

# Глава 4.

# Цикломатика графов

Свойства графов, которые мы будем изучать в данной главе, присущи графам общего вида и не зависят от ориентации.

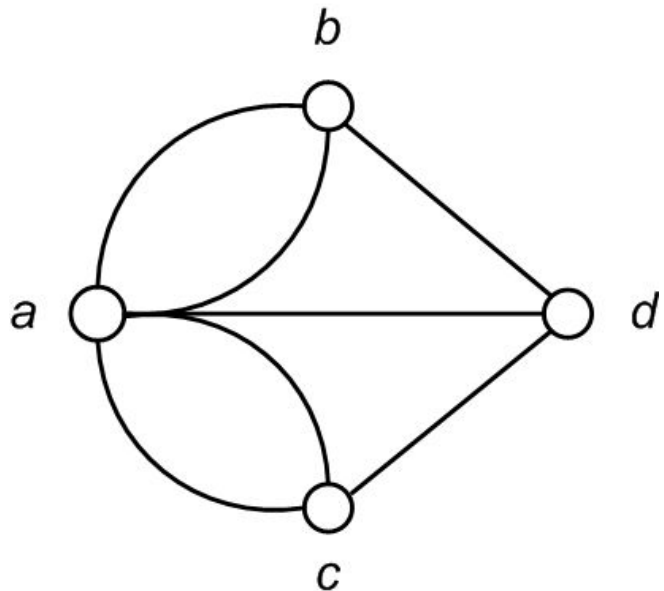
## 4.1. Цикломатическое число

## Определение

**Определение.** *Цикломатическое число (cyclomatic number)*, графа  $G(V, E; P)$  определяется как

$$\lambda(G) = m(G) - n(G) + \kappa(G),$$

где  $n(G) = |V|$  - число вершин,  $m(G) = |E|$  - число ребер,  $\kappa(G)$  - число компонент связности графа.



Пример.

$$n(G) = 4,$$

$$m(G) = 7,$$

$$\kappa(G) = 1,$$

$$\lambda(G) = m - n + \kappa = 7 - 4 + 1 = 4.$$

## Цикловые ребра и перешейки

Назовем ребро **цикловым**, если оно содержится хотя бы в одном цикле; в противном случае назовем его **перешейком (bridge)**.

**Точкой сочленения** (шарниром) называется вершина, при удалении которой число компонент увеличивается.

**Лемма.** Пусть  $G_0$  – суграф, полученный из  $G$  удалением ребра  $u$ . Тогда

$$k(G_0) = \begin{cases} k(G) & \text{если } u \text{ — цикловое ребро} \\ k(G) + 1 & \text{если } u \text{ — перешейок} \end{cases}$$

## Доказательство.

1. Пусть ребро  $u$  соединяет вершины  $x, y$  и входит в некоторый цикл. Если удалить ребро  $u$ , то можно построить цепь соединяющую те же вершины  $y$  и  $x$  в обратном порядке и не содержащую  $u$ . Следовательно, удаление  $u$  не увеличивает числа компонент то есть  $\kappa(G_0) = \kappa(G)$ .

2. Пусть ребро  $u$  не входит ни в какой цикл графа  $G$ . Тогда очевидно, при удалении ребра  $u$  из любой цепи, соединяющей вершины  $x, y$ , в графе  $G_0$  не найдется ни одной цепи, соединяющей эти же вершины и  $\kappa(G_0) = \kappa(G) + 1$ .



## Свойства цикломатического числа

Удаление ребра в графе может привести только к уменьшению цикломатического числа.

Действительно, если  $u$  – цикловое ребро, то

$$\begin{aligned}\lambda(G_0) &= m(G_0) - n(G_0) + \kappa(G_0) = \\ &= m(G) - 1 - n(G) + \kappa(G) = \lambda(G) - 1,\end{aligned}$$

а если  $u$  – перешеек, то

$$\begin{aligned}\lambda(G_0) &= m(G_0) - n(G_0) + \kappa(G_0) = \\ &= m(G) - 1 - n(G) + \kappa(G) + 1 = \lambda(G).\end{aligned}$$

$$\lambda(G_0) = \begin{cases} \lambda(G) & \text{если } u \text{ – перешеек} \\ \lambda(G) - 1 & \text{если } u \text{ – цикловое ребро.} \end{cases}$$

## Теорема.

1.  $\lambda(G) \geq 0$ .

2.  $\lambda(G) = 0$  тогда и только тогда, когда граф  $G$  не содержит циклов.

Доказательство.

1. Будем удалять из графа  $G$  по очереди ребра, при этом цикломатическое число может только уменьшиться. В конце концов получим пустой (безреберный) граф  $G_0$ , для которого  $\lambda(G_0) = m(G_0) - n(G_0) + \kappa(G_0) = 0 - n(G) + n(G) = 0$ .

Отсюда  $\lambda(G) \geq \lambda(G_0) = 0$ .

2. Пусть в графе нет циклов (все ребра – перешейки). Тогда, удаляя их, мы не изменяя  $\lambda$ , приходим к пустому графу, для которого  $\lambda(G_0) = 0$ . Наоборот, если  $\lambda(G) = 0$ , удалять мы можем только перешейки, так как иначе  $\lambda$  стало бы отрицательным.



