**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Алгоритмы анализа графов

Graph Analysis Algorithms

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 3

Регистрационный номер рабочей программы: 002249

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Цель изучения дисциплины: ознакомление учащихся с алгоритмами анализа графов; изучение обучающимися основных принципов алгоритмов анализа графов и подходов к решению задач на графах; изучение основных структур данных, основанных на графах; закрепление материала путём решения теоретических и практических задач на графах.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Программа курса предназначена для обучающихся 3 курса и рассчитана на обучающихся, изучавших мат. анализ, алгебру и программирование в объеме четырёх семестров.

Максимальная эффективность программы будет обеспечена при условии, что обучающийся:

* владеет основами программирования, достаточными для программирования сложных алгоритмов.

**1.3. Перечень результатов обучения (learning outcomes)**

* Знание ключевых алгоритмов на графах и общее представление о современных тенденциях развития алгоритмов на графах.
* Понимание деталей организации алгоритмов на графах в том объёме и с той глубиной, с которыми они были предложены на лекции.
* Умение объяснять принятые ранее и принимать самостоятельные решения при программировании известных алгоритмов на графах и разработке новых алгоритмов.

Дисциплина развивает следующие компетенции:

* ОПК-1 — способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности;
* ОПК-3 — способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения;
* ОПК-4 — способен участвовать в разработке технической документации программных продуктов и программных комплексов;
* ОПК-5 — способен инсталлировать и сопровождать программное обеспечение для информационных систем и баз данных, в том числе отечественного производства;
* ПКА-1 — способен демонстрировать базовые знания математических и естественных наук, программирования и информационных технологий;
* ПКП-1 — способность проводить под научным руководством исследование на основе существующих методов в конкретной области профессиональной деятельности;
* ПКП-2 — способен решать задачи в области развития науки, техники и технологии с учетом нормативного правового регулирования в сфере интеллектуальной собственности;
* ПКП-4 — способен применять современные информационные технологии при проектировании, реализации, оценке качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях;
* ПКП-5 — способен использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов;
* ПКП-6 — способен использовать знания направлений развития компьютеров с традиционной (нетрадиционной) архитектурой; современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в профессиональной деятельности;
* ПКП-8 — способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования;
* УКБ-3 — способен понимать сущность и значение информации в развитии общества, использовать основные методы получения и работы с информацией с учетом современных технологий цифровой экономики и информационной безопасности.

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Основной формой обучения программированию являются лекционные занятия в аудитории, проводимые в активной форме, а именно: предполагающих дискуссию с преподавателем.

Общий объём активных и интерактивных форм учебных занятий составляет 4 часа.

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 6 | 30 |  | 2 |  |  |  |  |  | 2 |  |  |  | 46 |  | 28 |  | 4 | 3 |
|  | 2-100 |  | 2-100 |  |  |  |  |  | 2-100 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 30 |  | 2 |  |  |  |  |  | 2 |  |  |  | 46 |  | 28 |  |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 6 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

