

Операции над графами и их свойства



Постановка задачи

Цель:

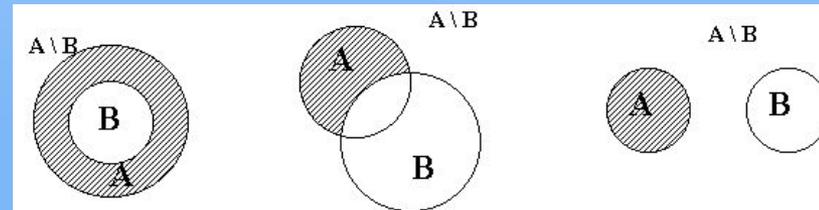
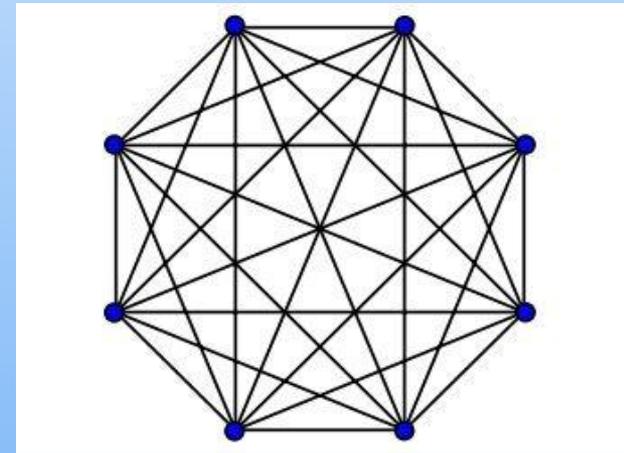
- Показать важность изучения дискретной математики на специальностях, связанных с информационными технологиями

Задачи:

- Описать функции теории графов в информационных технологиях
- Проиллюстрировать, какие основы теории графов используются в сфере информационных технологий

Дискретная математика

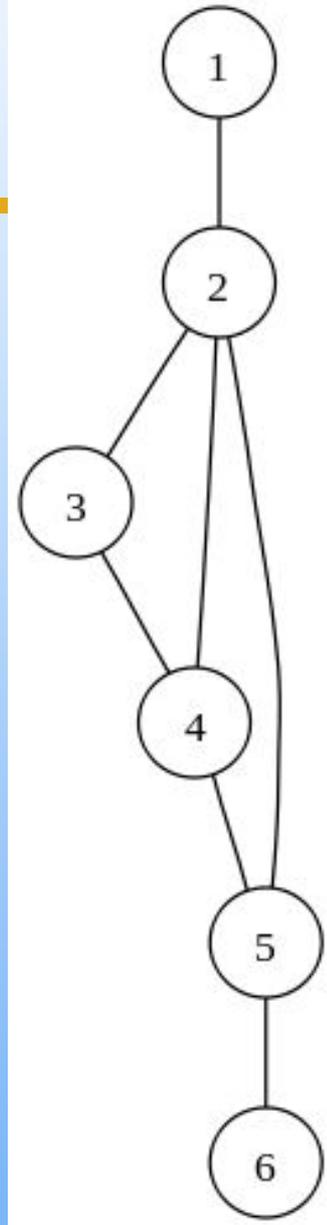
- Термин «дискретный» произошел от латинского слова *discretus* – прерывистый, состоящий из отдельных частей
- Дискретная математика изучает дискретные величины, а так же объекты, их свойства, состояния и связи между ними при помощи дискретных величин
- Разделы дискретной математики:
 - комбинаторика
 - теория чисел
 - теория множеств
 - математическая логика
 - теория алгебраических систем
 - теория графов и сетей
 - теория кодирования и т.д.



- Наиболее значимой областью применения методов дискретной математики является область компьютерных технологий.
- Дискретная математика помогает описывать данные с различной структурой и предлагает алгоритмы для их обработки, применяется при оптимизации поисковых алгоритмов в сети Интернет, конструировании баз данных, широко используется в программировании.
- Современные ученые подтверждают: подготовка специалиста в области информатики невозможна без освоения курса дискретной математики.

