

Графы

Вершины

- Точки
- Объекты
- Субъекты
- Количество вершин — порядок графа

Рёбра

- Связи между субъектами
- Дороги между объектами
- Количество рёбер — размер графа

Смежные вершины

- есть ребро, их соединяющее

Кратные рёбра

- Соединяют те же вершины

Петля

- Ребро, концы которого совпадают

Степень вершины

- Количество рёбер

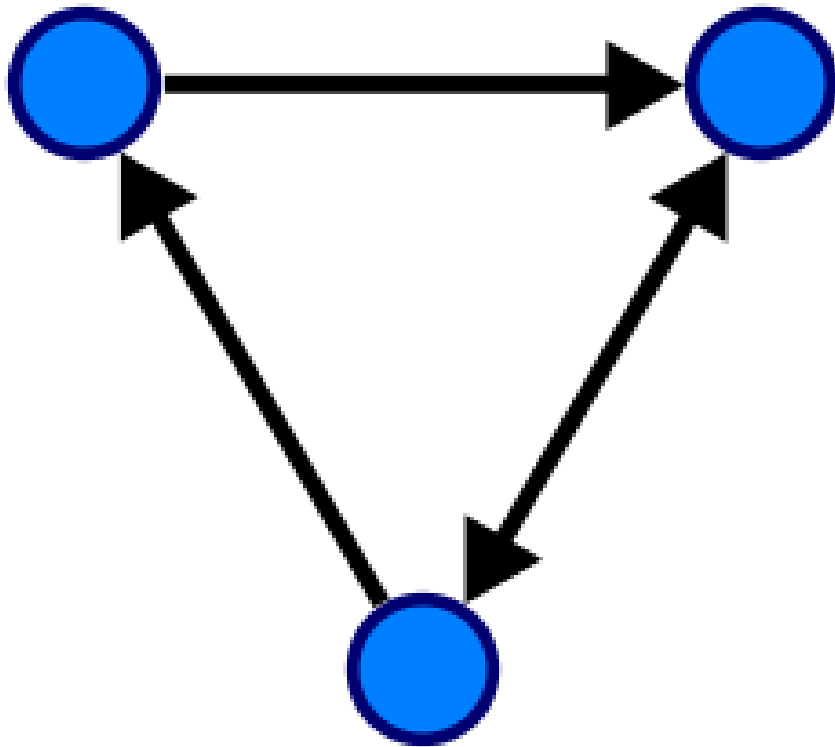
Изолированная вершина

- Рёбер нет

Висячая вершина

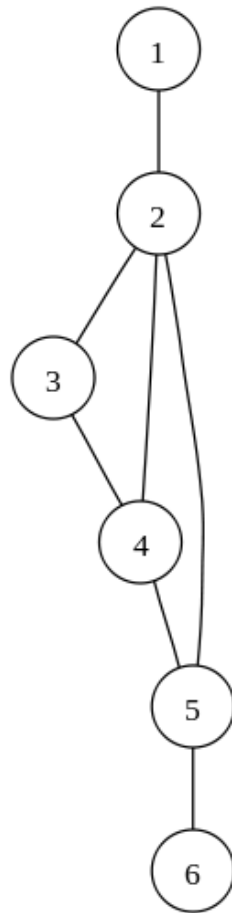
- Ребро единственное
- Она же - лист

Ориентированное ребро



- Явно выбрано начало ребра
- Другая вершина — конец ребра

Маршрут



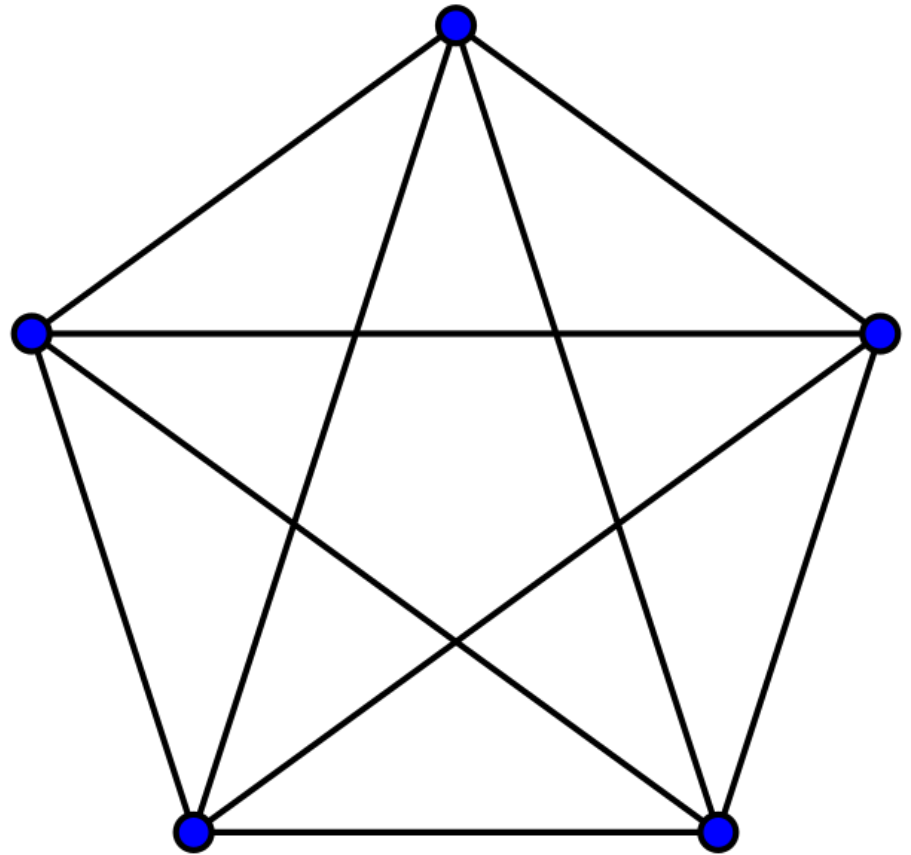
- Вершины последовательно соединены рёбрами