1. Введение в графы, простейшие алгоритмы на графах
   1. Основные определения теории графов.
   2. Способы хранения графов в памяти компьютеров.
   3. Поиск в глубину. Топологическая сортировка. Остовное дерево обхода. Классификация рёбер. Поиск цикла.
   4. Поиск в ширину. Поиск расстояния в невзвешенном графе. 0-1 BFS. 1-K BFS. Поиск в ширину на плотных графах.
2. Алгоритмы поиска кратчайших путей
   1. Алгоритм Дейкстры. Доказательство корректности алгоритма. Модификация с использованием множества или очереди с приоритетом.
   2. Алгоритм Форда-Беллмана. Доказательство корректности алгоритма. Поиск отрицательных циклов.
   3. Алгоритм Флойда-Уоршелла. Доказательство корректности алгоритма.
   4. Алгоритм Левита. SPFA.
3. Сильная связность и двусвязность.
   1. Поиск точек сочленения и мостов.
   2. Выделение компонент сильной связности и двусвязности.
4. Остовные деревья.
   1. Определения. Безопасное ребро. Лемма о безопасном ребре.
   2. Алгоритм Прима. Доказательство корректности алгоритма. Вариация на случай плотных графов. Связь с алгоритмом Дейкстры.
   3. Алгоритм Краскала. Доказательство корректности алгоритма. Наивная реализация.
   4. Система непересекающихся множеств. Эвристика сжатых путей. Ранговая эвристика. Объединение эвристик. Время работы. Применение в алгоритме Краскала.
5. Задача поиска ближайшего общего предка (LCA). Часть 1.
   1. Формулировка проблемы. Наивная реализация.
   2. Сведение задачи LCA к задаче поиска минимума на отрезке (RMQ). Популярные решения задачи RMQ. Использование sparse table для решения задачи.
6. Задача поиска ближайшего общего предка (LCA). Часть 2.
   1. Алгоритм Фарака-Колтона и Бендера.
   2. Решение задачи LCA с помощью алгоритма двоичного подъёма.
   3. Алгоритм Тарьяна.
7. NP-полные задачи в теории графов
   1. Задача коммивояжера. Задача поиска гамильтонова пути и цикла. Теорема Дирака и Оре. Решение с помощью метода динамического программирования по подмаскам. Алгоритм за O(n2n).
   2. Клика. Задача поиска числа клик в графе. Решение с помощью метода meet-in-the-middle. Алгоритм за O(n2n/2).
   3. Хроматическое число. Раскраска в 2 и 3 цвета. Задача поиска хроматического числа графа. Решение задачи методом динамического программирования по подмножествам. Алгоритм за O(2.44n).
8. Паросочетания
   1. Определения. Теорема Бержа. Алгоритм Эдмондса.
   2. Двудольные графы. Проверка на двудольность. Алгоритм Куна.
   3. Вершинное покрытие. Теорема Кёнига. Алгоритм построения минимального вершинного покрытия. Связь с максимальным множеством.
9. Потоки. Часть 1.
   1. Сеть. Поток в сети. Задача поиска потока заданного размера. Задача поиска потока максимального размера. Разрез в сети. Величина разреза в сети.
   2. Теорема Форда-Фалкерсона с доказательством. Алгоритм Форда-Фалкерсона.
   3. Алгоритм Эдмондса-Карпа. Доказательство корректности алгоритма.
10. Потоки. Часть 2.
    1. Алгоритм масштабирования. Доказательство корректности алгоритма. Улучшение алгоритма Эдмондса-Карпа.
    2. Алгоритм Диница. Доказательство корректности алгоритма.
    3. Единичные сети. Связь с задачей поиска паросочетаний.
11. Потоки. Часть 3.
    1. Задача поиска потока минимальной стоимости.
    2. Лемма о декомпозиции с доказательством. Лемма о разности потоков с доказательством.
    3. Теорема об отрицательных циклах с доказательством. Алгоритм удаления отрицательных циклов.
    4. Теорема о минимальном пути с доказательством. Алгоритм проталкивания вдоль путей минимального веса.
12. Центроидная декомпозиция.
    1. Определение. Алгоритм построения.
    2. Пример использования для решения задачи поиска количества простых путей заданной длины.
    3. Пример использования для решения задачи на раскраску вершин.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Для успешного усвоения дисциплины необходимо посещение лекционных занятий и самостоятельная работа с материалами лекций и рекомендованной литературой.

**3.1.2 Методическое обеспечение самостоятельной работы**

Основная и дополнительная литература.

**3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

Оценка за курс формируется на основе трёх составляющих:

1. Реализация алгоритмов путём решения задач на них и сдача их через автоматическую проверяющую систему. Курс подразумевает 3 отдельных набора задач, на решение каждого даётся месяц самостоятельной работы. Каждая задача оценивается либо в 1 балл (решена) или в 0 баллов (не решена). Каждую задачу можно сдавать неограниченное число раз.
2. Решение теоретических задач на применение изученных алгоритмов. Задачи по пройденной теме выкладываются после каждой лекции. Обучающиеся самостоятельно решают эти задачи и сдают в письменном виде. На решение каждой домашней работы даётся 2 недели. По каждой задаче обучающийся может разово получить комментарии. Каждая задача оценивается либо в 1 балл (решена) или в 0 баллов (не решена).
3. Устный экзамен с ответом на билет. Билет состоит из двух вопросов, на подготовку ответа на которые даётся не менее одного академического часа (при подготовке можно пользоваться литературой). После ответа на вопросы билета преподаватель вправе задать дополнительные вопросы по любой теме из списка вопросов, вынесенных на экзамен.

Каждый из пунктов оценивается независимо и варьируется в отрезке [0; 1] и зависит от общей доли сданных задач. Пусть X – оценка за практические задачи, Y – оценка за теоретические задачи, Z – оценка за экзамен. Результирующая оценка находится в диапазоне от 0 до 100 и вычисляется по формуле:

50 \* X + 30 \* Y + 20 \* Z

При этом обучающиеся могут получить до 20 дополнительных баллов за решение задач повышенной сложности.

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

*Пример практической задачи:*

Дано прямоугольное поле W×H. Некоторые клетки проходимы, через некоторые ходить нельзя. Из клетки можно ходить в соседние по ребру (слева, справа, сверху, снизу).

Нужно из клетки (x1, y1) найти любой (не обязательно кратчайший, даже не обязательно простой) путь в клетку (x2, y2).

Формат входных данных

На первой строке W, H, x1, y1, x2, y2 (1 ≤ x1, x2 ≤ W ≤ 1000, 1 ≤ y1, y2 ≤ H ≤ 1000). Далее H строк, в каждой из которых по W символов. Символ “.” означает, что клетка проходима, а символ “\*” означает, что по ней ходить нельзя.