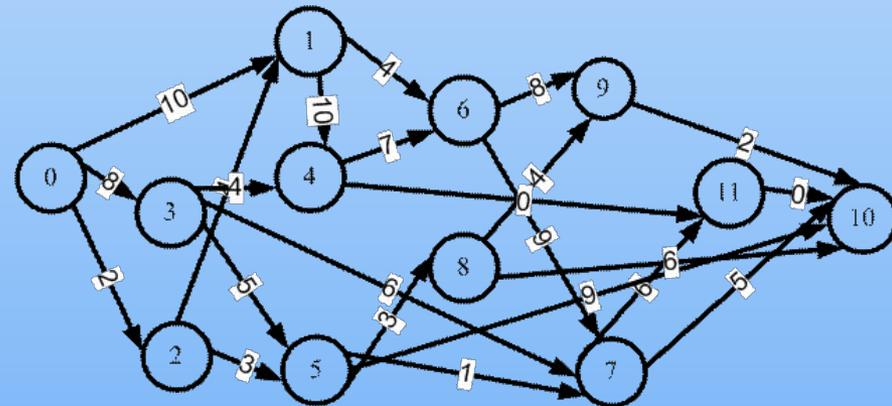
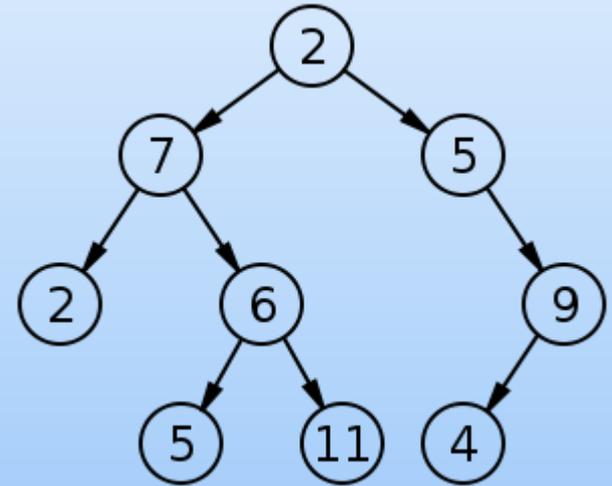
Теория графов

- **Граф** — совокупность непустого множества вершин и связей между вершинами
- Модели графов часто используются в тех случаях, когда рассматриваются системы каких-либо объектов, между которыми существуют определенные связи а также в тех случаях, когда изучается структура системы, возможности ее функционирования.
- В информатике графы используются в следующих разделах:
 - операционные системы;
 - алгоритмизация;
 - структуры данных;
 - моделирование и др.



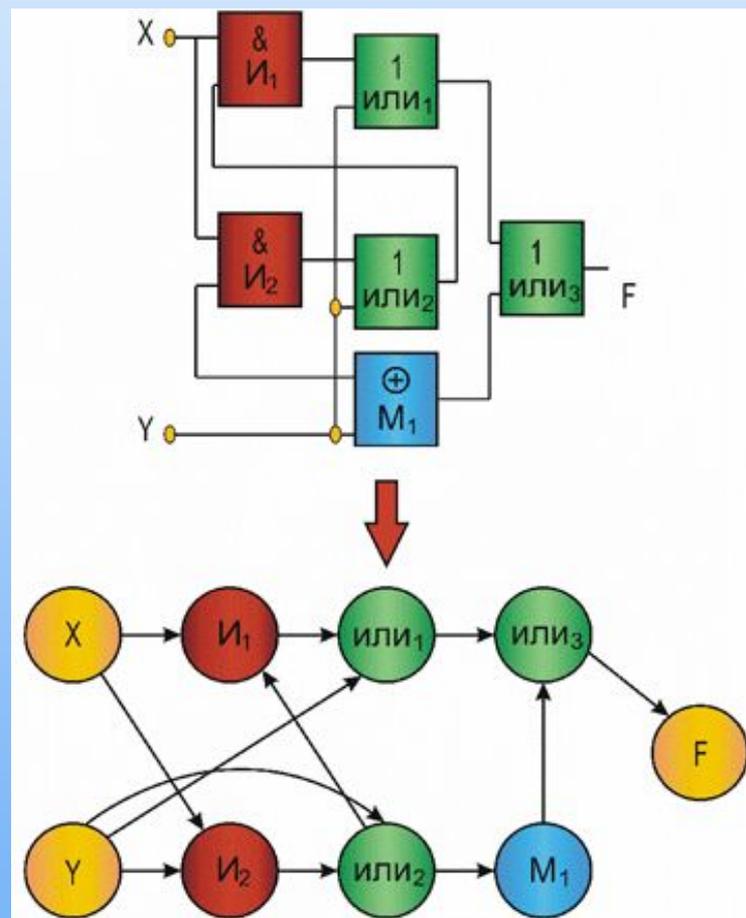
Наиболее часто в информатике используются следующие понятия о графах:

- Маршрут (путь) – упорядоченная последовательность вершин и рёбер (дуг) графа
- Граф связный, если для любых двух его вершин существует маршрут, соединяющий их.
- Дерево – связный граф, не имеющий циклов
- Сеть – связный ориентированный граф без ориентированного цикла



