе, сумме подмножества, рюкзаке и др. Теорема об иерархии классов сложности по времени.

*Тема 2.* Методы решения труднорешаемых задач

Локальный поиск, градиентный спуск, алгоритм Метрополиса, имитация отжига. Рандомизированные и приближенные алгоритмы. Приближенное решение задачи о максимальном разрезе. Алгоритмы типа Монте-Карло и Лас-Вегас. Простой рандомизированный алгоритм для вычисления означивания переменных, максимизирующего число истинных дизъюнктов в 3-КНФ.

*Тема 3.* Задачи и алгоритмы анализа данных

Сегментация изображений (передний/задний план) с помощью минимального разреза. NP - трудность задачи сегментации изображений на три класса. Приближенный алгоритм сегментации изображений на несколько классов.

Кластеризация на основе минимального остовного дерева, максимизирующая минимальное межкластерное расстояние. NP-трудность кластеризации, минимизирующей максимальное внутрикластерное расстояние. Приближенный алгоритм кластеризации, минимизирующий максимальное расстояние до центра кластера.

Потоковые 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 предусмотрено домашнее задание (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 (4 часа).

Преподаватель оценивает работу студентов на семинарских и 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. Оценка по 10-ти балльной шкале за работу на семинарских и практических занятиях определяется перед промежуточным или итоговым контролем — *Оаудиторная*.

Накопленная оценка по дисциплине рассчитывается по формуле:

О*накопленная* = 0,4·*Одз1* + 0,4·*Одз2* + 0,2·*Оаудиторная*.

В диплом выставляется результирующая оценка по учебной дисциплине.

*Орезульт* = 0,6·*Онакопленная* + 0,4·*Оэкз*

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

# ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ

Примерные задания для домашней работы:

* + 1. Покажите, что класс P замкнут относительно пересечения.
    2. Дано: формула *φ*. Спрашивается, существует ли по крайней мере два означивания переменных, при которых формула *φ* истинна? Докажите, что данная задача является NP- полной.
    3. Пусть существует алгоритм *A*(*G*, *k*) с полиномиальным временем работы, позволяющий определить, есть ли в графе *G* клика размера *k*. Покажите, что в этом случае существует алгоритм *B*(*G*) с полиномиальным временем работы, находящий клику максимального размера в графе *G* (именно саму клику, а не только ее размер).
    4. Рассмотрим модель вычислений, в которой программа представлена в виде ориентированного ациклического графа, все узлы которого кроме двух выделенных помечены переменными. Один выделенный узел помечен нулем, другой — единицей; они соответствуют значениям функции, вычисляемой программой. Из каждого узла, помеченного переменной, исходит два ребра: одно помечено нулем, другое — единицей. У выделенных узлов нет исходящих ребер. Один из узлов программы — начальный. Программа получает на вход набор булевых значений для переменных, которыми помечены ее узлы. Вычисление начинается в начальном узле. При достижении узла, помеченного переменной, программа проверяет значение этой переменной: если переменная истинна, программа переходит по ребру, помеченному единицей, иначе — по ребру, помеченному нулем. Вычисление завершается при достижении выделенного узла. Программа возвращает значение, которым помечен этот узел.

Оцените (асимптотически) количество узлов, необходимое для программы, получающей на вход *n* булевых переменных и возвращающей 1, если хотя бы половина этих переменных истинна, и 0 — в противном случае. Объясните ответ.

* + 1. Приведите полиномиальный алгоритм сведения задачи о раскраске графа:

**Дано:** Неориентированный граф и число *k*.

**Вопрос:** Можно ли раскрасить вершины графа в *k* цветов так, чтобы цвета всех смежных вершин были различны?

к следующей задаче о целочисленном решении неравенств:

**Дано:** Система неравенств. Неравенства могут быть как строгие, так и нестрогие. В неравенствах могут присутствовать переменные и константы из области вещественных чисел. Также в неравенствах допустимы следующие операции: сложение, вычитание, умножение, деление, возведение в степень.

**Вопрос:** Существует ли целочисленное решение данной системы?

Докажите корректность сведения.

* + 1. Реализуйте приближенный алгоритм сегментации изображений на несколько классов.
    2. Реализуйте приближенный алгоритм кластеризации, минимизирующий максимальное расстояние до центра кластера.