Клетки (x1, y1) и (x2, y2) не совпадают и обе проходимы.

Формат выходных данных

Если пути не существует, выведите NO.

Иначе выведите YES и последовательность клеток (xi, yi), в которой первая совпадает с клеткой (x1, y1), а последняя с клеткой (x2, y2).

*Пример теоретической задачи:*

Дано дерево T = ⟨V, E⟩.

За O(V +E) вычислить для каждого ребра, сколько простых путей проходит через него.

*Пример списка вопросов к экзамену:*

1. Представление графа в памяти компьютера: список рёбер, матрица смежности, списки смежности. Оценки памяти и сложности.
2. Dfs: описание, реализация, оценка сложности. Выделение компонент связности. Нахождение остова связного неориентированного графа.
3. Поиск цикла в графе. Топологическая сортировка ациклического графа.
4. Bfs: описание, реализация, оценка сложности.
5. Алгоритм выделения компонент сильной связности.
6. 2-SAT.
7. Алгоритм поиска мостов и точек сочленения.
8. Алгоритм Дейкстры. Реализации за O(n2+m), O((n + m)log n).
9. Алгоритм Форда-Беллмана. Поиск цикла отрицательного веса.
10. Алгоритм Флойда. Описание динамики.
11. SPFA.
12. Лемма о безопасном ребре. Наивная реализация алгоритма Краскала.
13. Алгоритм Прима. Реализации за O(nm), O(n2+m), O((n+m)log n).
14. Система непересекающихся множеств. Эвристики. Применение к алгоритму Краскала.
15. Определение гамильтонова пути и цикла, достаточные условия. Задача коммивояжера за O(n2n).
16. Клика в графе, максимальная и наибольшая клика. Поиск числа клик за O(n2n/2)..
17. Хроматическое число графа. Раскраска графа в два и три цвета.
18. Поиск хроматического числа произвольного графа за O(4n), O(3n) и O(2.44n).
19. Основные определения. Теорема Бержа.
20. Поиск максимального паросочетания в произвольном графе.
21. Алгоритм Куна. Оценки сложности + оптимизации.
22. Теорема Кёнинга. Построение минимального контролирующего множества по максимальному паросочетанию за O(m + n).
23. Метод двоичных подъемов. Предподсчет за [O(n log n), O(log n)]︀.
24. Алгоритм Тарьяна (offline). Решение за O(n + q), где q — количество запросов.
25. Сведение к поиску позиции минимума на отрезке. Sparse Table. Решение за [O(n log n), O(1)]︀.
26. Определение, построение. Минимум на пути за [O(n log n), O(1)]︀.
27. Решение задачи поиска ближайшей окрашенной вершины за O(log n) на запросы поиска и покраски.
28. Определения, теорема Форда-Фалкерсона, алгоритм Форда-Фалкерсона, оценка сложности через величину максимального потока.
29. Алгоритм Эдмондса-Карпа. Оценка сложности.
30. Алгоритм Диница. Оценка сложности.
31. Техника масштабирования потока для алгоритма Эдмондса-Карпа. Оценка сложности.
32. Алгоритм устранения отрицательных циклов. Оценка сложности.
33. Алгоритм дополнения потока вдоль путей минимальной стоимости. Оценка сложности.

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К преподаванию дисциплины могут быть допущены преподаватели, имеющие диплом о высшем образовании по соответствующему направлению.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Специальных требований нет.

**3.3. Материально-техническое обеспечение**

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

В аудиториях, где проводятся занятия, необходимо наличие досок и средств письма на них.

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Аудитории для проведения лекционных занятий должны быть оснащены проекционной техникой и компьютером с возможностью вывода изображения на проектор.

**3.3.3 Характеристики специализированного оборудования**

Специальных требований нет.

**3.3.4 Характеристики специализированного программного обеспечения**

Специальных требований нет.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Специальных требований нет.

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы**

1. Алгоритмы: Построение и анализ: Пер. с англ. / Т. Х. Кормен [и др.]. - 2-е изд. - М.; СПБ; Киев : Издательский дом "Вильямс", 2012-2014.. - 1290 с.

**3.4.2 Список дополнительной литературы**

1. Мальцев, Анатолий Иванович. Алгоритмы и рекурсивные функции: монография / А. И. Мальцев. - 2-е изд. - М.: Наука, 1986. - 367 с.
2. Кормен, Томас. Алгоритмы: построение и анализ: учебник / Т. Кормен, Ч. Лейзерсон, Р. Ривест. - М.: МЦНМО: БИНОМ, 2004-2005. - 955 с.

**3.4.3 Перечень иных информационных источников**

Не предусмотрены

**Раздел 4. Разработчики программы**

Григорьев Семён Вячеславович, доцент кафедры информатики, s.v.grigoriev@spbu.ru.

По материалам Лопатина Андрея Сергеевича