2 курс

ПЛАН – КОНСПЕКТ
проведения лекционного, практического занятий № 11
по дисциплине ОПЦ.11 «Прикладная математика»

Раздел 2. Основы дискетной математики, математические модели в транспортных системах

Тема № 2.2: «Основы теории графов»

Лекционное занятие № 11

Практическое занятие № 11

Подготовил: преподаватель
В.Н. Борисов

Рязань
2026

Лекционное, практическое занятия № 11 «Постановка и решение задачи о коммивояжёре»
по Теме № 2.2 «Основы теории графов»

Цель занятий: изучить со студентами постановку и решение задачи о коммивояжёре

Виды занятий: классно-групповые, комбинированные (по проверке знаний, умений по пройденному материалу, по изучению и первичному закреплению нового материала).

Методы проведения занятий: доведение теоретических сведений, выполнение практических заданий.

Время проведения: 4 ч (лекционное занятие – 2ч, практическое занятие – 2 ч)

Основные вопросы:

1. Постановка и решение задачи о коммивояжёре.
2. Практическое применение полученных знаний – решение задач.

Литература:

1. Учебник: Высшая математика : учебник и практикум для среднего профессионального образования / под общей редакцией М. Б. Хрипуновой, И. И. Цыганок. — Москва : Издательство Юрайт, 2026. — 472 с. — (Профессиональное образование). — ISBN 978-5-534-01497-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/584924, п. 3.3 главы 3 Раздела I.
2. Учебник: Босова, Л. Л., Информатика. 9 класс. Базовый уровень : учебник / Л.Л. Босова, А. Ю. Босова. — М. : БИНОМ. Лаборатория знаний, 2017. — 208с. , ISBN 978-5-9963-3045-4, § 1.3 главы 1, с 21-31.
3. Учебник: Босова, Л. Л., Информатика. 11 класс. Базовый уровень : учебник / Л.Л. Босова, А. Ю. Босова. — М. : БИНОМ. Лаборатория знаний, 2017. — 208с. , ISBN 978-5-9963-3142-0, § 10-11 главы 3, с 135-152.

Примерный расчет времени:

1. Вступительная часть – 20 мин.
2. Основная часть – 60 мин.
3. Заключительная часть – 10 мин.

Вступительная часть (по каждому занятию):

Занятие начать с объявления темы занятия, основных рассматриваемых вопросов, времени изучения темы (нового материала), закрепления на практике полученных знаний, перечисления литературы, опроса по пройденному материалу.

Основная часть (теоретическая):

Первый вопрос: Постановка и решение задачи о коммивояжёре.

Постановка задачи коммивояжера

Коммивояжер (бродячий торговец) должен выйти из первого города, посетить по одному разу в неизвестном порядке города 2,3,4..n и вернуться в первый город. Расстояния между городами известны. В каком порядке следует обходить города, чтобы замкнутый путь (тур) коммивояжера был кратчайшим?

Методы решения:

1. Жадный алгоритм

Жадный алгоритм – алгоритм нахождения наикратчайшего расстояния путём выбора самого короткого, ещё не выбранного ребра, при условии, что оно не образует цикла с уже выбранными рёбрами. «Жадным» этот алгоритм назван потому, что на последних шагах приходится жестоко расплачиваться за жадность.

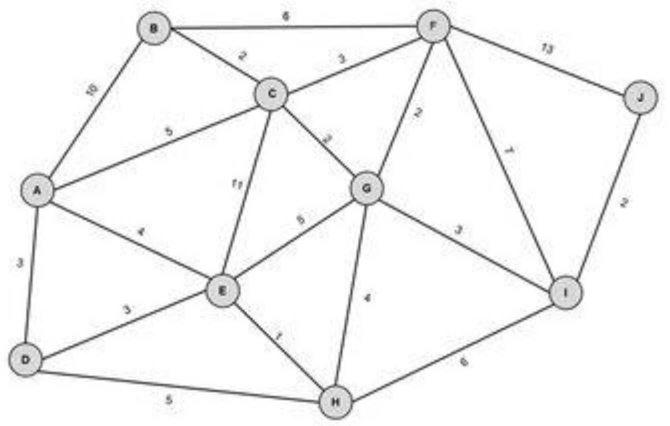
2. Полный перебор

Метод полного перебора (перебор может быть и не полным) заключается в том, что выполняется перебор всех возможных комбинаций точек (пунктов назначения). Как известно из математики, число таких перестановок равно n!, где n – количество точек. Так как в задаче коммивояжера исходный пункт обычно принимается одним и тем же (первая точка), то нам достаточно перебрать оставшиеся, т.е. количество перестановок будет равно (n–1)!. Этот алгоритм почти всегда дает точное решение задачи коммивояжера, однако продолжительность таких вычислений может занять непозволительно много времени.