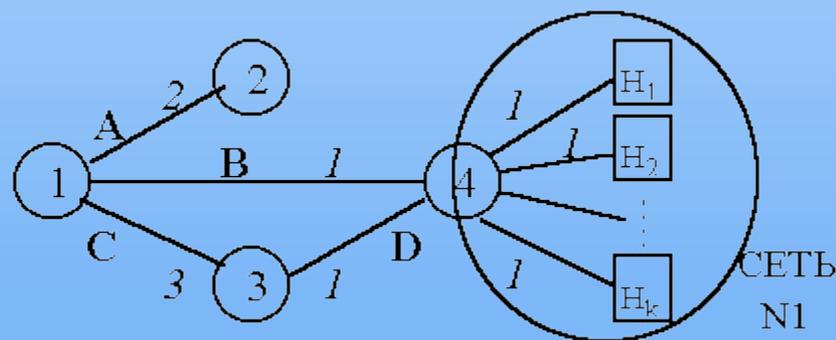
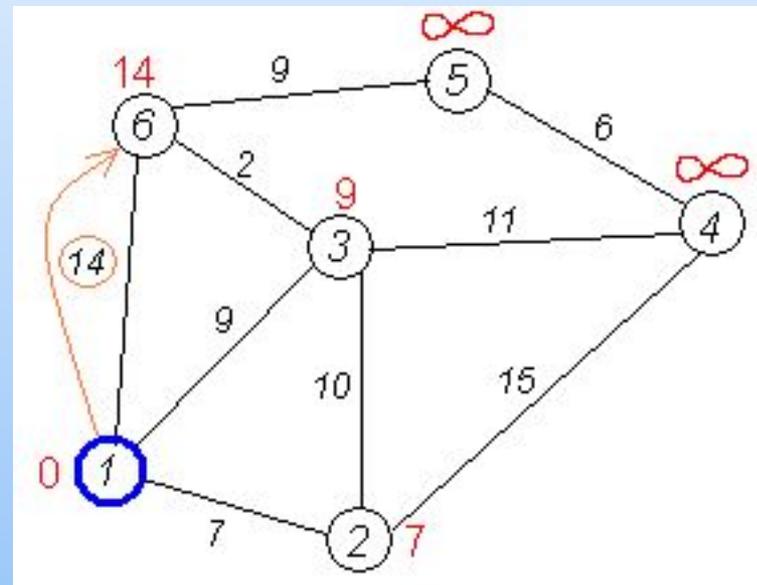
Графы в программировании

- Визуализация информации – это процесс преобразования больших и сложных видов абстрактной информации в интуитивно понятную визуальную форму. Универсальным средством такого представления структурированной информации являются графы.
- При описании большинства алгоритмов решения задачи в программировании, они визуализируются построением графов



Графы в сетевом планировании

- Решение задачи о кратчайшем пути в графе позволяет найти наиболее эффективный и удобный путь в коммуникационных системах.
- Например, для проектирования кратчайшей сети
- Оптимизации структуры ПЗУ
- Анализа надёжности сетей связи



- При помощи графа можно изобразить маршрутизацию данных в сетях
- Задача о максимальном потоке позволяет определить пропускную способность сети
- Организовать движение в сети
- Распределить интенсивность выполнения работ.

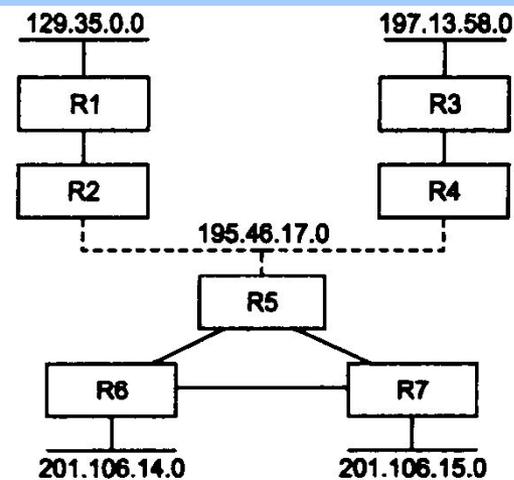
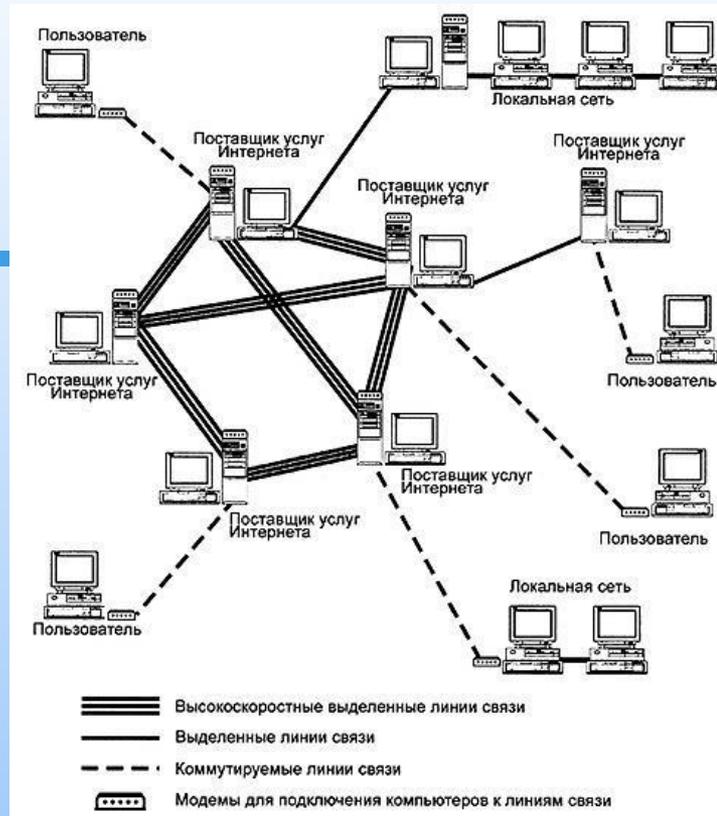
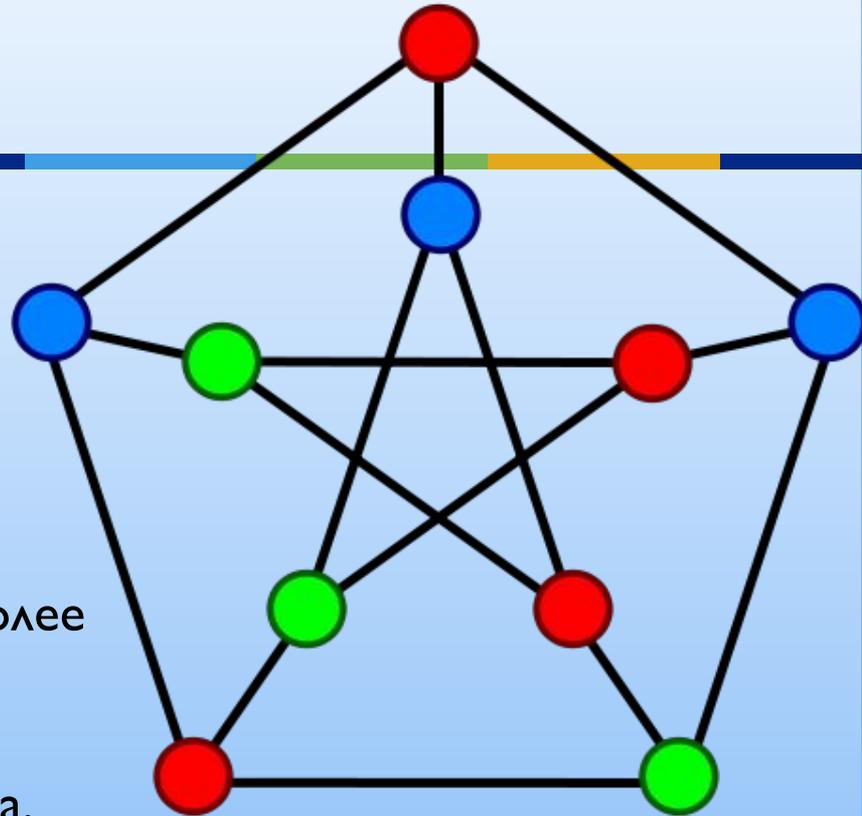


