Теория графов

Ю.Б.Буркатовская

1. Основные понятия теории графов

Определение графов и родственных объектов

Смежность вершин и ребер

Подграфы

Типы графов

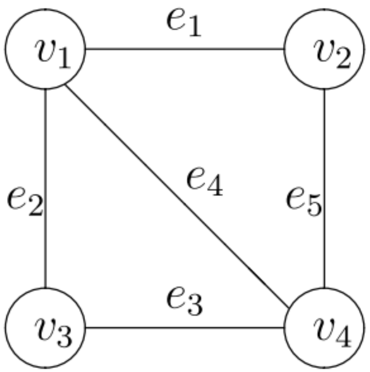
Изоморфизм графов

Операции над графами

Способы задания графов

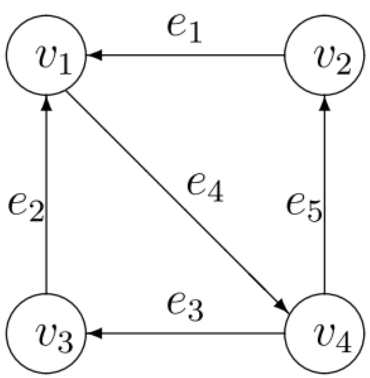
1.1. Определение графов и родственных объектов

- **Определение.** *Простым* графом $G(V, E)$ называется совокупность двух множеств – непустого множества V и множества E неупорядоченных пар различных элементов множества V. Множество V называется *множеством вершин*, множество E называется *множеством ребер*.
- **Обозначения:** p – число вершин, q – число ребер.
- **Пример.** $V = \{v_1, v_2, v_3, v_4\}$, $E = \{e_1, e_2, e_3, e_4, e_5\}$, $e_1 = v_1v_2$, $e_2 = v_3v_1$, $e_3 = v_4v_3$, $e_4 = v_1v_4$, $e_5 = v_4v_2$.



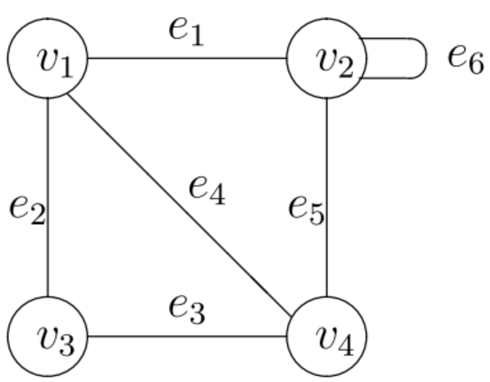
Определение графов и родственных объектов

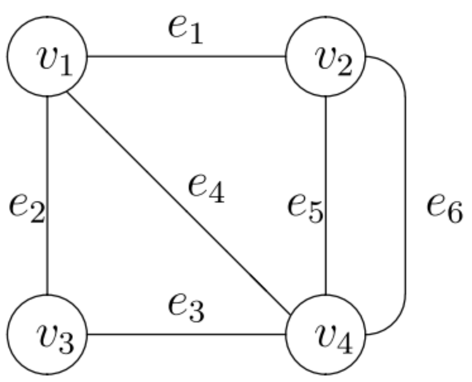
- **Определение.** Если элементами множества E являются *упорядоченные пары* (т.е. пары, в которых фиксирован порядок элементов), то граф называется *ориентированным* (или *орграфом*). В этом случае элементы множества V называются *узлами,* а элементы множества E – *дугами*. Первую вершину упорядоченной пары называют *началом дуги,* вторую – *концом*.
- **Пример.** $V = \{v_1, v_2, v_3, v_4\}$, $E = \{e_1, e_2, e_3, e_4, e_5\}$, $e_1 = v_1v_2$, $e_2 = v_3v_1$, $e_3 = v_4v_3$, $e_4 = v_1v_4$, $e_5 = v_4v_2$.



Определение графов и родственных объектов

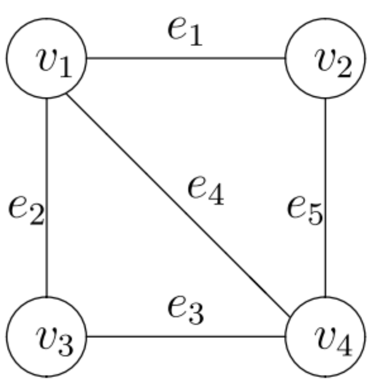
- **Определение.** Пара одинаковых элементов V вида vv называется *петлей*. Граф с петлями называется *псевдографом*.
- **Пример**. Добавим к графу из первого примера петлю $e_6 = v_2v_2$.
- **Определение.** Если E не множество, а *семейство*, то есть если E содержит одинаковые элементы, то такие элементы называются **кратными ребрами**, а граф называется *мультиграфом*.
- **Пример**. Добавим к графу из первого примера ребро $e_6 = v_4v_2$. Теперь ребра e_5, e_6 – кратные.





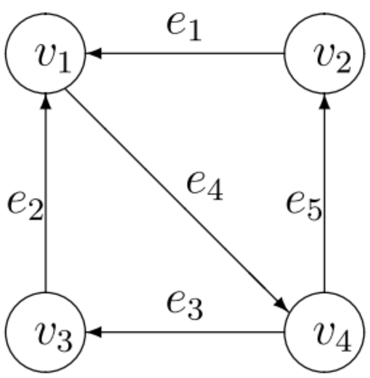
1.2. Смежность вершин и ребер

- **Определение.** Пусть v_1, v_2 – вершины, $e = v_1 v_2$ – соединяющее их ребро. Тогда вершина v_1 и ребро e инцидентны, вершина v_2 и ребро e также инцидентны. Два ребра, инцидентные одной вершине, называются *смежными*, две вершины, инцидентные одному ребру, также называются *смежными*.
- **Определение.** Множество вершин, смежных с вершиной v, называется *множеством смежности* вершины v и обозначается $\Gamma(v) = \{u : vu \in E\}$. Если $A \subset V$ – множество вершин, то $\Gamma(A)$ – множество всех вершин, смежных с вершинами из A : $\Gamma(A) = \bigcup_{v \in A} \Gamma(v)$.
- **Пример.**
- Вершины v_1, v_2 смежны
- Вершины v_2, v_3 не смежны
- Ребра $v_1 v_2$ и $v_1 v_3$ смежны
- $\Gamma(v_1) = \{v_2, v_3, v_4\}$



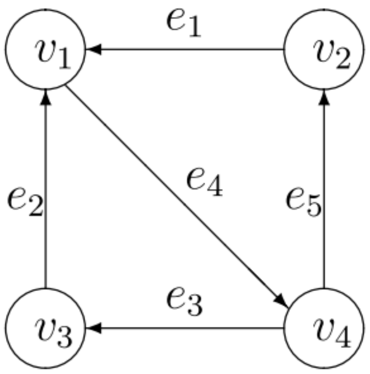
Смежность вершин и ребер

- **Пример.**
- Вершины v_1, v_2 смежны
- Вершины v_2, v_3 не смежны
- Ребра $v_2 v_1$ и $v_3 v_1$ смежны
- $\Gamma(v_1) = \{ v_4\}$
- $\Gamma(v_2) = \{ v_1\}$
- $\Gamma(v_3) = \{ v_1\}$
- $\Gamma(v_4) = \{ v_2, v_3\}$



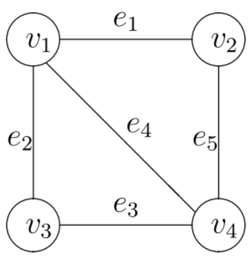
Смежность вершин и ребер

- **Пример.**
- Вершины v_1, v_2 смежны
- Вершины v_2, v_3 не смежны
- Ребра $v_2 v_1$ и $v_3 v_1$ смежны
- $\Gamma(v_1) = \{ v_4\}$
- $\Gamma(v_2) = \{ v_1\}$
- $\Gamma(v_3) = \{ v_1\}$
- $\Gamma(v_4) = \{ v_2, v_3\}$

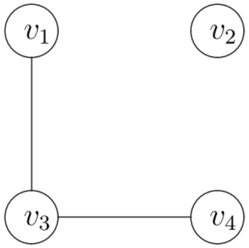


Смежность вершин и ребер

- **Определение.** Количество ребер, инцидентных вершине v, называется *степенью* (или *валентностью*) вершины v и обозначается $d(v)$.
- **Пример**. $d(v_1)$= $d(v_4)$=3, $d(v_2)$= $d(v_3)$=2

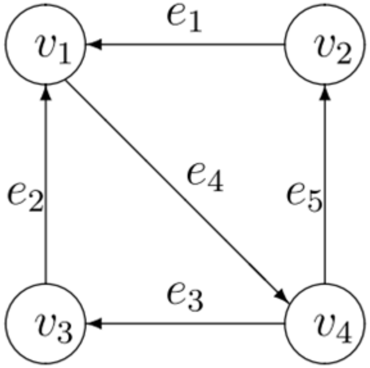


- **Определение.** Если степень вершины равна 0, то вершина называется *изолированной*. Если степень вершины равна 1, то вершина называется *висячей*.
- **Пример**. В данном графе v_1 и v_4 – висячие вершины, v_2– изолированная.



Смежность вершин и ребер

- **Определение.** Для орграфа число дуг, исходящих из вершины v, называется *полустепенью исхода* $d^{+}(v)$, а входящих – *полустепенью захода* $d^{-}(v)$.
- **Пример**. $d^{+}(v_1)$=1, $d^{-}(v_1) = 2$.



Смежность вершин и ребер

- **Лемма (Эйлера).** Сумма степеней вершин графа равна удвоенному количеству ребер

$$\sum_{v \in V} d(v) = 2q, \sum_{v \in V} (d^{+}(v) + d^{-}(v)) = 2q$$

- Эта лемма также называется «лемма о рукопожатиях» и имеет следующую интерпретацию: если произошло q рукопожатий, то всего участвующие в этом люди пожали руки $2q$ раз.
- **Теорема о числе вершин нечетной степени.** Число вершин нечетной степени в графе четно.

1.3. Подграфы

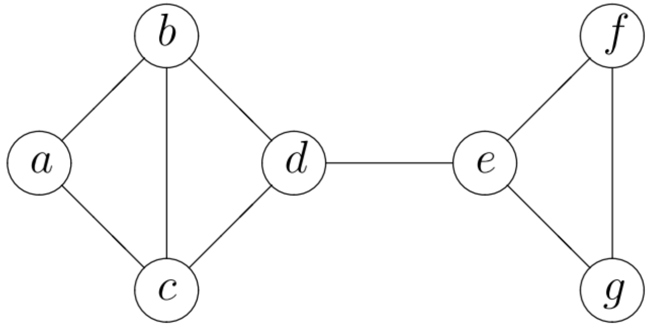
- **Определение.** Граф $G_0(V_0, E_0)$ называется *подграфом* графа $G(V,E)$ (обозначается $G_0 \subseteq G$), если $V_0 \subseteq V$ и $E_0 \subseteq E$.
- **Определение.** Если $V_0 = V$, то подграф $G_0(V_0, E_0)$ называется *остовным* подграфом графа $G(V,E)$.
- **Пример.** Слева – исходный граф, справа – его остовный подграф.

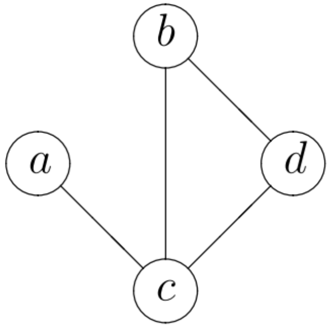
Подграфы

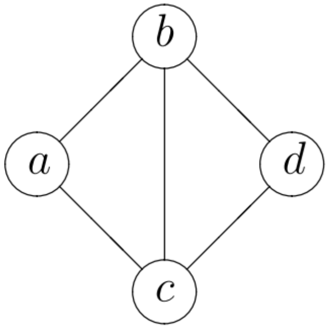
- **Определение.** Если $V_0 \neq V$, $E_0 \neq E$ то подграф $G_0(V_0, E_0)$ называется *собственным* подграфом графа $G(V, E)$.
- **Определение.** Подграф $G_0(V_0, E_0)$ называется *правильным* подграфом графа $G(V, E)$б если он содержит все возможные ребра исходного графа:

$$\forall\, u, v \in V_0 : uv \in E \Rightarrow uv \in E_0.$$

- **Пример.** Слева – исходный граф, справа – его собственные подграфы (правильный и неправильный).