3. Метод ветвей и границ

Метод, известный как метод ветвей и границ, похож на методы с отходами назад тем, что он исследует древовидную модель пространства решений и применим для широкого круга дискретных комбинаторных задач. Алгоритмы с отходами назад нацелены на то, чтобы найти одну или все конфигурации, моделируемые N-векторами, которые удовлетворяют определенным свойствам. В решаемой задаче определена числовая функция стоимости для каждой из вершин, появляющихся в дереве поиска. Цель - найти конфигурацию, на которой функция стоимости достигает максимального или минимального значения.

Задача коммивояжёра (TSP — Travelling Salesman Problem) — одна из известных задач комбинаторной оптимизации. Она заключается в поиске самого выгодного маршрута, который проходит через указанные города хотя бы по одному разу с последующим возвратом в исходный город. Критерием выгодности может быть кратчайший путь, минимальная стоимость, минимальное время и т. д..



Постановка задачи

Условия:

- **Города** — вершины графа.
- **Рёбра** — пути между городами. Вес ребра — расстояние, стоимость или время перемещения.
- **Матрица расстояний** — таблица, где строкам соответствуют города отправления, столбцам — города прибытия, а в ячейках указываются расстояния между ними.
- **Маршрут** — гамильтонов цикл (маршрут, включающий каждую вершину графа ровно один раз).

Цель — найти гамильтонов цикл минимального веса в полном взвешенном графе.

Некоторые частные случаи:

- **Геометрическая (планарна, евклидова) задача** — матрица расстояний отражает расстояния между точками на плоскости.
- **Метрическая задача** — на матрице стоимостей выполняется неравенство треугольника.
- **Симметричная и асимметричная задачи** — в зависимости от того, зависит ли стоимость от направления движения.

Оптимизационная постановка задачи относится к классу NP-трудных задач. Уже при относительно небольшом числе городов (более 66) метод перебора вариантов может не давать результатов за время, меньшее нескольких миллиардов лет.

Методы решения

1. Полный перебор (метод «грубой силы») — последовательное рассмотрение всех возможных маршрутов и выбор из них оптимального. Простой и точный метод, но при большом количестве городов его применение становится затруднительным из-за значительных затрат времени и ресурсов. ratcatcher.ru

2. Метод ветвей и границ — алгоритмический метод оптимизации, который систематически исследует возможные маршруты и обрезает ветви, которые превышают длину текущего лучшего решения, уменьшая пространство поиска. habr.com

Алгоритм метода ветвей и границ может включать следующие шаги:

- построение матрицы с исходными данными;
- нахождение минимумов по строкам и столбцам;
- редукция строк и столбцов;
- вычисление оценок нулевых клеток;
- выбор нулевой клетки с максимальной оценкой и построение ветвей решения;
- повторение шагов до нахождения полного маршрута.

3. Жадные алгоритмы — основаны на нахождении локально оптимальных решений на каждом этапе. Например, метод ближайшего соседа (Nearest Neighbor) выбирает ближайший непосещённый город на каждом шаге.

4. Эвристические и метаэвристические методы:

- **Муравьиный алгоритм (ACO)** — основан на моделировании поведения муравьёв, ищущих пути от колонии к источникам пищи.
- **Генетические алгоритмы** — используют механизмы, имитирующие естественный отбор (наследование, мутации, кроссинговер).
- **Алгоритм имитации отжига** — основан на имитации физического отжига металлов.

5. Динамическое программирование — позволяет существенно сократить объём вычислений за счёт вычисления и запоминания пройденного пути от исходного города до всех остальных.

Выбор метода зависит от конкретных условий задачи, количества городов и требуемой точности решения.

Задача: *Задача о деревенском почтальоне*

Почта расположена в деревне 1. Почтальон, выйдя из этой деревни, должен разнести корреспонденцию жителям деревней 2, 3, 4 и вернуться обратно. Под стоимостью ребра будем понимать количество километров, которые почтальон вынужден пройти пешком. Из всех замкнутых маршрутов надо выбрать такой, в котором пройденное пешком расстояние минимально.

Город	1	2	3	4
1	M	5	11	9
2	10	M	8	7
3	7	14	M	8
4	12	6	15	M

В нашем примере у нас 4 деревни и в таблице указано расстояние от каждой деревни к 3-м другим, в зависимости от направления движения (т.к. некоторые дороги могут быть с односторонним движением и т.д.).