Рис. 5.27. Построение таблицы маршрутизации по протоколу OSPF

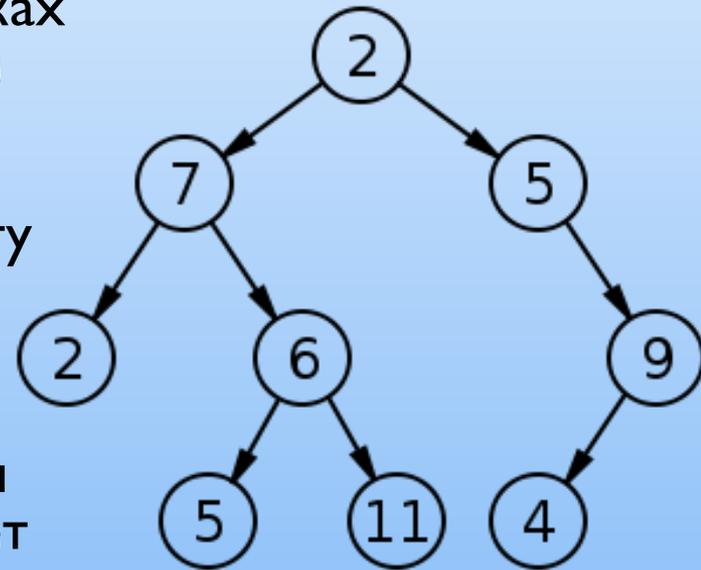
Раскраска графов

- При раскраске элементам графа ставятся в соответствие цветные метки с учетом определенных ограничений.
- Для улучшения времени выполнения результирующего кода, одной из техник компиляторной оптимизации, является распределение регистров, в которой наиболее часто используемые переменные компилируемой программы хранятся в быстродействующих регистрах процессора.
- Один из подходов к этой задаче состоит в построении модели раскраски графов. Компилятор строит граф, где вершины соответствуют регистрам, а грань соединяет две из них, если они нужны в один и тот же момент времени.