Связный граф

- Между любыми вершинами есть маршрут

Полный граф

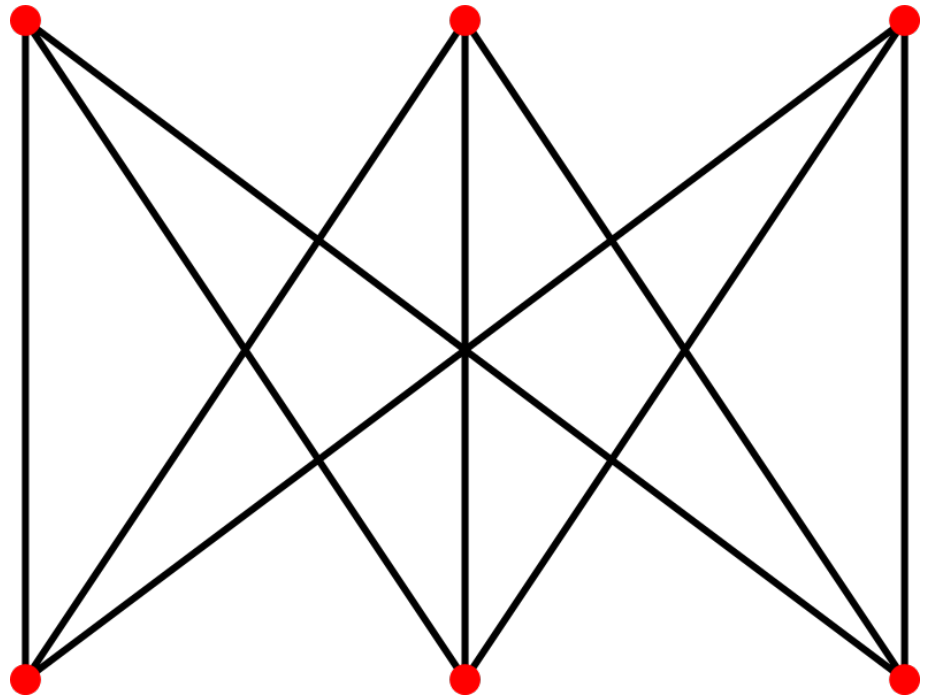
- Любые 2 вершины соединены ребром



Планарный граф

- Можно расположить на плоскости так, что рёбра не пересекаются

Пример непланарного графа



Как описать граф?

- Список вершин
- Список рёбер

Матрица смежности

- $a[i][j]$ — количество рёбер из i в j

Матрица инцидентности

- $a[i][j]$ — из i вершины выходит ребро j ?
- Да (1)
- Нет (0)
- Наоборот, входит (-1)

Цикл

- Замкнутый маршрут

Гамильтонов цикл

- Проходит через каждую вершину графа ровно 1 раз

Эйлеров цикл

- Проходит через каждое ребро графа ровно 1 раз

Простой цикл

- Ребра не повторяются

Эйлеров цикл

- Объединение всех простых циклов графа

Топологическая сортировка

- Другая интерпретация задачи — задания, зависящие друг от друга. Для каждого задания A дан список заданий, которые должны быть выполнены перед тем, как можно будет приступить к выполнению задания A . Необходимо упорядочить задания в том порядке, в котором они могут быть выполнены.
- Множество зависимых заданий естественно представлять как ориентированный граф, в котором вершины — задания, а ребро из A в B означает, что задание B должно идти раньше A .
- Нужного порядка может не быть (ориентированный граф с циклами) или может существовать несколько подходящих порядков.

Решение через DFS

```
function DFS(v):  
    v.color = GREY;  
    foreach w in ADJ(v):  
        next if w.color == BLACK  
        if(w.color == GRAY):  
            exit ( "Graph contains cycles. Topological sort inaplicable.");  
        DFS(w)  
    v.color = BLACK  
    print v
```

Package Manager

- Пакет X зависит от пакета Y версии $>1.2.3$ и пакета Z версии $= 4.5.6$
- Пакет X конфликтует с пакетом A
- Надо поставить X. Как это сделать в работающей системе?