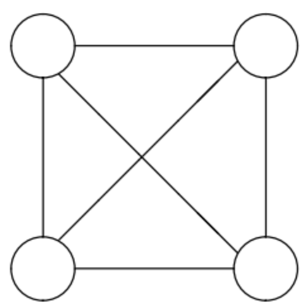




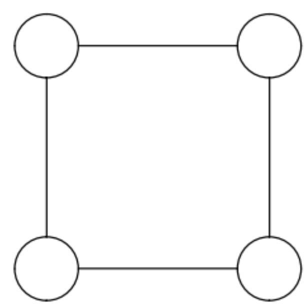


1.4. Типы графов

- **Определение.** Граф $G(V,E)$ называется *тривиальным*, если он состоит из одной вершины: $p = 1$, $E = \emptyset$.
- **Определение.** Граф $G(V,E)$ называется *пустым*, если $E = \emptyset$.
- **Определение.** Граф $G(V,E)$ называется *полным*, если в нем любые две вершины смежны, то есть $\forall\, u, v \in V: uv \in E$ (обозначение K_p).
- **Определение.** Если степени всех вершин равны k, то граф называется *регулярным степени k*.
- **Примеры.**



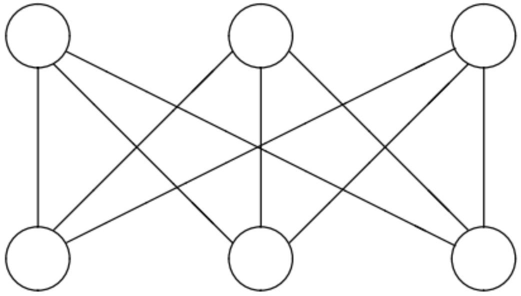
Полный граф K_4



Регулярный граф степени 2

Типы графов

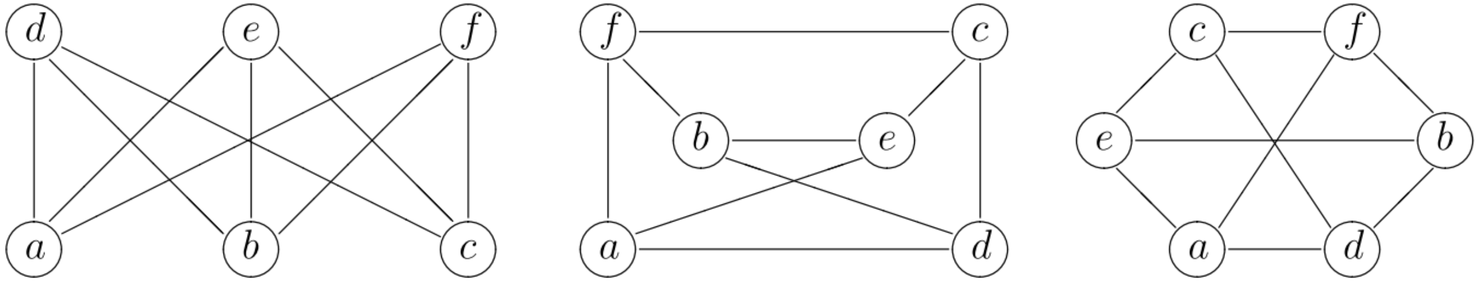
- **Определение.** Граф $G(V,E)$ называется *двудольным*, если множество V можно разбить на два непересекающихся подмножества V_1 и V_2 таким образом, что любое ребро из E соединяет вершину из V_1 с вершиной из V_2 . Множества V_1 и V_2 называются *долями* двудольного графа.
- **Определение.** Если двудольный граф содержит все возможные ребра, то есть $\forall\, v_1 \in V_1, v_2 \in V_2 \colon v_1 v_2 \in E$, то он называется *полным двудольным* графом и обозначается $K_{m,n}$, где $m = |V_1|$, $n = |V_2|$.
- **Пример.**



Полный двудольный граф $K_{3,3}$

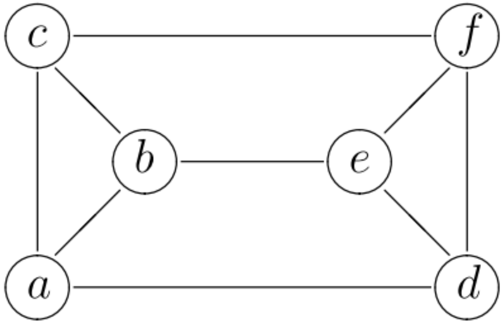
1.5. Изоморфизм графов

- **Определение.** Графы называются *изоморфными*, если между их вершинами можно установить взаимно-однозначное соответствие, сохраняющее смежность: Вершины в одном графе смежны тогда и только тогда, когда смежны соответствующие вершины во втором графе.
- **Пример**. Все приведенные графы являются изоморфными, это различные диаграммы двудольного графа $K_{3,3}$.



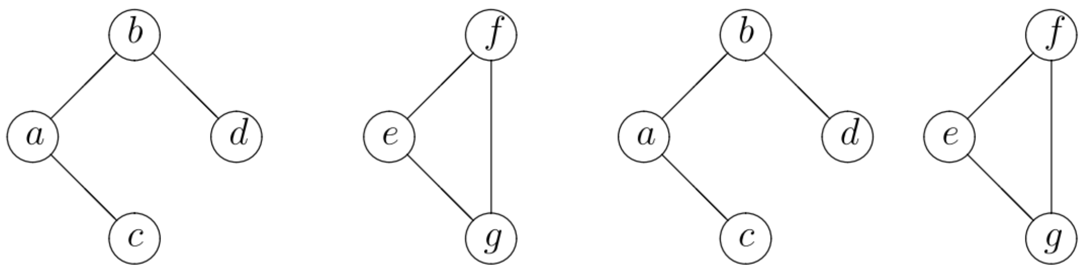
Изоморфизм графов

- Если графы являются изоморфными, то они имеют одинаковое количество вершин и ребер, а также одинаковое число вершин одной валентности. Обратное неверно: графы, у которых все эти характеристики совпадают, могут не быть изоморфными.
- **Пример**. Граф, неизоморфный $K_{3,3}$



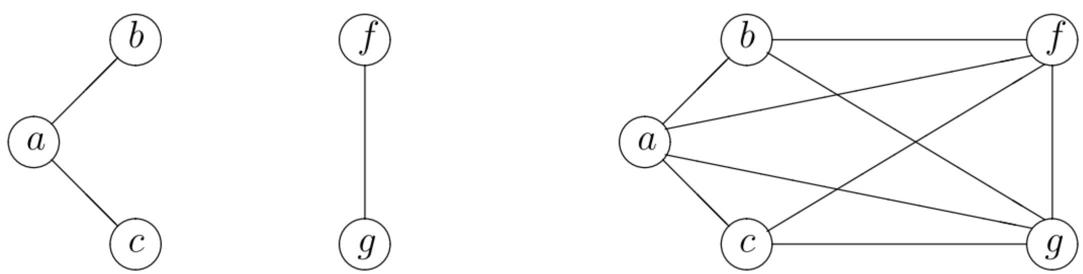
1.6. Операции над графами.

- **Объединение**



$G_1(V_1, E_1)$ $\quad$ $G_2(V_2, E_2)$ $\quad$ $G(V, E) = G_1(V_1, E_1) \cup G_2(V_2, E_2)$

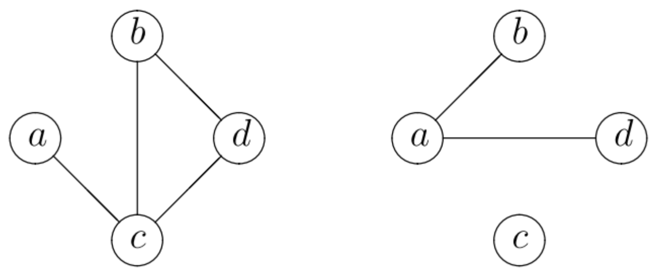
- **Соединение**



$G_1(V_1, E_1)$ $\quad$ $G_2(V_2, E_2)$ $\quad$ $G(V, E) = G_1(V_1, E_1) + G_2(V_2, E_2)$

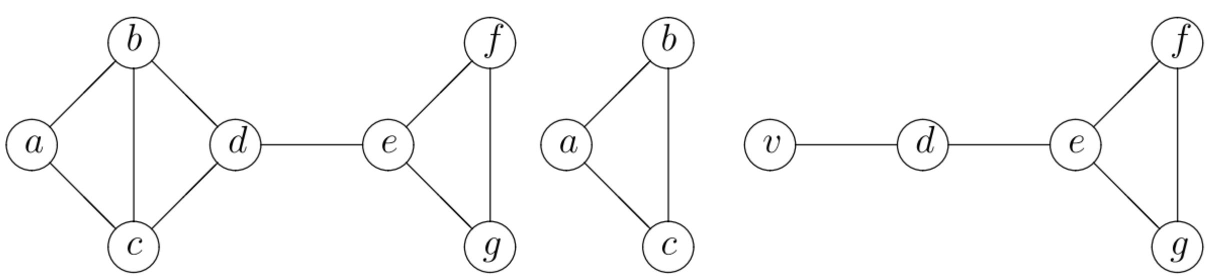
Операции над графами.

- **Дополнение**



$G_1(V_1, E_1)$ $\quad$ $G_2(V_2, E_2) = \overline{G_1}(V_1, E_1)$

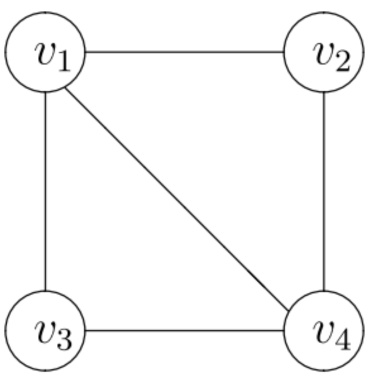
- **Стягивание правильного подграфа в вершину**



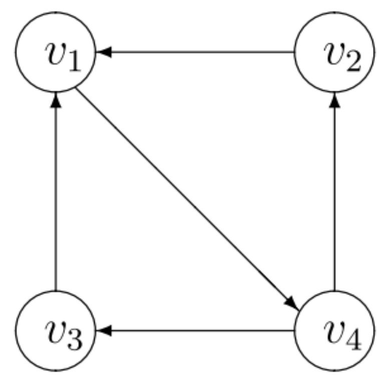
$G_1(V_1, E_1)$ $\quad$ $A(V, E)$ $\quad$ $G_2(V_2, E_2)$

1.7. Способы задания графа

- *Списки вершин и ребер* уже рассмотрены выше.
- *Списки смежности.* Каждой вершине графа сопоставляется список смежных ей вершин.
- **Примеры.**



$$
\begin{aligned}
v_1 &: \quad v_2, v_3, v_4;\\
v_2 &: \quad v_1, v_4;\\
v_3 &: \quad v_1, v_4;\\
v_4 &: \quad v_1, v_2, v_3.
\end{aligned}
$$



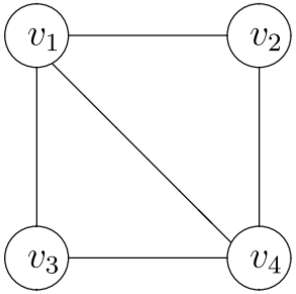
$$
\begin{aligned}
v_1 &: \quad v_4;\\
v_2 &: \quad v_1;\\
v_3 &: \quad v_1;\\
v_4 &: \quad v_2, v_3.
\end{aligned}
$$

Способы задания графа

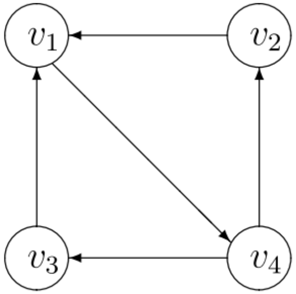
- *Матрица смежности.* Так называется матрица $M : p \times p$,

$$m_{i,j} = \begin{cases} 1, & ij \in E; \\ 0 & ij \notin E. \end{cases}$$

- **Примеры.**



$$M(G) = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 1 & 1 & 1 \\ v_2 & 1 & 0 & 0 & 1 \\ v_3 & 1 & 0 & 0 & 1 \\ v_4 & 1 & 1 & 1 & 0 \end{array}$$



$$M(D) = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 0 & 0 & 1 \\ v_2 & 1 & 0 & 0 & 0 \\ v_3 & 1 & 0 & 0 & 0 \\ v_4 & 0 & 1 & 1 & 0 \end{array}$$

Способы задания графа

- *Матрица инциденций*. Так называется матрица $H : p \times q$,

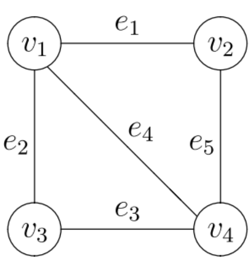
$$h_{i,j} = \begin{cases} 1, & \text{вершина } i \text{ инцидентна вершине } j; \\ 0 & \text{иначе.} \end{cases}$$

для неориентированного графа,

$$h_{i,j} = \begin{cases} 1, & \text{ребро } j \text{ выходит из вершины } i, \\ -1, & \text{ребро } j \text{ заходит в вершину } i, \\ 0, & \text{в остальных случаях.} \end{cases}$$

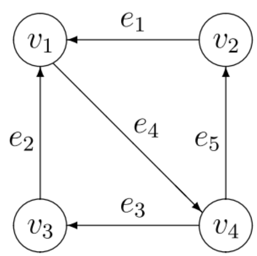
для ориентированного графа.

- **Примеры.**



$H(G) =$

	e_1	e_2	e_3	e_4	e_5
v_1	1	1	0	1	0
v_2	1	0	0	0	1
v_3	0	1	1	0	0
v_4	0	0	1	1	1



$H(D) =$

	e_1	e_2	e_3	e_4	e_5
v_1	−1	−1	0	1	0
v_2	1	0	0	0	−1
v_3	0	1	−1	0	0
v_4	0	0	1	−1	1

Способы задания графа

	Список ребер	**Списки смежности**	**Матрица смежности**	**Матрица инциденций**
Память	q	q	$p \times p$	$\boldsymbol{p \times q}$
Проверка смежности двух вершин	$\boldsymbol{q}$	p	1	$\boldsymbol{q}$

2. Связность

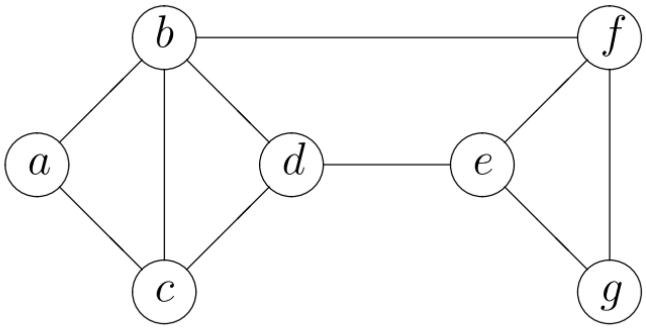
Маршруты, цепи, циклы
Расстояния в графе
Связность неориентированных графов
Оценка числа ребер в графе
Связность орграфов

2.1. Маршруты, цепи, циклы

- **Определение.** *Маршрутом* в графе называется последовательность вершин и ребер вида $v_0 e_1 v_1 e_2 \dots\ v_{k-1} e_k v_k$, в которой $e_i = v_{i-1} v_i$. Вершина v_0 называется *начальной*, а v_k – *конечной* вершиной маршрута.
- Для графа без кратных ребер достаточно указать только последовательность вершин.
- **Определение.** Если все ребра в маршруте различны, то маршрут называется *цепью*. Если все вершины (а значит и ребра) в маршруте различны, то маршрут называется *простой цепью*.
- **Определение.** Если $v_0 = v_k$, маршрут называется замкнутым.
- **Определение.** Если все ребра в замкнутом маршруте различны, то он называется *циклом*. Если все вершины в замкнутом маршруте, кроме первой и последней, различны, то он называется *простым циклом*.
- В орграфе цепь называется *путем*, а цикл – *контуром*.