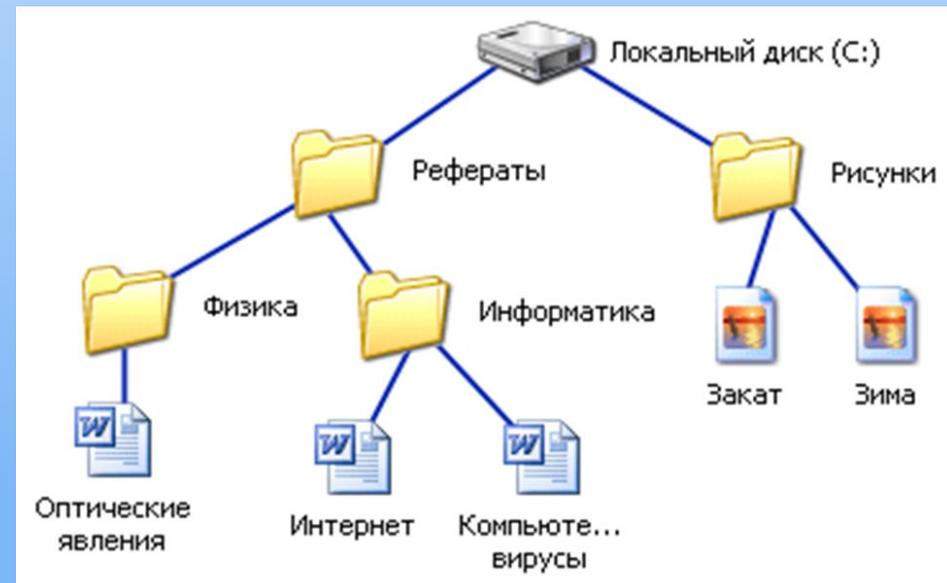
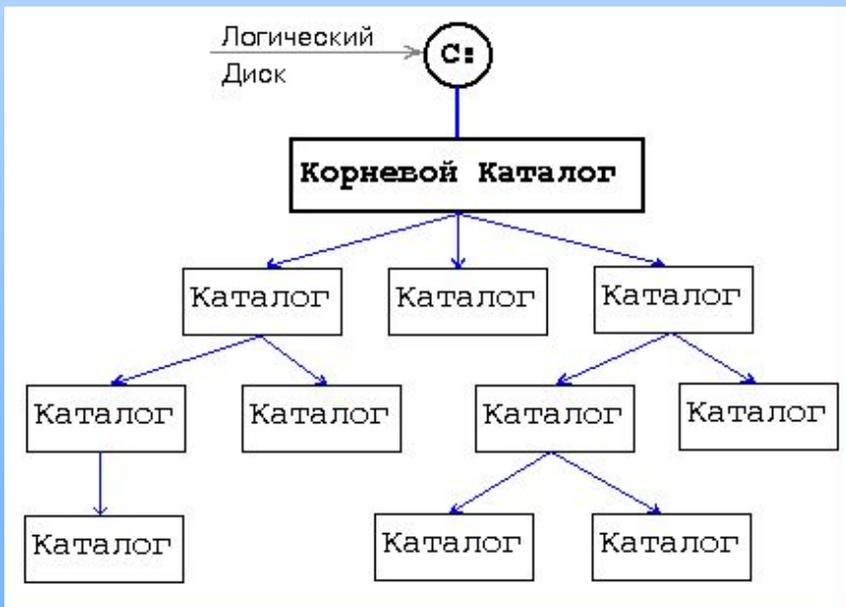
Двоичные деревья

- Двоичные деревья позволяют удобно представить нужную информацию.
- Например, интерпретация деревьев в рамках теории поиска. Каждой вершине при этом сопоставляется вопрос, ответить на который можно либо "да", либо "нет". Утвердительному и отрицательному ответу соответствуют два ребра, выходящие из вершины. "Опрос" завершается, когда удастся установить то, что требовалось.
- Таким образом, если кому-то понадобится взять интервью у различных людей, и ответ на очередной вопрос будет зависеть от заранее неизвестного ответа на предыдущий вопрос, то план такого интервью можно представить в виде двоичного дерева.



Структура дерева

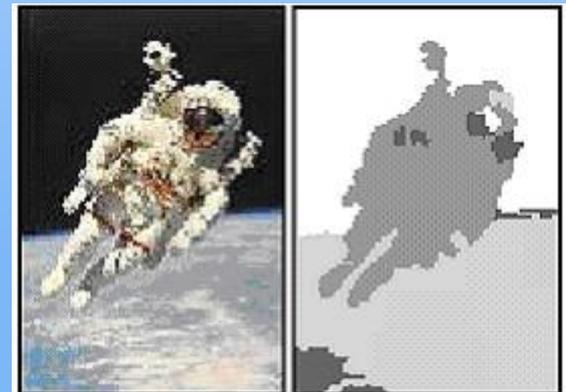
- Каталоги, папки и прочая информация в компьютере хранится в виде дерева.
- Чтобы открыть какой-то каталог, надо прописать маршрут (путь) к нему из корневого каталога.



Графы в компьютерной графике.

Сегментация изображения

- Сегментация — процесс разделения цифрового изображения на несколько сегментов. Цель сегментации заключается в упрощении и/или изменении представления изображения, чтобы его было проще и легче анализировать
- При сегментации применяются методы разреза. Изображение представляется как взвешенный неориентированный граф. Обычно пиксель или группа пикселей ассоциируется вершиной, а веса рёбер определяют схожесть соседних пикселей. Затем граф разрезается согласно заданному критерию. Каждая получаемая часть вершин получаемая считается объектом на изображении.



ОПЕРАЦИИ НАД ГРАФАМИ

▣ Одноместные операции

1. **Удаление ребра графа** — при этом все вершины графа сохраняются
2. **Добавление ребра графа** между двумя существующими вершинами.
3. **Удаление вершины** (вместе с инцидентными ребрами).
4. **Добавление вершины** (которую можно соединить с некоторыми вершинами графа).
5. **Стягивание ребра** — отождествление пары вершин, т.е. удаление пары смежных вершин, и добавление новой вершины, смежной с теми вершинами, которые были смежны, хотя бы одной из удаленных вершин)
6. **Подразбиение ребра с-** удаление ребра и добавление новой вершины, которая соединяется ребром с каждой из вершин удаленного ребра.

