Полное построение алгоритма ч 1.

Задача коммивояжера

Содержание лекции

- Постановка задачи коммивояжера.
- Построение модели (в терминах теории графов).
- Исчерпывающий алгоритм для задачи коммивояжера.
- Оценка сложности алгоритма.
- Решение "задачи коммивояжера" методом полного перебора (исчерпывающий алгоритм).
- Отладка и документирование программ.

Полное построение алгоритма. Этапы:

- Постановка задачи.
- Построение модели решения (математической модели, аналога и т.д.).
- Разработка алгоритма.
- Проверка правильности алгоритма.
- Реализация алгоритма.
- Анализ алгоритма и его сложности.
- Проверка программы.
- Составление документации

Постановка задачи.

- Прежде, чем понять задачу её, необходимо четко сформулировать. Обычно процесс точной формулировки задачи сводится к постановке правильных вопросов.
  - Понятна ли терминология?
  - Что дано?
  - Что нужно найти?
  - Как определить решение?
  - Каких данных не хватает и все ли они нужны?
  - Являются ли какие-то данные бесполезными?
  - Какие сделаны допущения?

Постановка задачи.

- **Сформулируем постановку на примере.**
- **"Задача Коммивояжера".**
- **Агент по продаже компьютеров должен посетить 20 городов. Компания возмещает ему 50% стоимости дорожных расходов. Известна цена проезда между каждыми двумя городами. Коммивояжеру хотелось бы снизить дорожные расходы.**

Постановка задачи
Что дано?

- **Исходная информация в виде перечня городов. Известно:**
- **Количество городов**
- **Стоимость переезда из города i в город j**
- ***Комментарий: в принципе, можно сразу отметить, что дана матрица стоимостей C:***
- ***cij- стоимость переезда из i в j.***

Постановка задачи.
Что хотим найти?

- **Как снизить дорожные расходы:**
  - **найти такую последовательность объезда городов, что стоимость всего пути будет наименьшей.**
- **Необходима ли дополнительная информация?**
- **Есть ли приоритеты в городах?**

Постановка задачи

- **<u>Дополнительная информация:</u> Маршрут начинается и кончается в базовом городе и проходит по одному разу через все остальные города.**

Постановка задачи
Что надо получить?

- **Список городов, содержащий каждый город только один раз, (за исключением базового города, который стоит в списке первым и последним), который был бы оптимальным для коммивояжера.**
- **Что значит «оптимальный»?**

Постановка задачи

Что надо получить?

- ***Сумма стоимостей между каждыми двумя городами маршрута - это общая стоимость маршрута представленного списка.***
- ***Необходимо представить список наименьшей стоимости.***

Построение модели решения (математической модели, аналога и т.д.).

- Задача четко сформулирована, теперь необходимо составить для неё математическую модель. Выбор модели существенно влияет на остальные этапы решения задачи.
- Невозможно предложить набор правил, автоматизирующих этап моделирования.

Построение модели решения (математической модели, аналога и т.д.).

- Приступая к разработке модели, следует, по крайней мере, задать два основных вопроса:
  - Какие математические структуры больше всего подходят для задачи (это может сразу упростить ее и повлиять на выбор алгоритма)
  - Существует ли решенные аналогичные задачи. (На что похоже, в чем отличие)

Построение модели решения (математической модели, аналога и т.д.).

- Гаусс, Лейбниц, Эйнштейн?
- Ищем похожую задачу
- Что нужно для модели?:
  - описать на языке математики, что нам дано и что хотим найти,
  - сделать выбор математических структур,
  - переформулировать задачу необходимо в терминах соответствующих математических объектов.

Построение модели решения (математической модели, аналога и т.д.).

- Модель построена, если можно утвердительно ответить на следующие вопросы:
  - Вся ли важная информация задачи описана математическими объектами?
  - Существует ли математическая величина, ассоциируемая с искомым результатом?
  - Выявлено ли какое-нибудь полезное соотношение между объектами модели?
  - Можно работать с моделью?
  - Насколько удобно ли с ней работать?

Построение модели для задачи коммивояжера

- Решали ли мы раньше подобные задачи?
- Вероятно, нет, однако мы сталкивались с задачей выбора пути по дорожным картам или в лабиринте.
- Представим нашу задачу в виде карты:
  - Города - точки, соединенные отрезками, на которых проставлена стоимость проезда из первого города во второй. (Длины отрезков при этом роли не играют).