Маршруты, цепи, циклы

- **Примеры.**

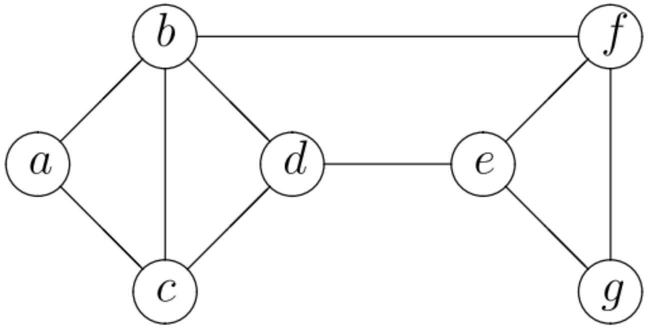


$abdbc$ – маршрут, но не цепь;
$abdcb$ – цепь, но не простая цепь;
$abcde$ – простая цепь;
$abdbca$ – замкнутый маршрут, но не цикл;
$abfedbca$ – цикл, но не простой цикл;
$abca$ – простой цикл.

- **Лемма о цепи.** Если есть цепь, соединяющая вершины *u*, *v*, то есть и простая цепь, соединяющая вершины *u*, *v*.

2.2. Расстояния в графе

- **Определение.** *Длиной маршрута* называется количество ребер в нем. Если маршрут $\mu = v_0 e_1 v_1 e_2 \ldots\ v_{k-1} e_k v_k$, то длина μ равна *k* (| μ | = *k*).
- **Определение.** *Расстоянием* между вершинами *u*, *v* (обозначается $s(u,v)$) называется наименьшая длина цепи $\langle u,v\rangle$. Цепь μ $= \langle u,v\rangle$, для которой $|\mu| = s(u,v)$, называется *кратчайшей* цепью.
- Если $\nexists\langle u,v\rangle$, то по определению $s(u,v) = \infty$.
- **Пример**. В рассмотренном графе $s(a,b) = 2$, кратчайшая цепь, например, *abd*.

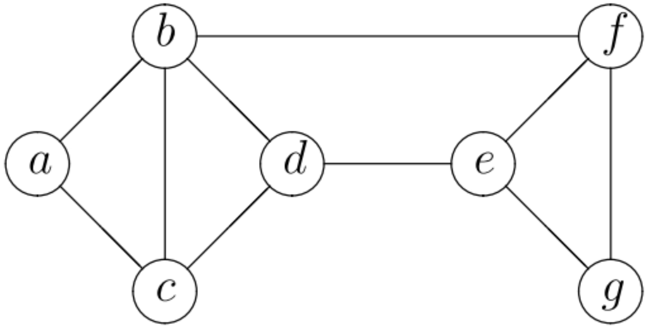


Расстояния в графе

- **Определение.** *Диаметром графа* $G(V,E)$ (обозначается $D(G)$) называется наибольшее расстояние между двумя его вершинами
$$D(G) = \max_{u,v \in V} s(u,v)$$
- **Определение.** Максимальное из расстояний между вершиной u и остальными вершинами из $G(V,E)$ называется *максимальным удалением* в графе $G(V,E)$ от вершины u (обозначается $r(u)$)
$$r(v) = \max_{u,v \in V} s(u,v)$$
- **Определение.** *Радиусом графа* $G(V,E)$ (обозначается $R(G)$ называется минимальное из максимальных удалений
$$R(G) = \min_{u \in V} r(u) = \min_{u \in V} r(u) \max_{u,v \in V} s(u,v)$$
- **Определение.** Любая вершина $v \in V$, такая что $r(v) = R(G)$, называется *центром графа*.

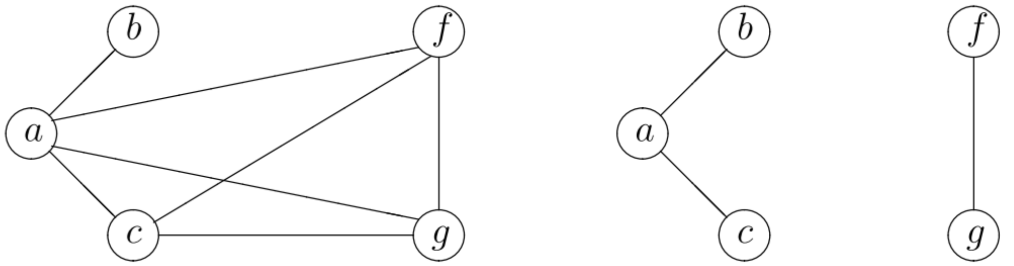
Расстояния в графе

- **Примеры.**
- $D(G) = 3$
- $r(a) = r(c) = r(e) = r(f) = r(g) = 3$
- $r(b) = r(d) = 2$
- $R(G) = 2$



2.3. Связность неориентированных графов

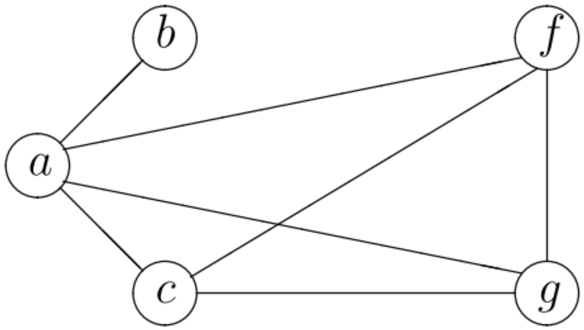
- **Определение.** Две вершины в графе *связны*, если существует соединяющая их цепь.
- **Определение**. Граф называется *связным*, если любые две вершины в нем связны.
- **Определение.** *Компонентой связности* графа $G(V,E)$ называется его правильный связный подграф, не являющийся собственным подграфом никакого другого связного подграфа графа $G(V,E)$.
- **Примеры.**

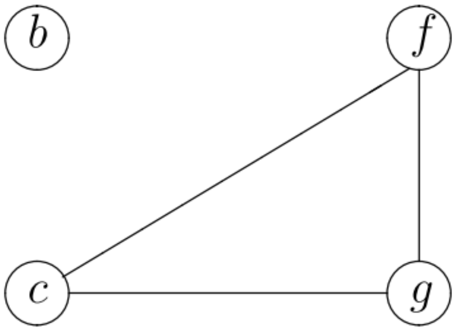


Связный граф Граф с двумя компонентами связности

Связность неориентированных графов

- **Определение.** Вершина графа $G(V,E)$ называется *точкой сочленения*, если ее удаление (вместе с инцидентными ей ребрами) увеличивает число компонент связности графа.
- **Пример.** Для графа слева a – точка сочленения. Результат ее удаления (граф с двумя компонентами связности) показан на рисунке справа.

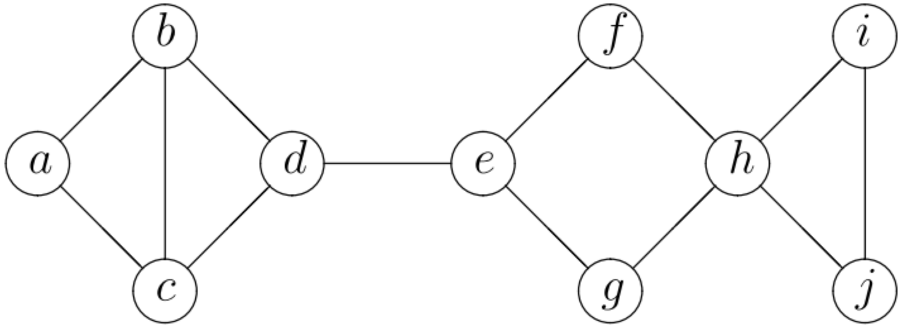




- **Лемма о точках сочленения.** В любом графе $G(V,E)$ с $p \geq 2$ есть по крайней мере две вершины, не являющиеся точками сочленения.

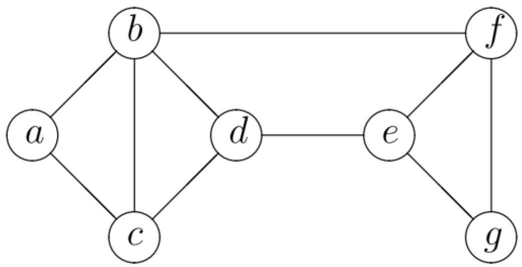
Связность неориентированных графов

- **Определение.** Ребро графа $G(V,E)$ называется *мостом*, если его удаление увеличивает число компонент связности графа.
- **Пример.** Мостом является ребро de.



Связность неориентированных графов

- **Определение.** *Числом вершинной связности* графа $G(V,E)$ (обозначается $\kappa(G)$) называется наименьшее число вершин, удаление которых приводит к несвязному или тривиальному графу. Граф $G(V,E)$ называется m*-связным*, если $\kappa(G) = m$.
- **Определение.** *Числом реберной связности* графа $G(V,E)$ (обозначается $\lambda(G)$) называется наименьшее число ребер, удаление которых приводит к несвязному графу.
- **Примеры.** Здесь $\kappa(G) = 2$ (удаление вершин b и c приведет к несвязному графу), $\lambda(G) = 2$ (например, можно удалить ребра bf и de).



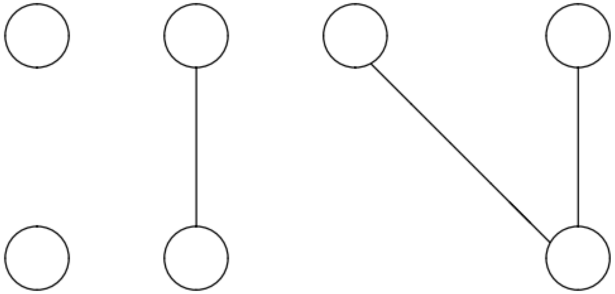
$$\kappa(G) \leq \lambda(G) \leq \delta(G)$$

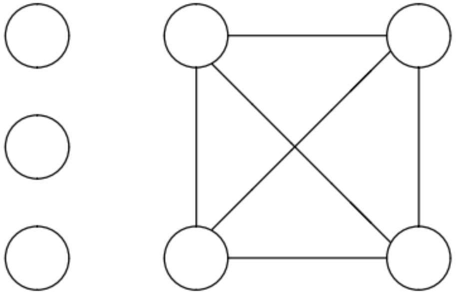
2.4. Оценка числа ребер в графе

- **Теорема.** Число вершин p, ребер q и компонент связности k графа $G(V,E)$ удовлетворяет неравенствам

$$p - k \leq q \leq \frac{(p-k)(p-k+1)}{2}$$

- **Пример.** p=7, k=4, $3 \leq q \leq 6$.



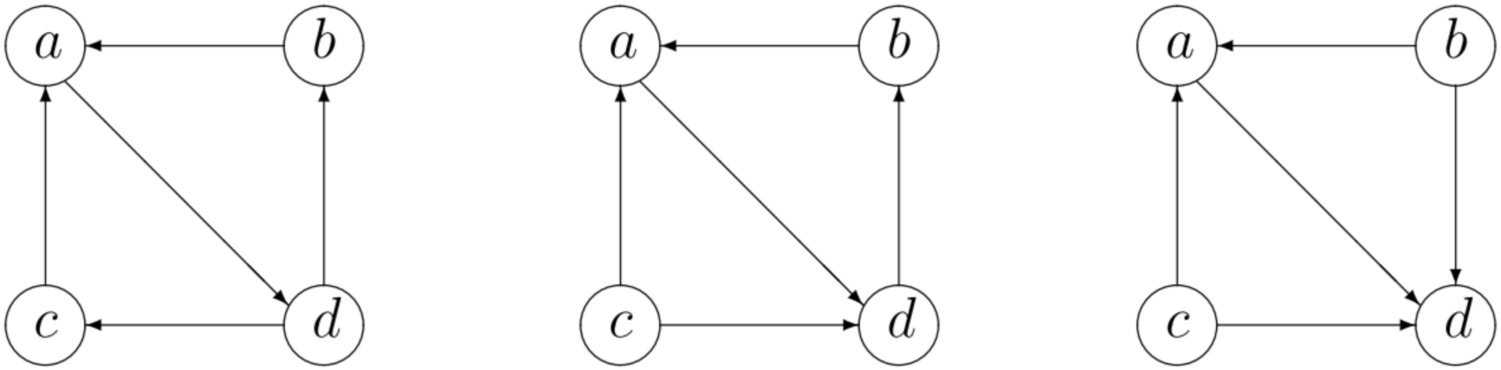


2.5. Связность орграфов

- **Определение.** Говорят, что два узла v_1 и v_2 *сильно связны* в орграфе $D(V,E)$, если существует путь из v_1 в v_2 и из v_2 в v_1.
- **Определение.** Говорят, что два узла v_1 и v_2 *односторонне связны* в орграфе $D(V,E)$, если есть путь либо из v_1 в v_2, либо из v_2 в v_1.
- **Определение.** Говорят, что два узла v_1 и v_2 *слабо связны* в орграфе $D(V,E)$, если они связны в графе $D'(V',E')$, полученном из $D(V,E)$ отменой ориентации ребер.