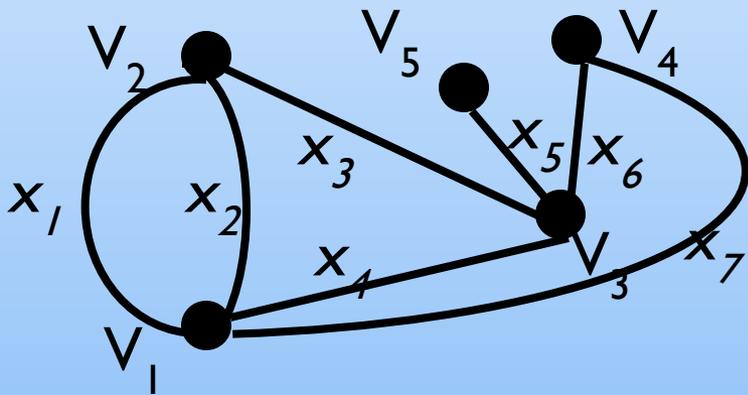
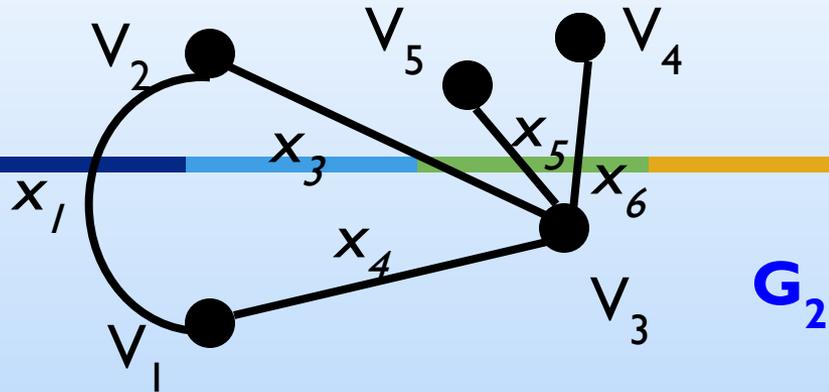
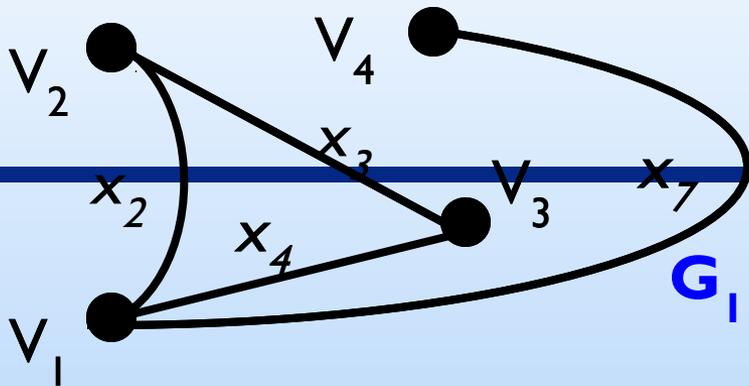
ОПЕРАЦИИ НАД ГРАФАМИ

▣ Двуместные операции

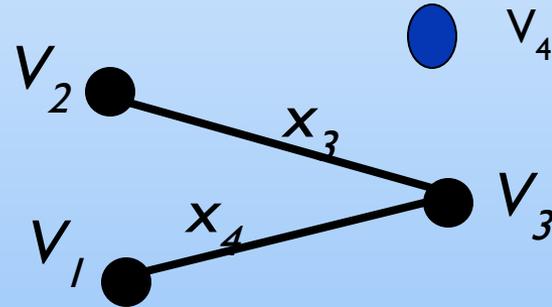
Объединением графов $G_1 = (V_1, X_1)$ и $G_2 = (V_2, X_2)$ называется граф $G = G_1 \boxplus G_2$, множество вершин которого $V = V_1 \boxplus V_2$, а множество рёбер $X = X_1 \boxplus X_2$.

Пересечением графов G_1 и G_2 называется граф $G = G_1 \boxtimes G_2$, для которого множество вершин $V = V_1 \boxtimes V_2$ и множество рёбер $X = X_1 \boxtimes X_2$.