Примерные вопросы и задачи для экзамена:

* + 1. Приведите формальное определение детерминированной и недетерминированной машин Тьюринга.
    2. Пусть задано конечное множество *S* и конечный набор его подмножеств *S*1, *S*2, ..., *Sk*. Для каждого множества *Si* заданы два числа *lowi* и *highi*. Требуется выяснить, существует ли подмножество *T* ⊆ *S*, такое что для каждого *i* выполняется *lowi* ≤ |*T* ⋂ *Si*| ≤ *highi*. Докажите, что эта задача является NP-трудной, сведя к ней задачу о выполнимости 3- КНФ (3SAT).
    3. Верно ли, что SAT ∈ coNP?
    4. Задача по определению того, содержит ли граф простой (без повторяющихся вершин) путь с числом ребер не менее *m* является NP-полной. Почему из этого следует, что существование алгоритма, вычисляющего самый длинный простой цикл в графе за полиномиальное время от длины входа, маловероятно?
    5. Обозначим за *I* некоторое множество предметов, имеющих целочисленные «вес» *w*(*i*) и «ценность» *v*(*i*). Пусть есть оракул, моментально отвечающий на вопрос, можно ли выбрать из множества *I* предметы, общий вес которых не выше, а общая ценность не ниже заданных. Объясните, как при помощи такого оракула можно за полиномиальное время определить максимальную общую ценность набора предметов из *I*, общий вес которых не превышает заданный максимальный вес *M*.
    6. Задача о доминирующем множестве:

DS = {⟨*G*, *k*⟩ | *G* — неориентированный граф (*V*, *E*); существует подмножество *C* ⊆ *V*, такое что |*C*| = *k* и каждая вершина в *V* \ *C* связана ребром из *E* с некоторой вершиной из *C*}.

Докажите, что DS ∈ NP, описав недетерминированную машину Тьюринга с полиномиальным временем работы, решающую DS. Докажите, что DS — NP-трудная задача, сведя к ней (за полиномиальное время) задачу о вершинном покрытии. (По графу *G* = (*V*, *E*) постройте граф *G*′ = (*V*′, *E*′), в котором *V*′ ⊆*V* ∪*E*.)

* + 1. Докажите, что существует язык, разрешимый за время *O*(*n*3) на одноленточной машине Тьюринга, но не разрешимый за время *O*(*n*) на многоленточной машине Тьюринга.
    2. Рассмотрим задачу о поиске кратчайшего маршрута в неориентированном графе: требуется обойти граф, побывав во всех его вершинах и вернувшись в исходную вершину, так, чтобы сумма весов на ребрах пройденного пути была минимально возможной. Приближенное решение, отличающееся от оптимального не более чем вдвое, можно получить, если построить минимальное остовное дерево графа и обойти его. Такое решение можно найти за время, полиномиальное от размера входных данных. Предложите модификацию описанного алгоритма, работающую за полиномиальное время и вычисляющую решение, которое никогда не хуже решения, полученного исходным алгоритмом, но зачастую лучше.

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.

Leskovec, J. Mining of Massive Datasets / Jure Leskovec, Jeffrey David Ullman, Anand Rajaraman. – Cambridge University Press, 2014 (или более поздние издания). – URL: https://library.books24x7.com/toc.aspx?bookid=74213 – Books 24x7 IT Pro Collection

* 1. **Дополнительная литература**

Arora, S. Computational Complexity: A Modern Approach / Sanjeev Arora; Boaz Barak. – Cambridge University Press, 2009 (или более поздние издания). – URL: https://library.books24x7.com/toc.aspx?bookid=31235 – Books 24x7 IT Pro Collection

Dahlhaus, E. The Complexity of Multiterminal Cuts / E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, M. Yannakakis // SIAM Journal on Computing. – 1994. – Vol.23 (4). – P. 864–894 – URL: https://search.proquest.com/docview/919651579 – ProQuest

Johnson, D.S. On generating all maximal independent sets / D.S. Johnson, M. Yannakakis, C.H. Papadimitriou // Information Processing Letters. – 1988. Vol.27 (3). – P. 119–123 – URL: https://www.sciencedirect.com/science/article/pii/0020019088900658?via%3Dihub – Science Direct

* 1. **Программное обеспечение**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
| 1. | Microsoft Windows 7 Professional RUS  Microsoft Windows 10  Microsoft 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)* |
| 3. | ProQuest | *Из внутренней сети университета (договор)* |
| 4. | Science Direct | *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.