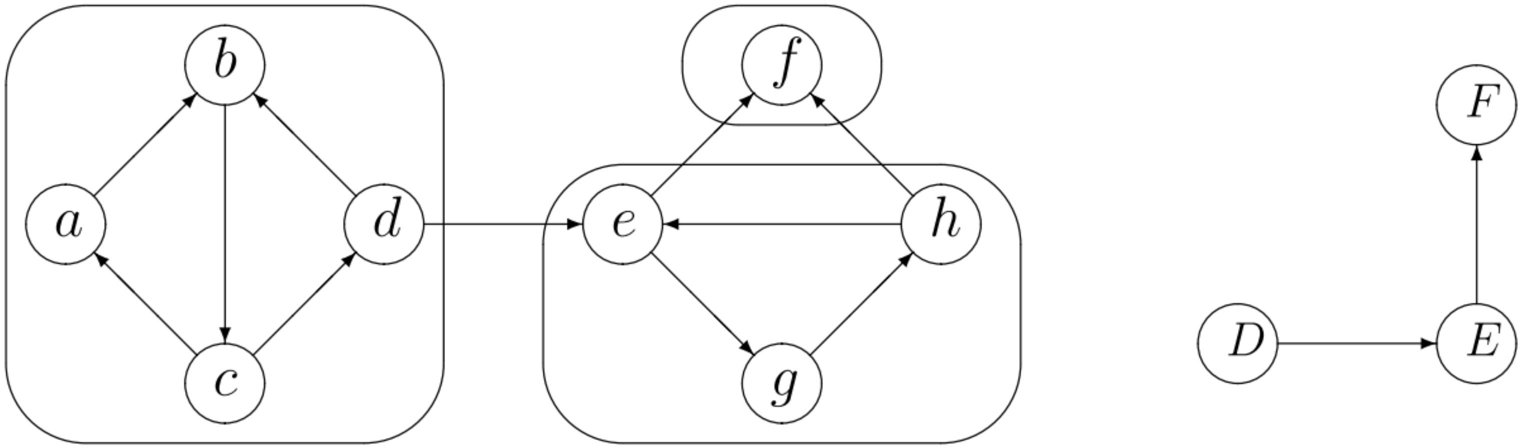
Связность орграфов

- **Определение.** Орграф $D(V,E)$ называется *сильно* (*односторонне*, *слабо*) *связным*, если любые два узла в нем сильно (односторонне, слабо) связны.
- **Пример.** Слева – сильно связный орграф, в центре – односторонне связный (нет путей до узла c из остальных), справа – слабо связный (нет путей между узлами b и c).



Связность орграфов

- **Определение.** *Компонентой сильной связности* орграфа $D(V,E)$ называется его правильный сильно связный подграф, не являющийся собственным подграфом никакого другого сильно связного подграфа орграфа $D(V,E)$.
- **Определение.** *Конденсацией* (*фактор-графом*) орграфа $D(V,E)$ называется орграф, который получается стягиванием в один узел каждой компоненты сильной связности орграфа $D(V,E)$.
- **Пример.**

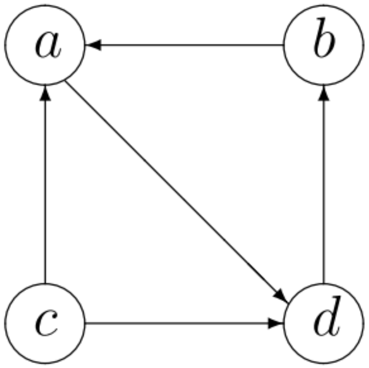


Связность орграфов

- **Определение.** *Матрицей достижимости* орграфа $D(V,E)$ называется квадратная матрица $T(D)$ размерности $p \times p$,
$$t_{i,j} = \begin{cases} 1, & \exists\langle i,j\rangle; \\ 0, & \text{иначе.} \end{cases}$$
- Считается, что путь $\langle i,i\rangle$ существует всегда (это путь длины 0).
- **Определение.** *Матрицей сильной связности* орграфа $D(V,E)$ называется квадратная матрица $S(D)$ размерности $p \times p$, где
$$s_{i,j} = \begin{cases} 1, & \exists\langle i,j\rangle, \langle j,i\rangle; \\ 0, & \text{иначе.} \end{cases}$$
- Иначе говоря, $s_{i,j} = 1$ тогда и только тогда, когда узлы i и j принадлежат одной компоненте сильной связности орграфа $D(V,E)$.

Связность орграфов

- **Пример.** Рассмотрим орграф $D(V, E)$, его матрицу достижимости и сильной связности.



$$T(D) = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 0 & 1 \\ b & 1 & 1 & 0 & 1 \\ c & 1 & 1 & 1 & 1 \\ d & 1 & 1 & 0 & 1 \end{array} \quad S(D) = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 0 & 1 \\ b & 1 & 1 & 0 & 1 \\ c & 0 & 0 & 1 & 0 \\ d & 1 & 1 & 0 & 1 \end{array}$$

Связность орграфов

Алгоритмы построения матрицы сильной связности

- *Алгоритм, основанный на операциях над булевыми матрицами*, вычислительная сложность p^4. Также позволяет проверить наличие путей заданной длины.
- *Алгоритм Уоршалла*, вычислительная сложность p^3. Также позволяет проверить наличие путей, проходящих через заданное множество вершин..

Поиск путей в графе

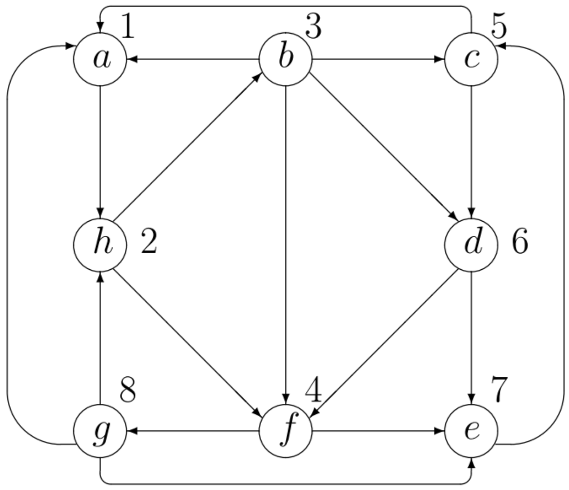
Обходы графа

Поиск кратчайших путей

Поиск минимальных путей

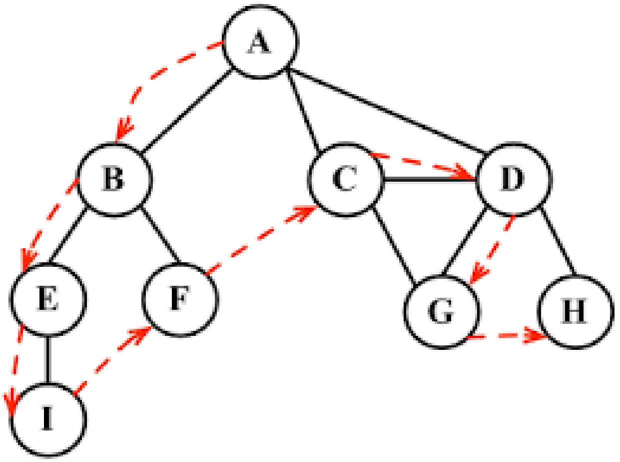
3.1. Стратегии обхода графа

- **Определение.** *Обходом графа* называется систематическое перечисление его вершин.
- **Стратегия обхода «в ширину».** Начиная со стартовой вершины, обходим все смежные с ней вершины. Затем для каждой из них повторяем эту процедуру, заходя только в непосещенные вершины. Процесс заканчивается, если посещены все вершины, либо нет непосещенных вершин, смежных с уже посещенными.
- **Пример.**



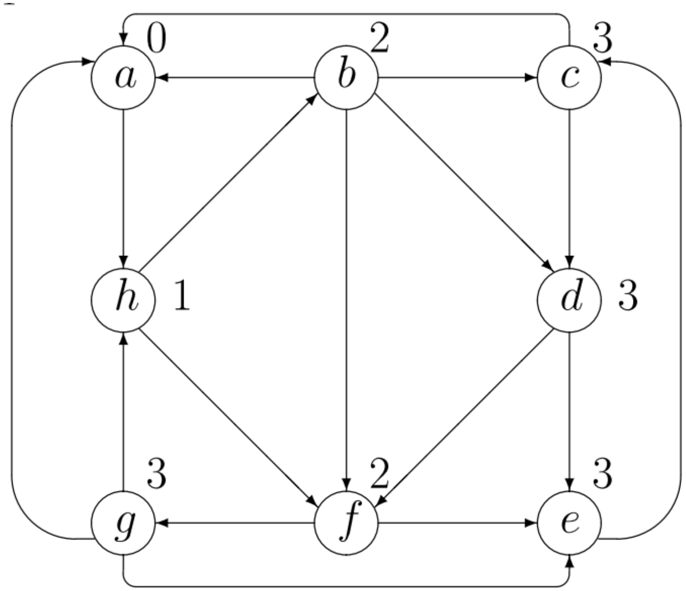
Стратегии обхода графа

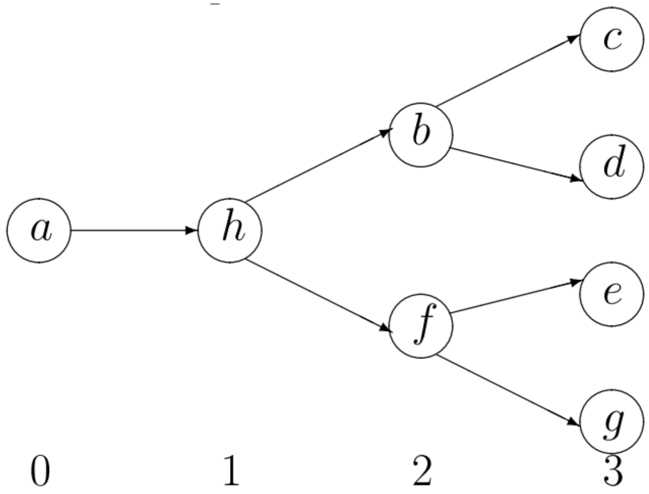
- **Стратегия обхода «в глубину».** Начиная со стартовой вершины, идем в смежную с ней вершину. Затем из этой вершины идем в смежную с ней непосещенную вершину, и т. д. Если нет непосещенных вершин, смежных с текущей, возвращаемся на шаг назад и идем в смежную непосещенную вершину, если нет таких вершин, возвращаемся еще на шаг назад, и т. д. Процесс заканчивается, если посещены все вершины, либо мы вернулись в стартовую вершину и не осталось смежных с ней непосещенных вершин.
- **Пример**.



3.2. Поиск кратчайших путей

- **Определение.** *Кратчайшим путем* $\langle u, v \rangle$ в графе называется путь наименьшей длины.
- *Волновой алгоритм* (*алгоритм Ли*). Вычислительная сложность p^2.
- **Пример**.