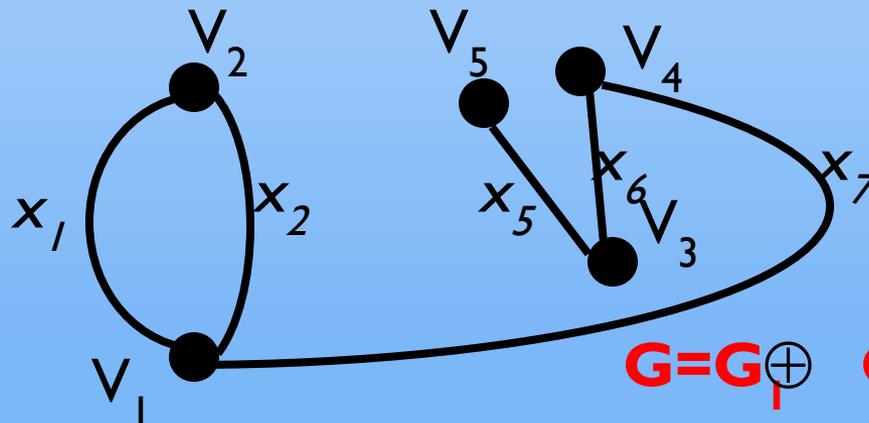
Кольцевой суммой двух графов называется граф $G = G_1 \oplus G_2$, порождённый множеством вершин $V = V_1 \oplus V_2$ и множеством рёбер $X = (X_1 \oplus X_2) \setminus (X_1 \boxtimes X_2)$, т.е. множеством рёбер, содержащихся либо в G_1 , либо в G_2 , но не в $G_1 \boxtimes G_2$.



$$G = G_1 \cup G_2$$



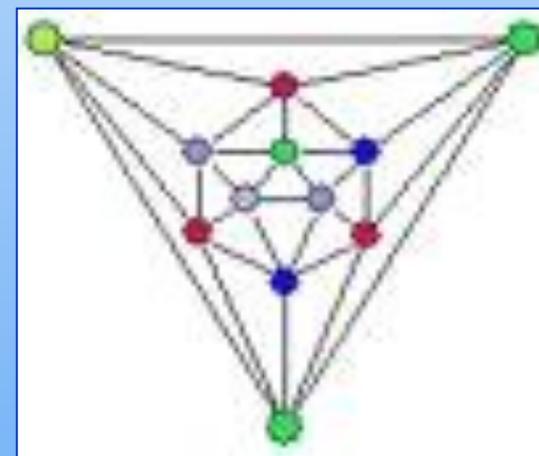
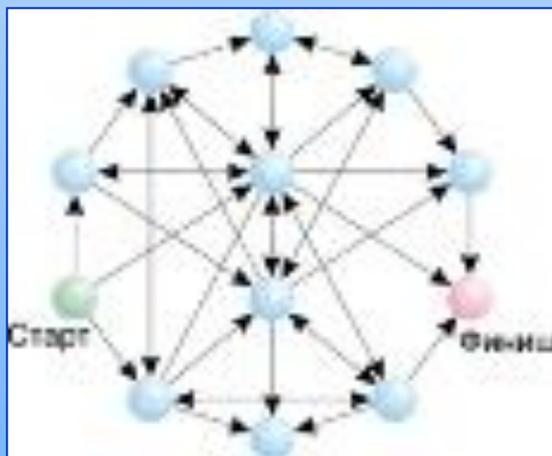
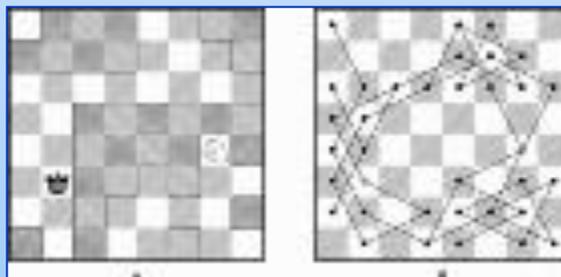
$$G = G_1 \cap G_2$$



$$G = G_1 \oplus G_2$$

ПРИМЕНЕНИЕ ГРАФОВ

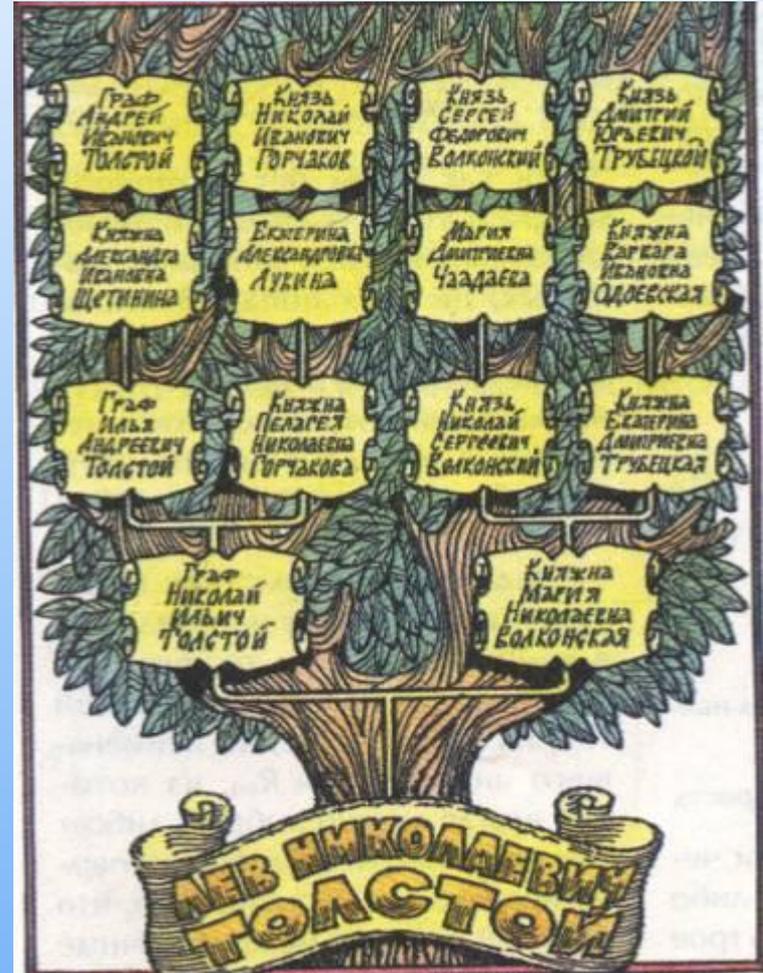
С помощью графов упрощается решение математических задач, головоломок, задач на смекалку.