Построение модели для задачи коммивояжера

- Точка - город.
- расстояние между каждой парой точек, соответствующих городам i и j, - c_{ij}
- Расположим точки любым удобным способом, соединим точки *i* и *j* линиями и проставим на них «веса» c_{ij}

Построение модели для задачи коммивояжера

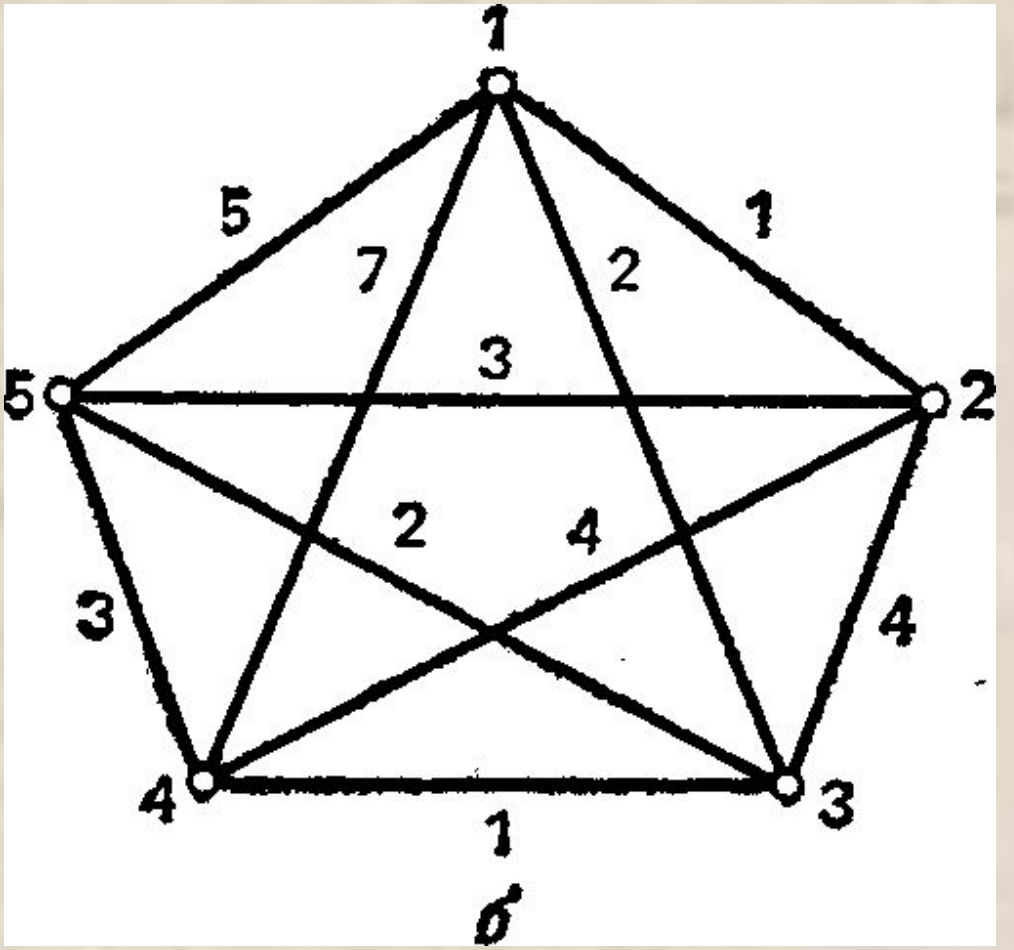


Схема - частный случай известного в математике *графа,* или *сети.*

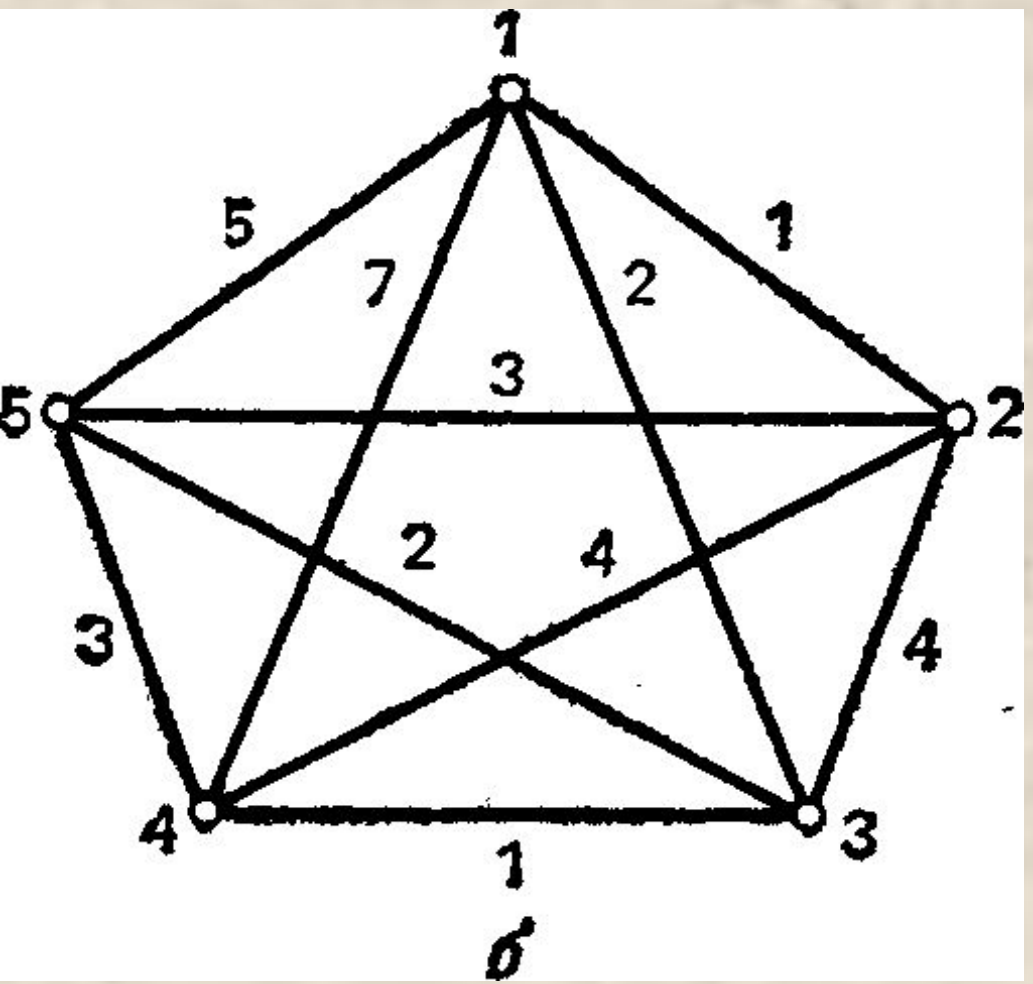
Обоснование модели

- В общем случае сеть — *это множество точек (на плоскости) вместе с линиями, соединяющими некоторые или все пары точек; над линиями могут быть проставлены веса*
- Каждый граф можно представить на плоскости множеством точек, соответствующих вершинам, которые соединены линиями, соответствующими ребрам.