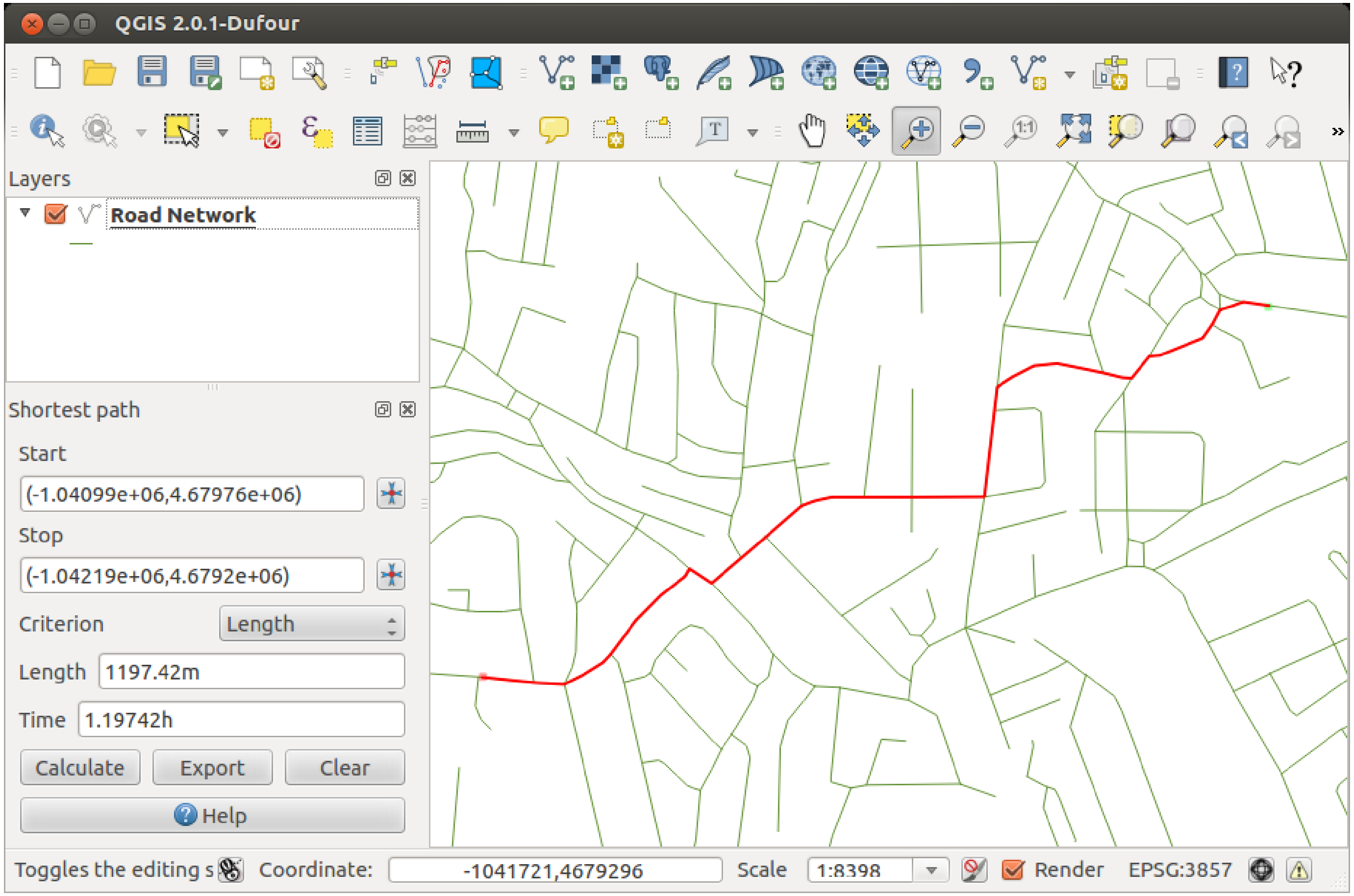
3.3. Поиск минимальных путей

- **Определение.** *Взвешенным графом* $G(V,E)$ называется граф, каждому ребру которого сопоставлено некоторое число, называемое *весом*.
- **Определение.** *Минимальным путем* $\langle u, v \rangle$ в графе называется путь с наименьшей суммой весов ребер.

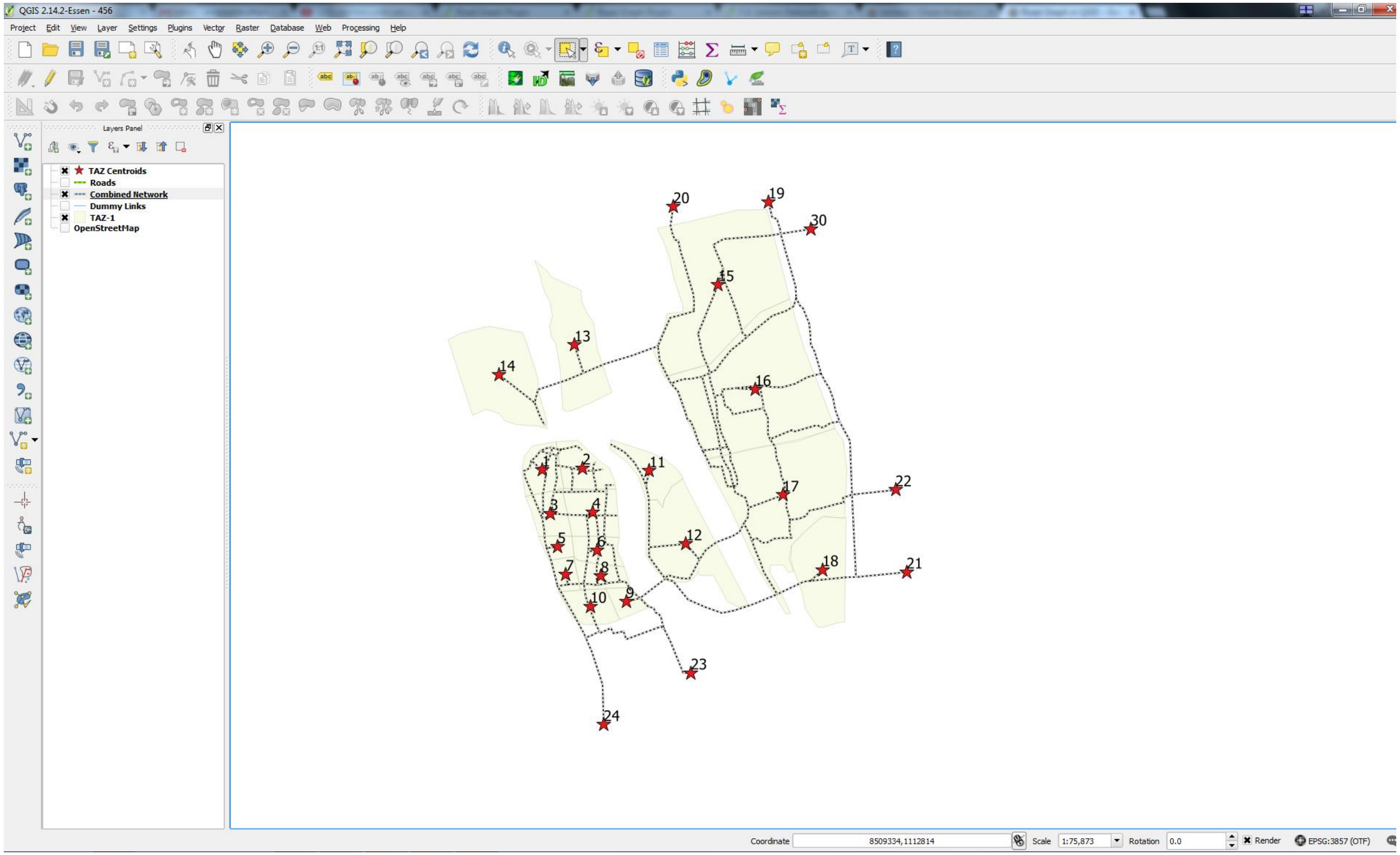
Задачи:

- Поиск пути между заданными вершинами;
- Поиск путей от заданной вершины до всех остальных;
- Поиск путей между всеми парами вершин.

Поиск минимальных путей



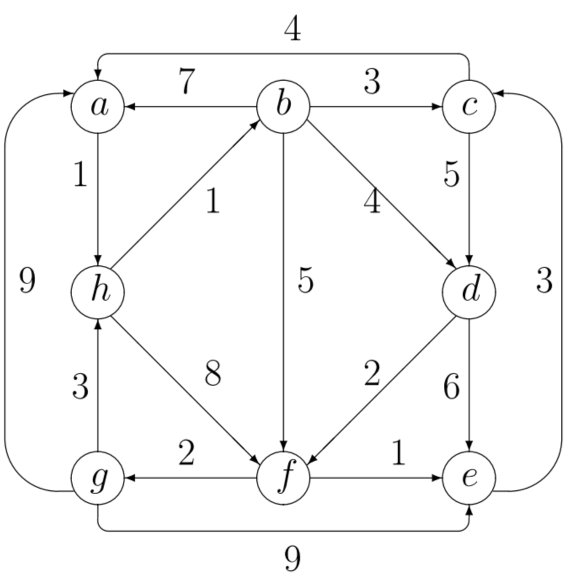
Поиск минимальных путей



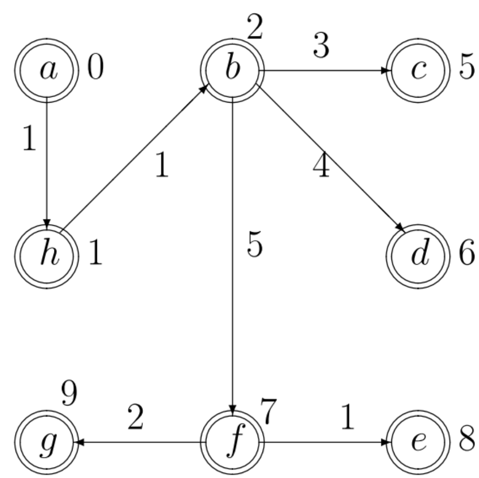
Поиск минимальных путей

Поиск пути от заданной стартовой вершины

- *Алгоритм Дейкстра*: жадный алгоритм, вычислительная сложность p^2, веса ребер должны быть неотрицательны.
- **Пример.**



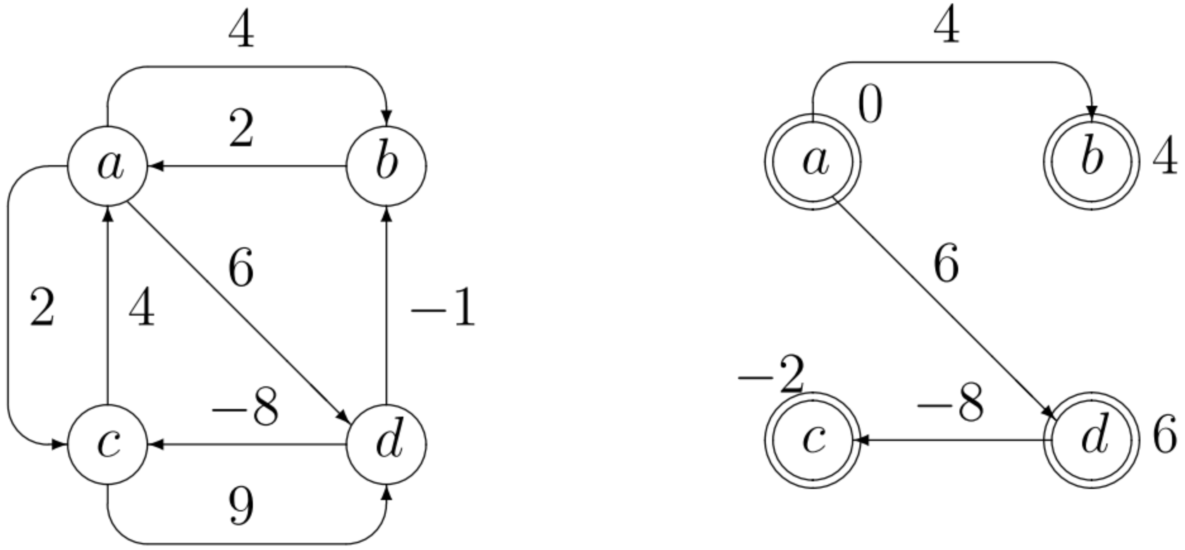
$$\begin{aligned} \lambda(a) &= 0; & \lambda(d) &= 6;\\ \lambda(h) &= 1; & \lambda(f) &= 7;\\ \lambda(b) &= 2; & \lambda(e) &= 8;\\ \lambda(c) &= 5; & \lambda(g) &= 9. \end{aligned}$$



Поиск минимальных путей

Поиск пути от заданной стартовой вершины

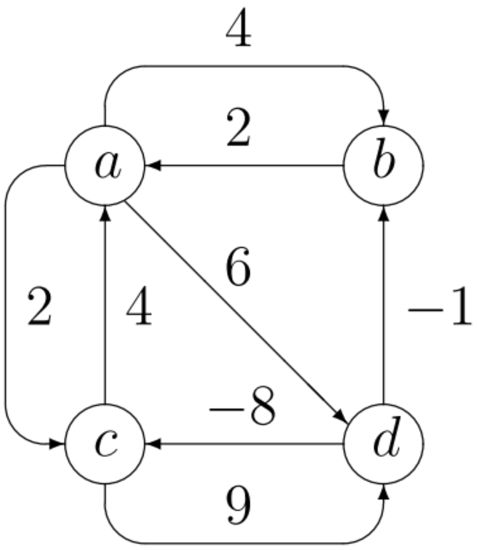
- *Модификация алгоритма Дейкстра* для произвольных весов ребер, вычислительная сложность p^3.
- *Алгоритм Форда-Беллмана-Мура*: динамическое программирование, для произвольных весов ребер, вычислительная сложность pq.
- **Пример.**



Поиск минимальных путей

Поиск пути между всеми парами вершин

- *Алгоритм Флойда-Уоршалла*: динамическое программирование, для произвольных весов ребер, вычислительная сложность p^3. .
- **Пример.**



$\Lambda =$

	a	b	c	d
a	0	4	−2	6
b	2	0	0	8
c	4	8	0	9
d	−4	−1	−8	0

Деревья

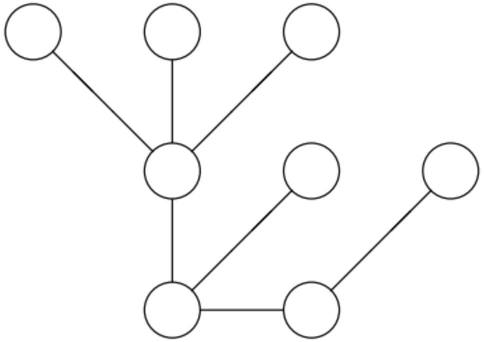
Свободные деревья

Кратчайшее остовное дерево

Ориентированные, упорядоченные и бинарные деревья

4.1. Свободные деревья

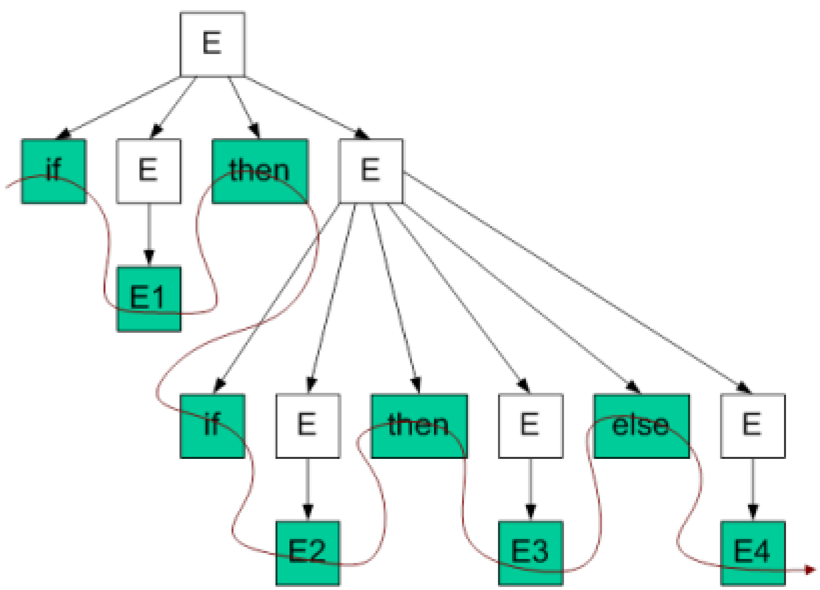
- **Определение**. Граф $G(V,E)$ называется *деревом*, если он является связным и не имеет циклов.
- **Определение.** Граф $G(V,E)$, все компоненты связности которого являются деревьями, называется *лесом*.
- **Пример.** Диаграмма дерева.