ПРИМЕНЕНИЕ ГРАФОВ

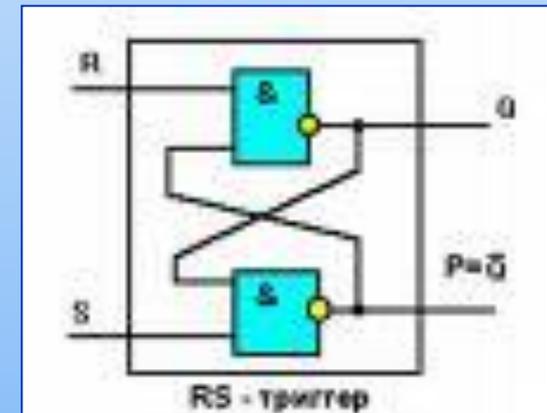
Использует графы и дворянство.

На рисунке приведена часть генеалогического дерева знаменитого дворянского рода Л. Н. Толстого. Здесь его вершины — члены этого рода, а связывающие их отрезки — отношения родственности, ведущие от родителей к детям.



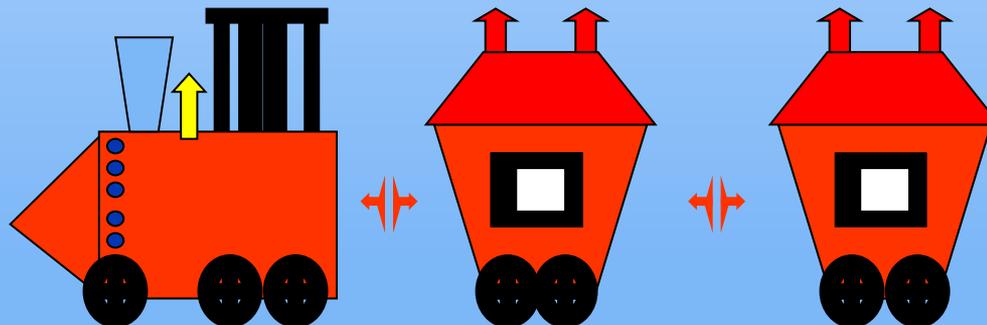
ПРИМЕНЕНИЕ ГРАФОВ

Графами являются блок – схемы программ для ЭВМ.



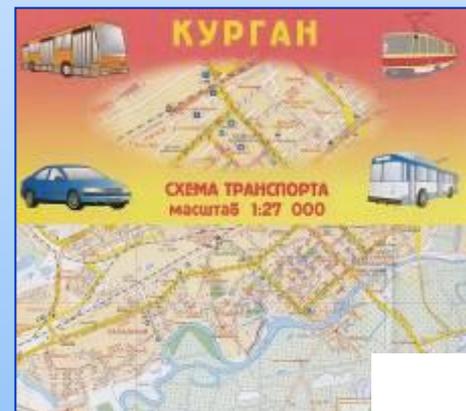
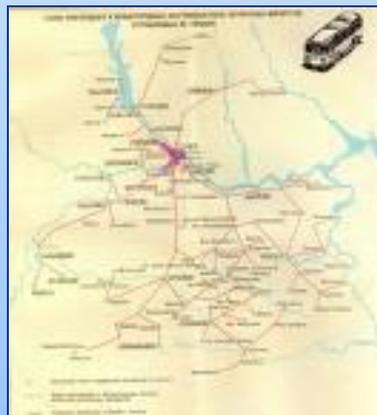
ПРИМЕНЕНИЕ ГРАФОВ

Типичными графами на географических картах являются изображения железных дорог.



ПРИМЕНЕНИЕ ГРАФОВ

Типичными графами на картах города являются схемы движения городского транспорта.



Вывод

- Теория графов позволяет упростить решение многих задач в сфере компьютерных технологий
- Благодаря графам можно наглядно проиллюстрировать многие процессы в компьютере и лучше понять их
- Изучение теории графов, как и всей дискретной математики очень важно для студентов, обучающихся на компьютерных специальностях

Источники информации

- http://www.0zd.ru/programmirovanie_kompyutery_i/primeneniye_teorii_grafov_v_informatike.html
- <http://bourabai.ru/dm/graph.htm>
- <https://ru.wikipedia.org/wiki/>
- **Касьянов В. Н., Евстигнеев В. А. Графы в программировании: обработка, визуализация и применение.**