Расстояние от деревни к этой же деревни обозначено буквой М. Также используется знак бесконечности. Это сделано для того, чтобы данный отрезок путь был условно принят за бесконечно длинный. Тогда не будет смысла выбрать движение от 1-й деревни к 1-й, от 2-й ко 2-й, и т.п. в качестве отрезка маршрута.

Находим минимальное значение в каждой строке (di) и выписываем его в отдельный столбец.

Город	1	2	3	4	di
1	M	5	11	9	5
2	10	M	8	7	7
3	7	14	M	8	7
4	12	6	15	M	6

Производим редукцию строк – из каждого элемента в строке вычитаем соответствующее значение найденного минимума (di).

Город	1	2	3	4	di
1	M	0	6	4	5
2	3	M	1	0	7
3	0	7	M	1	7
4	6	0	9	M	6

В итоге в каждой строке будет хотя бы одна нулевая клетка.

Далее находим минимальные значения в каждом столбце (dj). Эти минимумы выписываем в отдельную строку.

Город	1	2	3	4	di
1	M	0	6	4	5
2	3	M	1	0	7
3	0	7	M	1	7
4	6	0	9	M	6
dj	0	0	1	0	

Вычитаем из каждого элемента матрицы соответствующее ему dj.

Город	1	2	3	4	di
1	M	0	5	4	5
2	3	M	0	0	7
3	0	7	M	1	7
4	6	0	8	M	6
dj	0	0	1	0	

В итоге в каждом столбце будет хотя бы одна нулевая клетка.

Для каждой нулевой клетки получившейся преобразованной матрицы находим «оценку». Ею будет сумма минимального элемента по строке и минимального элемента по столбцу, в которых размещена данная нулевая клетка. Сама она при этом не учитывается. Найденные ранее di и dj не учитываются. Полученную оценку записываем рядом с нулем, в скобках.

Город	1	2	3	4
1	M	0 (4)	5	4
2	3	M	0	0
3	0	7	M	1
4	6	0	8	M

И так по всем нулевым клеткам:

Город	1	2	3	4
1	M	0 (4)	5	4
2	3	M	0 (5)	0 (1)
3	0 (4)	7	M	1
4	6	0 (6)	8	M

Редукция матрицы

Выбираем нулевую клетку с наибольшей оценкой. Заменяем ее на «М». Мы нашли один из отрезков пути. Выписываем его (от какой деревни к какой движемся, в нашем примере от 4-й ко 2-й).

Город	1	2	3	4
1	M	0 (4)	5	4
2	3	M	0 (5)	0 (1)
3	0 (4)	7	M	1
4	6	0 (6)	8	M

Ту строку и тот столбец, где образовалось две «М» полностью вычеркиваем. В клетку соответствующую обратному пути ставим еще одну букву «М» (т.к. мы уже не будем возвращаться обратно).

Город	1	2	3	4
1	M	0 (4)	5	4
2	3	M	0 (5)	M
3	0 (4)	7	M	1
4	6	M	8	M

Если полный путь еще не найден, переходим к пункту 2, если найден к пункту 9.

Если мы еще не нашли все отрезки пути, то возвращаемся ко 2-му пункту и вновь ищем минимумы по строкам и столбцам, проводим их редукцию, считаем оценки нулевых клеток и т.д.

Если все отрезки пути найдены (или найдены еще не все отрезков, но оставшаяся часть пути очевидна) – переходим к пункту 9.

Вычисление итоговой длины пути и построение маршрута:
Найдя все отрезки пути, остается только соединить их между собой и рассчитать общую длину пути (стоимость поездки по этому маршруту, затраченное время и т.д.). Длины дорог соединяющих деревни берем из самой первой таблицы с исходными данными.
В нашем примере маршрут получился следующий: 4 → 2 → 3 → 1 → 4.
Общая длина пути: L = 30.

Практическая часть:

Второй вопрос: <u>Практическое применение полученных знаний – решение задач.</u>

Решить задания, указанные преподавателем:

Задача для самостоятельной работы:

	1	2	3	4	5
1	бескон.	25	40	31	27
2	5	бескон.	17	30	25
3	19	15	бескон.	6	1
4	9	50	24	бескон.	6
5	22	8	7	10	бескон.

Заключительная часть занятия (по каждому занятию):

1. Закончить изложение материала.
2. Ответить на возникшие вопросы.
3. Подвести итоги занятия.
4. Выдать задание на самоподготовку (домашнее задание).

Задание на самоподготовку (по каждому занятию):

1. Детально проработать материал занятия, размещенный в текущем План-конспекте, приложении, в учебниках, указанных на с.2 текущего План-конспекта.
2. Решить задания, указанные преподавателем.
3. Подготовиться к опросу по пройденному материалу.