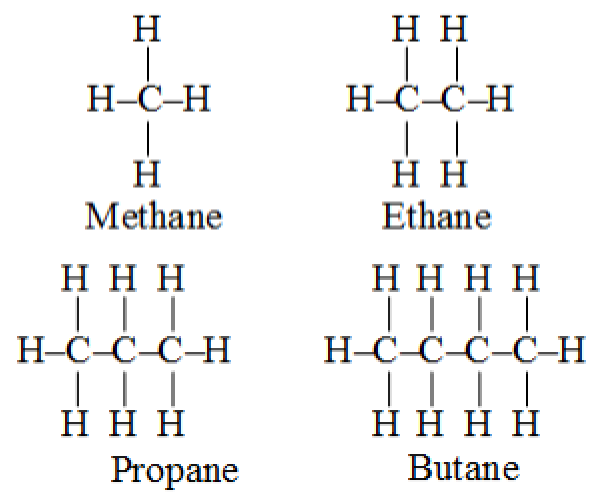


Свободные деревья

- **Химия** (насыщенные углеводороды)
- **Компиляторы** (парсинг)

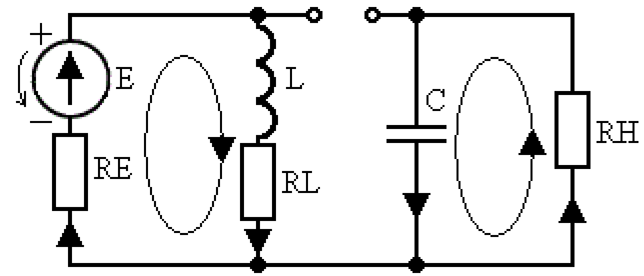
if E1 then if E2 then E3 else E4

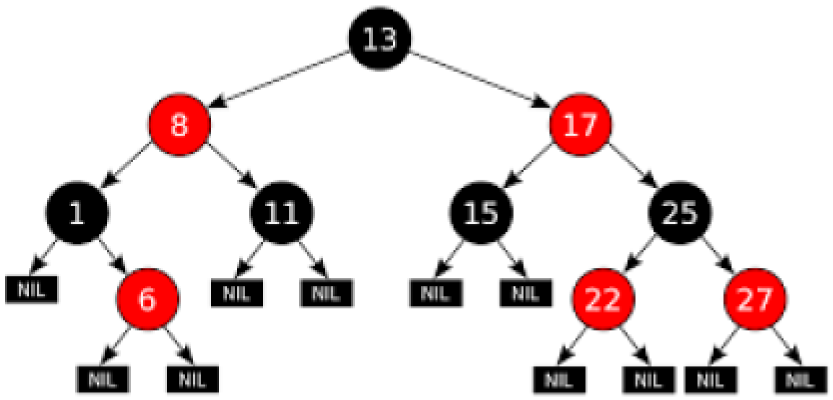


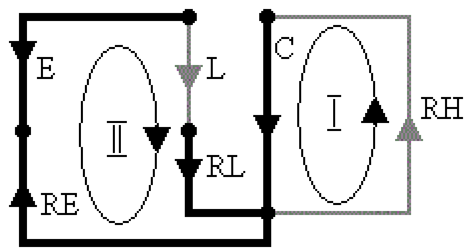


Свободные деревья

- **Физика** (электрические цепи)
- **Программирование** (деревья поиска)







Свободные деревья

Теорема о шести эквивалентных утверждениях о дереве. Следующие утверждения эквивалентны:

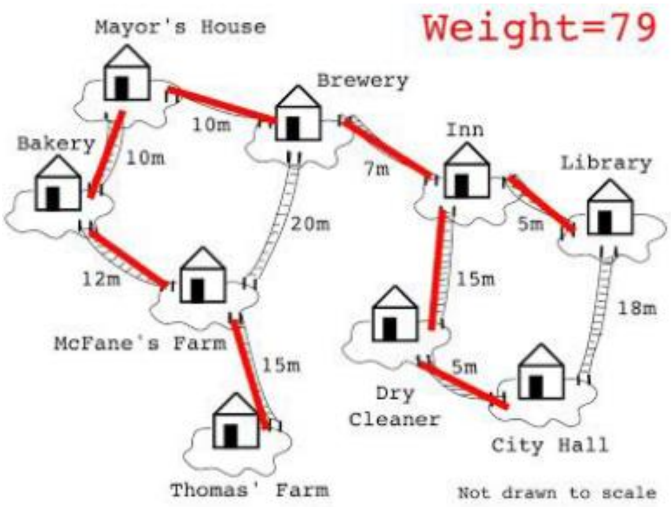
- 1) граф $G(V,E)$ – дерево, то есть связный граф без циклов;
- 2) любые две вершины графа $G(V,E)$ соединены единственной простой цепью;
- 3) граф $G(V,E)$ связный, и любое ребро есть мост;
- 4) граф $G(V,E)$ связный, и $q = p - 1$;
- 5) граф $G(V,E)$ не содержит циклов, и $q = p - 1$;
- 6) граф $G(V,E)$ не содержит циклов, но добавление к нему любого нового ребра приводит к образованию ровно одного простого цикла, проходящего через это ребро.

4.2. Кратчайший остов

Задача о кратчайшем соединении городов. Как соединить дома с минимальной суммарной длиной путей?

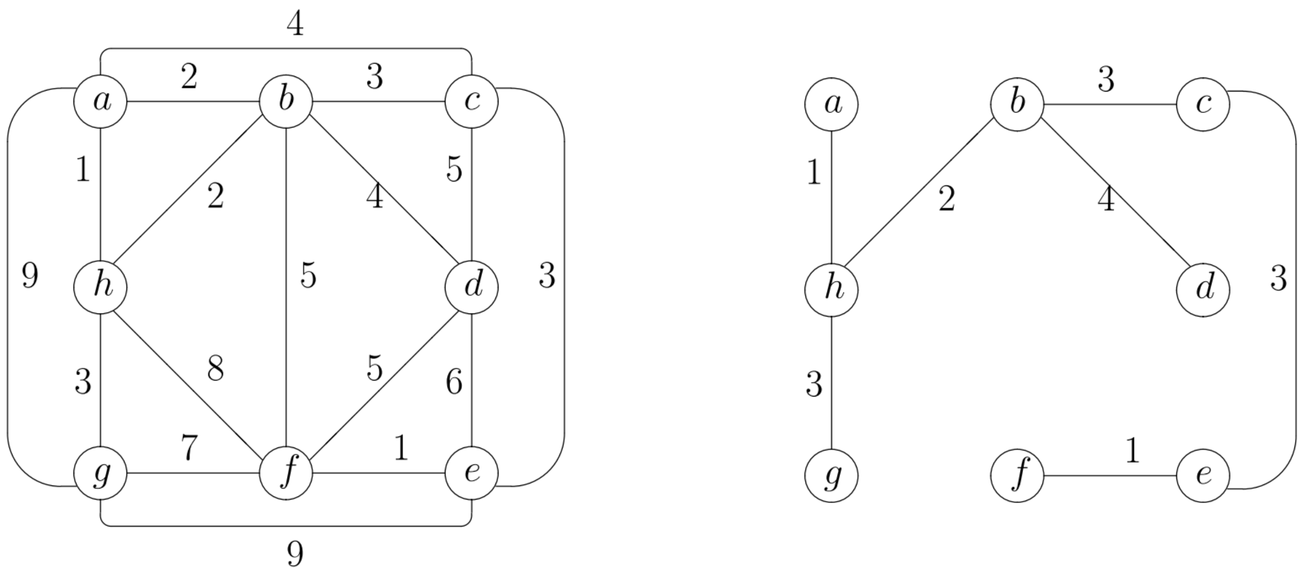
Применение:

- Проектирование сетей
- Аппроксимация для NP- полных задач
- Кластерный анализ



Кратчайший остов

- **Определение.** *Остовным деревом* (*остовом*) связного графа $G(V,E)$ называется его подграф, содержащий все вершины графа $G(V,E)$ и являющийся деревом.
- **Определение.** Во взвешенном графе *кратчайший остов* – остов с минимальной суммой весов ребер.



Кратчайший остов

- **Жадный алгоритм** — алгоритм, заключающийся в принятии *локально оптимальных решений* на каждом этапе, допуская, что конечное решение также окажется оптимальным (во многих случаях это не так).
- Для задачи поиска кратчайшего остова жадный алгоритм строит оптимальное решение.

Решение: жадные алгоритмы!

- *Алгоритм Краскала*
- *Алгоритм Прима*

4.3. Ориентированные, упорядоченные и бинарные деревья

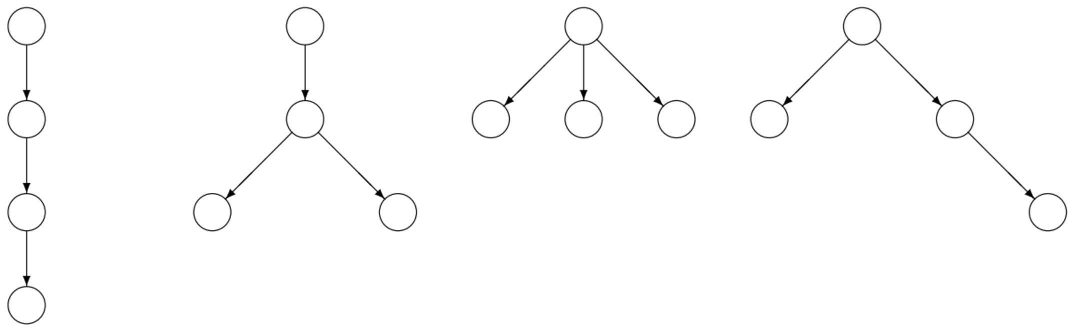
- **Определение.** *Ориентированным деревом* называется орграф со следующими свойствами:

1) существует единственный узел, полустепень захода которого равна 0 (он называется *корнем ордерева*);

2) полустепень захода всех остальных узлов равна 1;

3) каждый узел достижим из корня.

- **Пример.** Все различные ордеревья с четырьмя узлами.

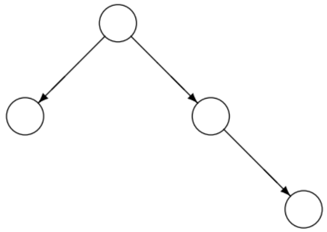


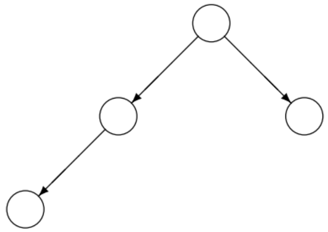
Ориентированные, упорядоченные и бинарные деревья

- **Теорема.** Ордерево обладает следующими свойствами:
- 1) $q = p - 1$;
- 2) если в ордереве отменить ориентацию ребер, то получится свободное дерево;
- 3) в ордереве нет контуров;
- 4) для каждого узла существует единственный путь, ведущий в этот узел из корня;
- 5) правильный подграф, определяемый множеством узлов, достижимых из узла v, является ордеревом с корнем v (это дерево называется *поддеревом* узла v);
- 6) если в свободном дереве любую вершину назначить корнем, то можно ориентировать ребра таким образом, что получится ордерево.

Ориентированные, упорядоченные и бинарные деревья

- **Определение.** Если порядок поддеревьев в ордереве фиксирован,то дерево называется *упорядоченным*.
- **Пример.** Данные поддеревья изоморфны как ориентированные деревья, но неизоморфны как упорядоченные.





Ориентированные, упорядоченные и бинарные деревья

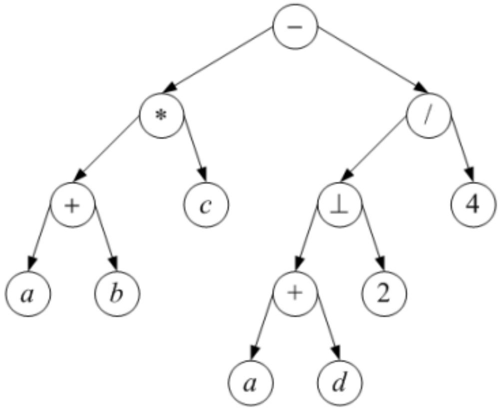
Применение в программировании.

1. Различные выражения языков программирования задаются упорядоченными деревьями.

2. Представление блочной структуры программы и связанной с ней структуры областей определения идентификаторов.