- Вершины графа на рисунке выделяют обычно кружочками или квадратиками, *так как не всегда точки пересечения ребер являются вершинами графа.*

Обоснование модели. Представление графа в виде матрицы

- Для нашей задачи рассмотрим представление графа в виде матрицы стоимостей.
- Предположим, что стоимость проезда из города i в город j такая же как и из города j в город i, хотя, вообще говоря, это не всегда так.
- Как видно из примера на рис.1, в нашем случае число городов равно 5. Заполним матрицу стоимостей ***C***

Матрица стоимостей для задачи коммивояжера



б

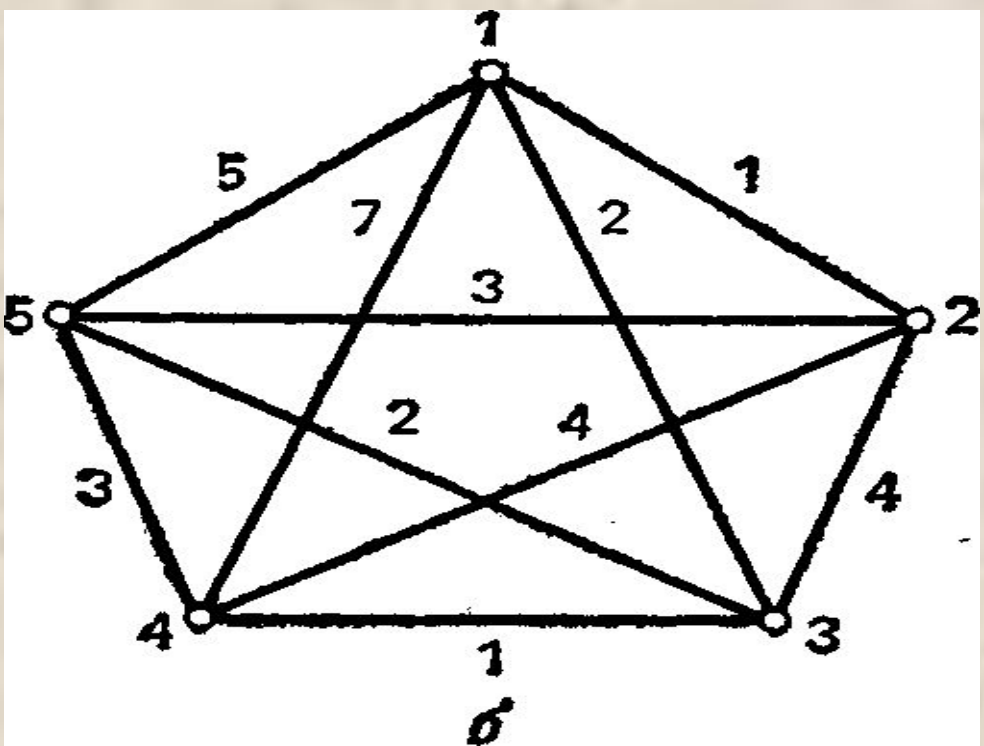
	1	2	3	4	5
1	-	1	2	7	5
2	1	-	4	4	3
3	2	4	-	1	2
4	7	4	1	-	3
5	5	3	2	3	-