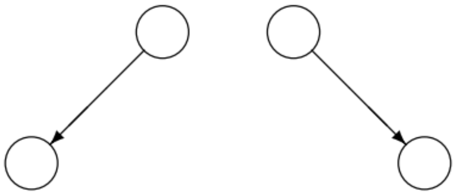
3. Для представления различных иерархических структур.

4. Для уравновешенных скобочных структур.



Ориентированные, упорядоченные и бинарные деревья

- **Определение.** Если поддеревьев в ордереве ровно два – левое и правое (могут быть пустыми), то дерево называется *бинарным*.
- **Пример.** Деревья изоморфны как ориентированные и упорядоченные, но неизоморфны как бинарные.

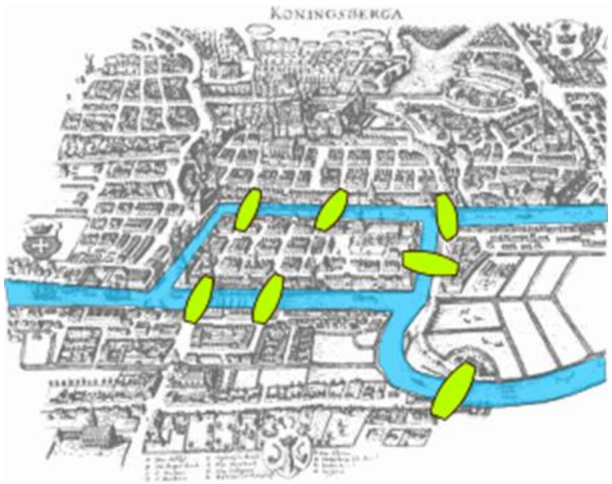


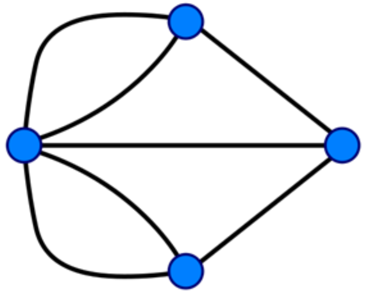
Эйлеровы графы

- Эйлеровы и полуэйлеровы графы
- Построение эйлерова цикла
- Задача почтальона

5.1. Эйлеровы и полуэйлеровы графы

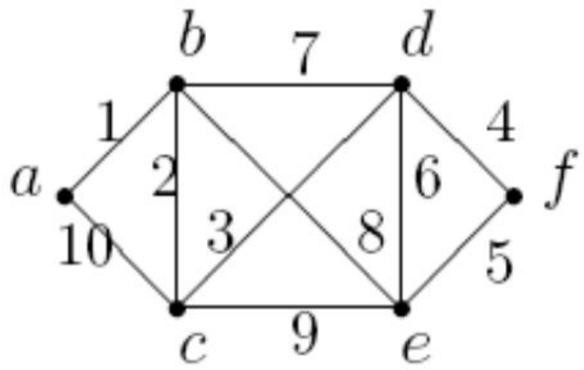
- Задача о Кенигсбергских мостах: как пройти по всем мостам, не проходя ни по одному из них дважды?
- Считается, что именно эта задача положила начало теории графов, поскольку Леонард Эйлер в 1736 году ввел модель графа в том виде, в котором она используется сейчас, сопоставив вершинам участки суши, а ребрам – соединяющие их мосты.





Эйлеровы и полуэйлеровы графы

- **Определение.** Если граф имеет цикл (не обязательно простой), содержащий все ребра графа по одному разу, то такой цикл называется *эйлеровым циклом*, а граф называется *эйлеровым графом*. Если граф имеет цепь (не обязательно простую), содержащую все ребра графа по одному разу, то такая цепь называется *эйлеровой цепью*, а граф называется *полуэйлеровым графом*.
- **Пример.** Цифрами обозначен порядок прохождения ребер.



Эйлеровы и полуэйлеровы графы

- **Теорема.** *Для того, чтобы связный граф G(V, E) был эйлеровым, необходимо и достаточно, чтобы степени всех вершин были четны.*
- **Теорема.** *Для того, чтобы связный граф G(V, E) был полуэйлеровым, необходимо и достаточно, чтобы он содержал ровно две вершины нечетной степени.*

5.2. Построение эйлерова цикла

Алгоритм Флери (1883).

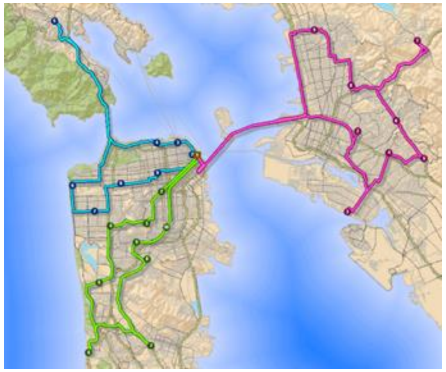
- Ориентирован на визуальное решение.
- Основан на поиске мостов.
- Вычислительная сложность от q^2 до qp^3.

Алгоритм, основанный на объединении циклов

- Использует стек
- Вычислительная сложность от q до qp.

5.3. Задача почтальона

- Почтальон должен обойти все улицы по крайней мере один раз и вернуться в начальную точку, при этом его маршрут должен быть как можно более коротким.
- На языке теории графов ставится задача поиска минимального замкнутого маршрута (т.е. маршрута с минимальной суммой весов ребер), содержащего все ребра взвешенного графа.

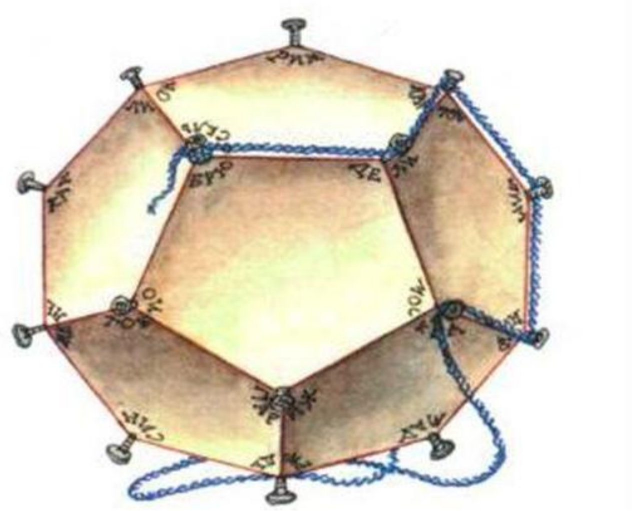


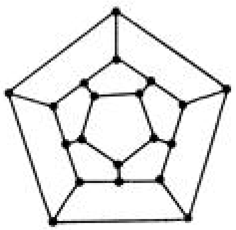
Гамильтоновы графы

- Гамильтоновы и полугамильтоновы графы
- Построение гамильтонова цикла
- Задача коммивояжера

6.1. Гамильтоновы и полугамильтоновы графы

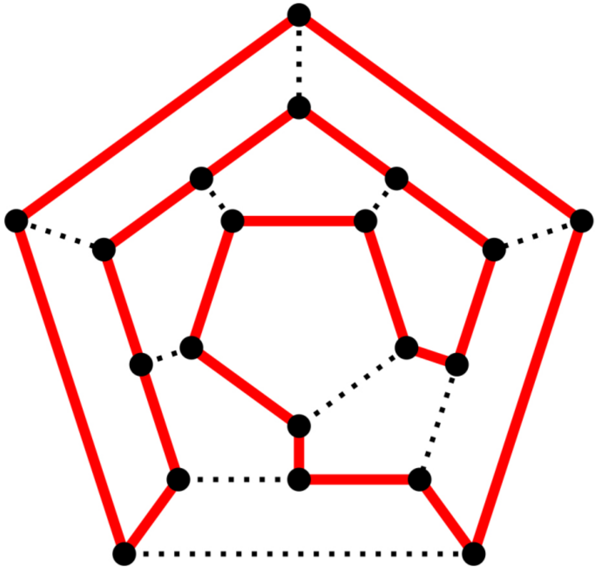
- **Уильям Роуэн Гамильтон** исследовал задачу "кругосветного путешествия" по додекаэдру.
- В этой задаче вершины додекаэдра символизировали известные города, такие как Брюссель, Амстердам, Эдинбург, Пекин, Прага, Дели, Франкфурт и др., а ребра – соединяющие их дороги. Путешествующий должен пройти "вокруг света", найдя путь, который проходит через все вершины ровно один раз.
- Гамильтон предложил новый вариант игры, заменив додекаэдр плоским графом.





Гамильтоновы и полугамильтоновы графы

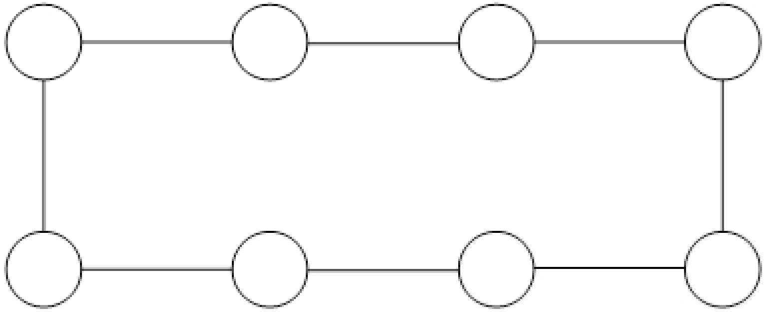
- Если граф имеет простой цикл, содержащий все вершины графа по одному разу, то такой цикл называется *гамильтоновым циклом*, а граф называется *гамильтоновым графом*.
- Если граф имеет простую цепь, содержащую все вершины графа по одному разу, то такая цепь называется *гамильтоновой цепью*, а граф называется *полугамильтоновым графом*.



Гамильтоновы и полугамильтоновы графы

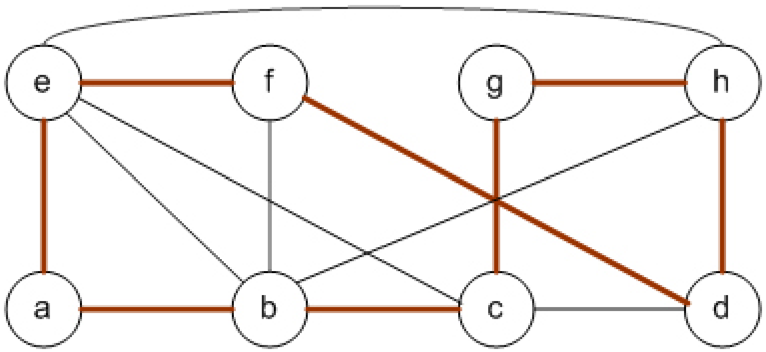
- **Теорема Дирака.** *Если в графе* $G(V,E)$ *с* $p \geq 3$ *вершинами* $\forall v \in V$ *верно* $d(v) \geq p/2$, *то граф* $G(V,E)$ *является гамильтоновым.*

- Обратное неверно, т. е. условие теоремы не является необходимым для существования гамильтонова цикла, что легко показать на примере.



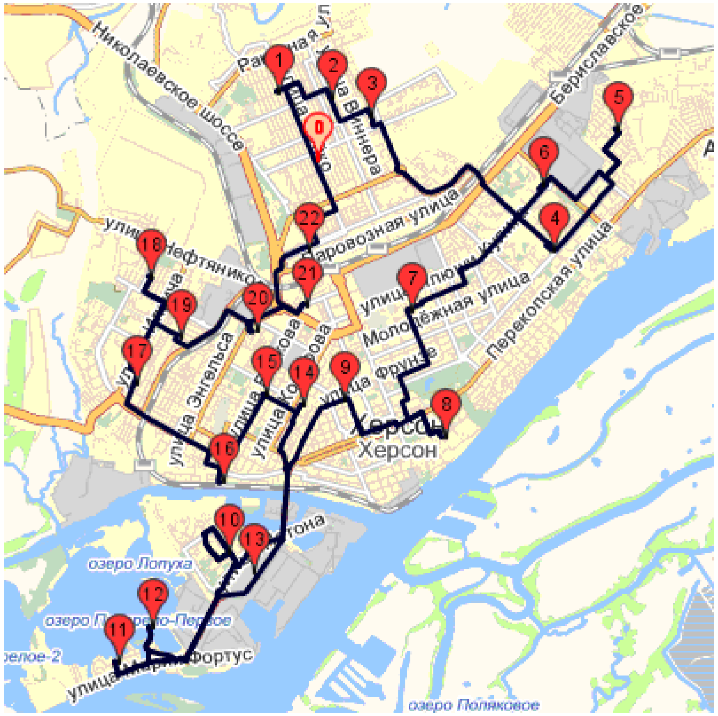
6.2. Поиск гамильтонова цикла

- Если условие теоремы Дирака, либо другой подобной теоремы, выполняется, то граф является гамильтоновым. В противном случае проверить свойство гамильтоновости можно, лишь построив гамильтонов цикл, либо показав, что его не существует.
- **Не существует** алгоритмов получения гамильтонова цикла, эффективных в вычислительном смысле, то есть требующих полиномиального времени.



6.3. Задача коммивояжера

- Имеется *p* городов, расстояния между которыми известны. Коммивояжер (странствующий торговец) должен посетить все *p* городов по одному разу, вернувшись в тот, с которого начал. Требуется найти такой маршрут движения, при котором суммарное пройденное расстояние будет минимальным.
- **NP-полная задача!**



Задача коммивояжера

- Многие современные распространенные методы дискретной оптимизации, такие как метод отсечений, ветвей и границ и различные варианты эвристических алгоритмов, были разработаны на примере задачи коммивояжера.
- *Алгоритм полного перебора*. Считая, что коммивояжер выезжает из одного и того же города, сгенерировать $(p-1)!$ перестановок остальных вершин и для каждой перестановки вычислить длину пути, выбрав затем из них минимальную. Этот алгоритм имеет большую вычислительную сложность, уже для 100 городов количество вариантов будет представляться 158-значным числом. Мощный компьютер, способный перебирать миллион вариантов в секунду, потратит на эту задачу примерно 3×10^{144} лет.