Модель для задачи коммивояжера

- **Что ищем?**
- В терминах теории графов список городов определяет *замкнутый цикл, начинающийся с базового города и возвращающийся туда же после прохождения каждой вершины графа по одному разу.*
- *Такой цикл называется **гамильтоновым** циклом.*
- *Задача решена, если мы нашли гамильтонов цикл с наименьшей стоимостью.*

Модель для задачи коммивояжера

- *Например, для рассматриваемого графа гамильтонов цикл **1 – 5 – 3 – 4 – 2 – 1** имеет стоимость:*
- ***5+2+1+4+1=13***
- *Является ли он маршрутом с минимальной стоимостью? Это пока неизвестно.*



б

Разработка алгоритма.

- Выбор алгоритма зависит от выбранной модели.
- Два разных алгоритма могут быть правильными, но сильно отличаться по эффективности работы.
- Критерии эффективности различных алгоритмов и способы оценки мы рассмотрим позже, а сейчас попытаемся описать самый очевидный подход к алгоритму решения нашей задачи.

«Исчерпывающий алгоритм» решения задачи коммивояжера

- Произвольно пронумеруем города целыми числами от 1 до n. Базовому городу припишем номер n. Таким образом, каждый гамильтонов цикл однозначно соответствует перестановке целых чисел:
- **n** 1, 2, 3, … n-1, **n**
- **n** n-5, 2, 3, …, n-1, n-2 **n** и др.
- Для любой перестановки мы можем проследить гамильтонов цикл на графе, и в то же время вычислить стоимость соответствующего пути.

Исчерпывающий алгоритм (ETS):

1. Образуем перестановки первых n-1 чисел
2. Выбираем первую перестановку, строим соответствующий путь и вычисляем его стоимость. Принимаем данную стоимость за минимальную.
3. Выбираем перестановку, строим соответствующий путь и вычисляем его стоимость.
4. Сравниваем стоимость текущего пути с минимальной. Запоминаем минимальную из них. Возвращаемся к шагу 3.

Такой алгоритм называется исчерпывающим или переборным алгоритмом.

Проверка правильности алгоритма.

- Это один из наиболее трудных этапов.
- Проверка правильности алгоритма часто заменяется проверкой правильности программы, то есть прогонкой её на различных тестах.
- Если выданные программой ответы могут быть подтверждены известными или вычисленными вручную данными, возникает искушение сделать вывод, что программа работает.

Проверка правильности алгоритма.

- Для большинства алгоритмов очень сложно составить систему тестов, проверяющую все особенности, тонкости работающей программы. 3% ошибок считается нормой.
- В документации должны быть описаны ситуации возникновения ошибок (ограничения).

Методика доказательства правильности алгоритма.

Предположим, что алгоритм описан в виде последовательности шагов: от шага 1 до шага n.

1. Предложим обоснование правомерности каждого шага (выделение инварианта).
2. Проведем доказательство конечности алгоритма, при этом будут проверены все подходящие входные данные и получены все подходящие выходные данные.

Доказательство для алгоритма «задачи коммивояжера».

1. Проверяется каждый цикл.
2. При этом будет проверен и цикл с минимальной стоимостью; он будет запомнен (не потеряем).
3. Этот путь будет отброшен только в том случае, если существует путь с меньшей стоимостью.
4. Алгоритм должен закончить работу, так как число путей, которые нужно проверить, конечно: (n-1)!

Доказательство для алгоритма «задачи коммивояжера».

- Подобный метод известен как "**доказательство исчерпыванием**".
- Это самый грубый из всех методов доказательства.
- Правильность алгоритма ничего не говорит о его эффективности.
- Исчерпывающие алгоритмы редко бывают хорошими во всех отношениях.