Задача коммивояжера

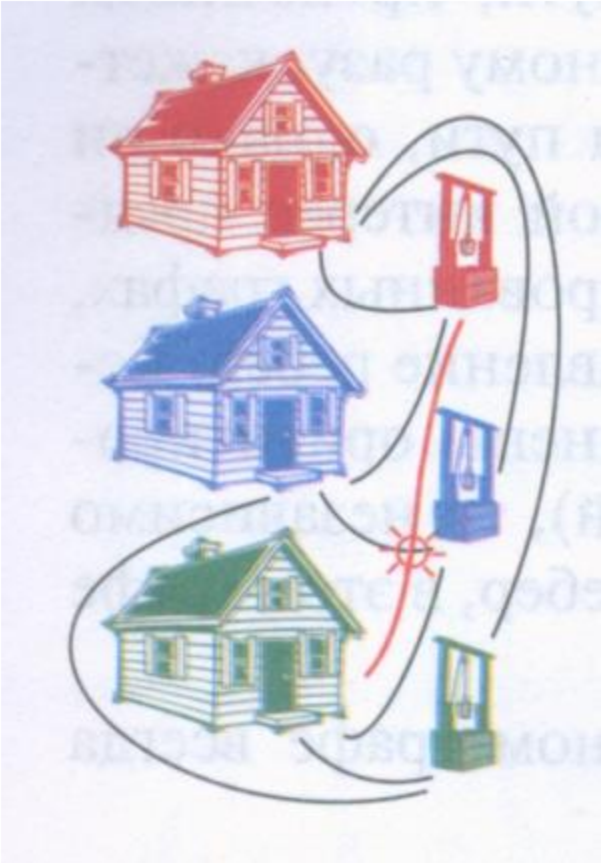
- *Методы сокращения перебора*. Эти методы позволяют не рассматривать часть вариантов решения, основываясь на некоторых оценках качества решения (например, метод ветвей и границ).
- *Эвристические методы*. Эти методы находят приближенное решение, т. е. в данном случае гамильтонов цикл, близкий к кратчайшему, за короткое время. Приближенные методы отличаются друг от друга вычислительной сложностью и качеством найденного решения.

Планарность

- Планарные и плоские графы
- Условия планарности
- Толщина графа

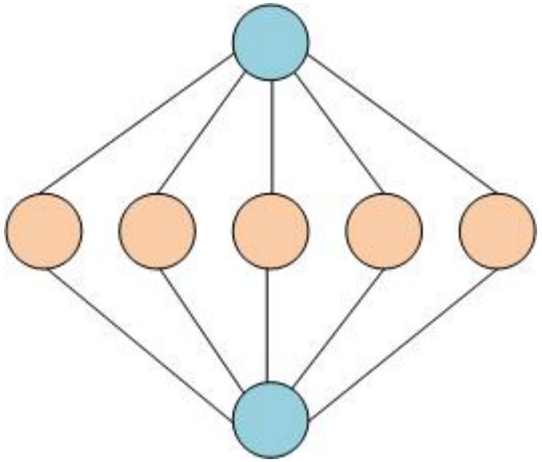
7.1. Планарные и плоские графы

- **Задача о трех домах и трех колодцах (Леонард Эйлер)**: есть три дома и три колодца. Можно ли так проложить дорожки между домами и колодцами, чтобы от каждого дома к каждому колодцу вела дорожка, и никакие две дорожки не пересекались бы.
- На языке теории графов эта задача звучит так: является ли граф $K_{3,3}$ *планарным*, т.е. можно ли нарисовать его диаграмму на плоскости без пересечения ребер?



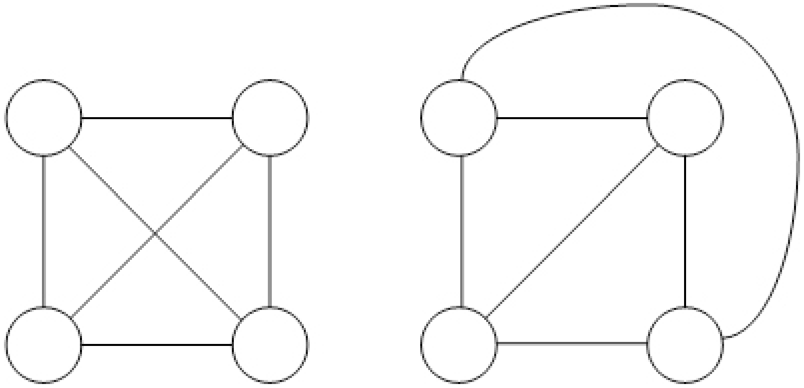
Планарные и плоские графы

- Граф *укладывается* на некоторой поверхности, если его диаграмму можно нарисовать на этой поверхности без пересечения ребер.
- Граф называется *планарным*, если его можно уложить на плоскости.
- Граф называется *плоским*, если он уложен на плоскости.
- **Пример.** Любой полный двудольный граф $K_{2,n}$ является планарным, его укладка $K_{2,5}$ для представлена на рисунке.



Планарные и плоские графы

- Область, ограниченная ребрами в плоском графе и не содержащая внутри себя вершин и ребер, называется *гранью*. Число ребер плоского графа G обозначается $r(G)$. Внешняя часть плоскости также образует грань. Неограниченная грань называется *внешней*, ограниченные – *внутренними*.
- **Пример**. Диаграммы планарного графа K_4 и его укладки на плоскости. Граф имеет 4 грани.

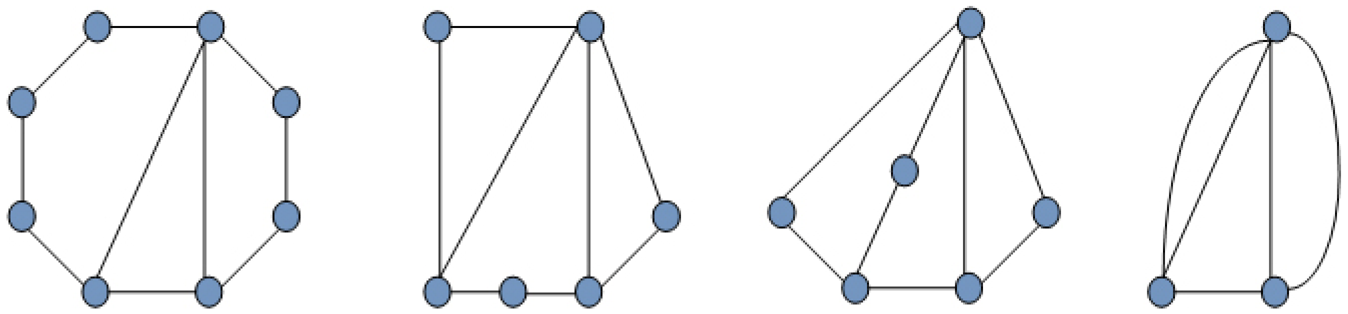


Планарные и плоские графы

- **Теорема (формула Эйлера)**. *В связном плоском графе* $p - q + r = 2$.
- **Следствие 1.** *В связном планарном графе при* $p > 3$ *верно:* $q \leq 3p - 6$
- **Следствие 2.** *В любом простом планарном графе существует вершина, степень которой не больше* 5.
- **Теорема о графах $\boldsymbol{K_5}$ и $\boldsymbol{K_{3,3}}$.** *Графы* K_5 *и* $K_{3,3}$ *непланарны.*

7.2. Условия планарности

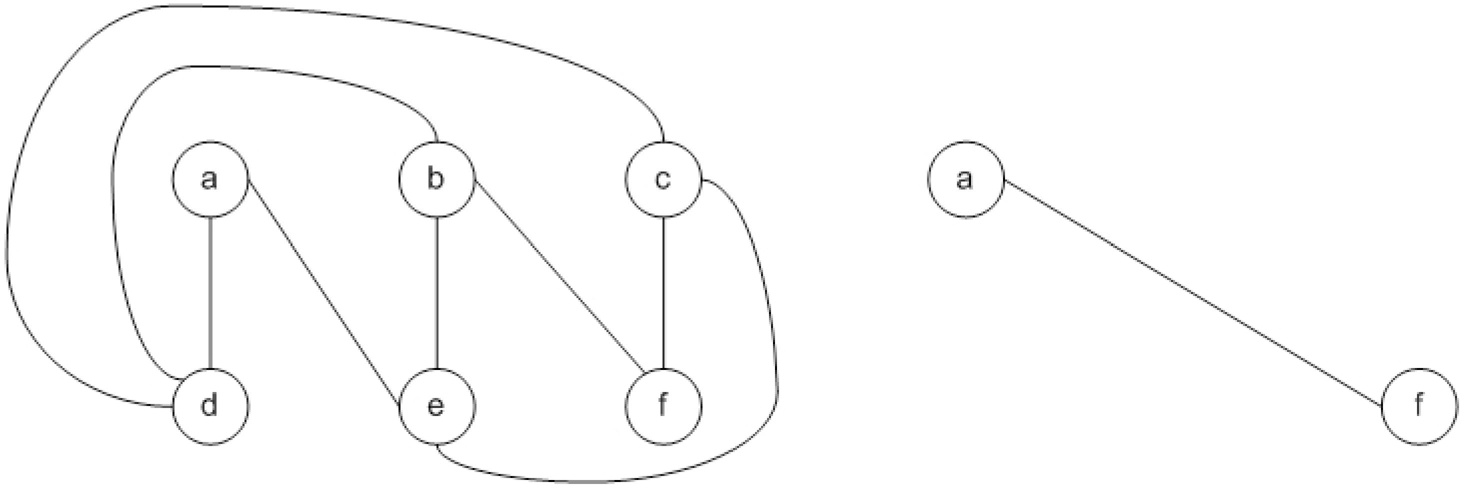
- Графы называются *гомеоморфными*, если графы, полученные из них включением вершин в ребро и удалением вершин степени 2, изоморфны.
- **Пример.** Все эти графы гомеоморфны.



- **Теорема Понтрягина–Куратовского.** *Граф планарен, если и только если он не содержит подграфа, гомеоморфного* K_5 *или* $K_{3,3}$.

7.3. Толщина графа

- *Толщина графа* $t(G)$ – наименьшее число планарных графов, объединение которых дает G.
- **Пример.** Граф $K_{3,3}$ является объединением следующих двух планарных подграфов. Пример показывает, что при удалении любого ребра граф становится планарным.



Толщина графа

- **Теорема о нижней границе толщины графа.** Толщина $t(G)$ графа G удовлетворяет следующим неравенствам:

$$t(G) \geq \left\lceil \frac{q}{3p-6} \right\rceil, t(G) \geq \left\lfloor \frac{q+3p-73}{p-6} \right\rfloor.$$

Раскраска

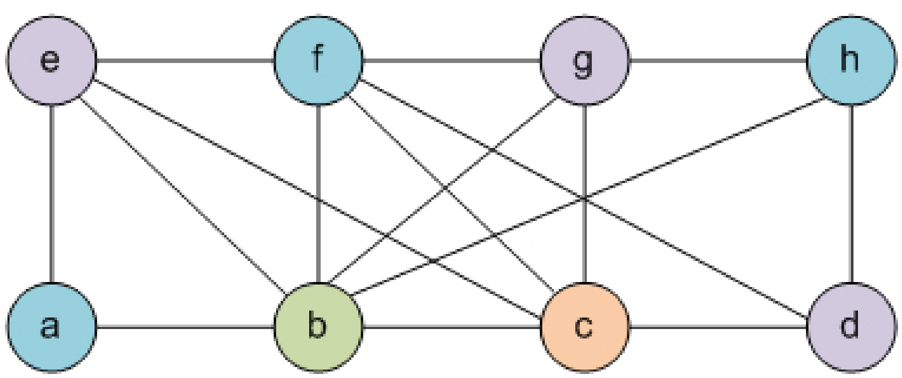
- Хроматическое число
- Алгоритмы раскраски

8.1. Хроматическое число

- *Задача раскрашивания графов* – это задача приписывания вершинам (ребрам) графа некоторых меток, традиционно называемых *цветами*, так, чтобы никакие две смежные вершины (два смежных ребра) не были окрашены в один цвет.

Хроматическое число

- Граф G называется k-*раскрашиваемым*, если каждой его вершине можно приписать один из k цветов таким образом, чтобы никакие две смежные вершины не были одного цвета.
- **Определение.** Если граф G является k-раскрашиваемым, но не является $(k-1)$ -раскрашиваемым, то граф называется k -*хроматическим*, а k – его *хроматическим числом* $\chi(G)$.
- **Пример.** Граф 4-хроматический.



Хроматическое число

- Граф G называется k-*раскрашиваемым*, если каждой его вершине можно приписать один из k цветов таким образом, чтобы никакие две смежные вершины не были одного цвета.
- **Определение.** Если граф G является k-раскрашиваемым, но не является $(k-1)$-раскрашиваемым, то граф называется k-*хроматическим*, а k – его *хроматическим числом* $\chi(G)$.
- **Пример.** Граф 4-хроматический.
- **Теорема 1.** *Если наибольшая из степеней вершин графа равна* ρ, *то граф* $\rho+1$-*раскрашиваем.*
- **Теорема о четырех красках.** Любой планарный граф является 4-раскрашиваемым.