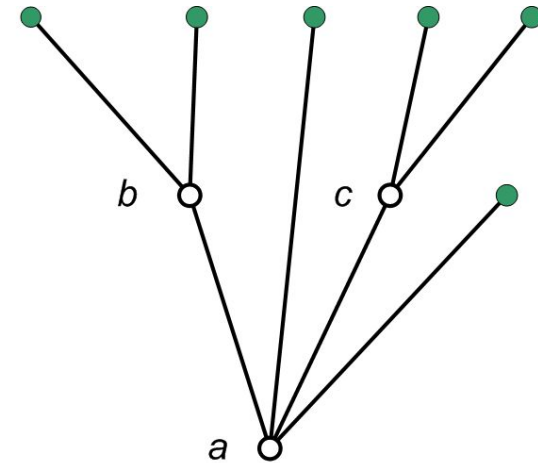
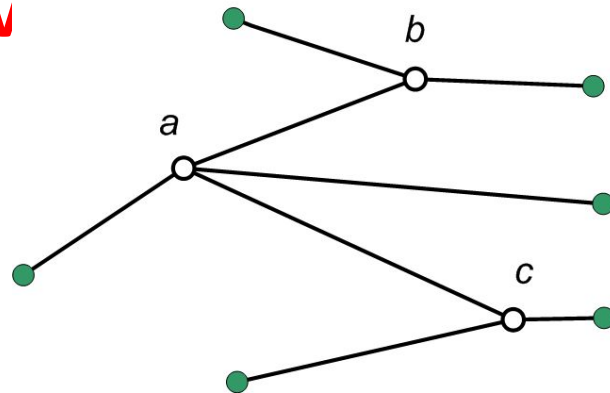


## 4.2. Деревья

## Определения

Связный граф без циклов называется **деревом (tree)**. Висячие вершины дерева называются **ЛИСТЬЯМИ (leaf - leaf)**

$$T = (V, E)$$



Граф, все компоненты связности которого – деревья, называется **лесом (forest)**.

## Свойства дерева

1.  $\kappa(T) = 1$  &  $\lambda(T) = 0$  (определение дерева).
2.  $\lambda(T) = 0$  &  $m = n - 1$  (из определения).
3. Каждое ребро в  $T$  – перешеек (т.е. при удалении любого ребра  $\kappa$  увеличивается на единицу).
4. Для любых двух вершин  $x, y$  дерева  $T$  существует одна и только одна соединяющая их цепь, и эта цепь простая.
5. Соединение любых двух вершин  $x, y$  дерева  $T$  новым ребром приводит к появлению цикла.
6. Вершина  $x$  дерева  $T$  является точкой сочленения (шарниром) тогда и только тогда, когда ее степень  $s(x) > 1$ .
7. Если  $n(T) \geq 2$ , то в  $T$  есть по крайней мере две висячие вершины.

Доказательство.

**Свойства 1, 2, 3** непосредственно следуют из определения.

**Свойство 4.** (Доказательство от противного). Предположим, что существуют две различные цепи

$x_0 u_1 x_1 u_2 x_2 \dots x_{l-1} u_l x_l$  и  $x_0 v_1 y_1 v_2 y_2 \dots y_{k-1} v_k x_l$ ,

соединяющие вершину  $x = x_0$  с вершиной  $y = x_l$ , причем

$l \geq k$ . При этом первая из этих цепей имеет ненулевую

длину. Тогда ребро  $u_i$ , где  $i = \min\{j \mid u_j \neq v_j\}$

(если  $l > k$  и  $u_j = v_j$  при  $j = 1, 2, \dots, k$ , то положим

$i = k + 1$ ), является цикловым, так как после его удаления

из  $G$  вершины  $x_{i-1}$  и  $x_i$  останутся соединенными маршрутом

$x_{i-1} v_i y_i \dots y_{k-1} v_k x_l u_l x_{l-1} \dots x_{i+1} u_{i+1} x_i$ .

Следовательно, граф имеет циклы, что противоречит определению дерева.

**Свойство 5.** Добавление ребра (разумеется, без добавления вершины) приводит к увеличению  $\lambda$  на единицу, то есть к появлению цикла.

**Свойство 6** проверяется непосредственно: все вершины дерева, кроме висячих, являются точками сочленения.

**Свойство 7.** В любом графе с числом вершин  $\geq 2$  есть по крайней мере две вершины, не являющиеся точками сочленения. Для дерева это висячие вершины.

## 4.3. Қаркасы

## Определение

**Определение.** Всякий суграф  $T$  графа  $G$ , у которого  $\kappa(T) = \kappa(G)$ ,  $\lambda(T) = 0$  называется **каркасом** графа  $G$  (синонимы: остов, стягивающее дерево, **spanning tree, ST**).

Каркас связного графа является деревом.

Поскольку  $T$  – суграф, то  $n(G) = n(T)$  и

$$\begin{aligned}\lambda(G) &= m(G) - n(G) + \kappa(G), \\ \lambda(T) = 0 &= m(T) - n(T) + \kappa(T).\end{aligned}$$

Отсюда  $m(T) = m(G) - \lambda(G)$ .

## Алгоритм нахождения каркаса

Алгоритм является модификацией волнового алгоритма нахождения кратчайшей цепи.

**Шаг 1.** Выберем произвольную вершину и пометим ее меткой 0.

**Шаг 2.** Все непомеченные вершины, смежные с вершинами, имеющими метку  $k$ , помечаем меткой  $k + 1$ . Разметка продолжается до тех пор, пока все вершины не будут помечены. При такой разметке метки смежных вершин не могут отличаться более чем на единицу.



**Шаг 3.** После окончания разметки будем просматривать вершины в любом порядке и удалять некоторые ребра по следующим правилам:

- если в данный момент мы находимся в вершине  $x$  с меткой  $l(x)$ , то удаляем все ребра, которые соединяют  $x$  с вершинами, имеющими ту же метку;
- из ребер, соединяющих  $x$  с вершинами, имеющими метку  $l(x) - 1$ , удаляем все, кроме любого одного.

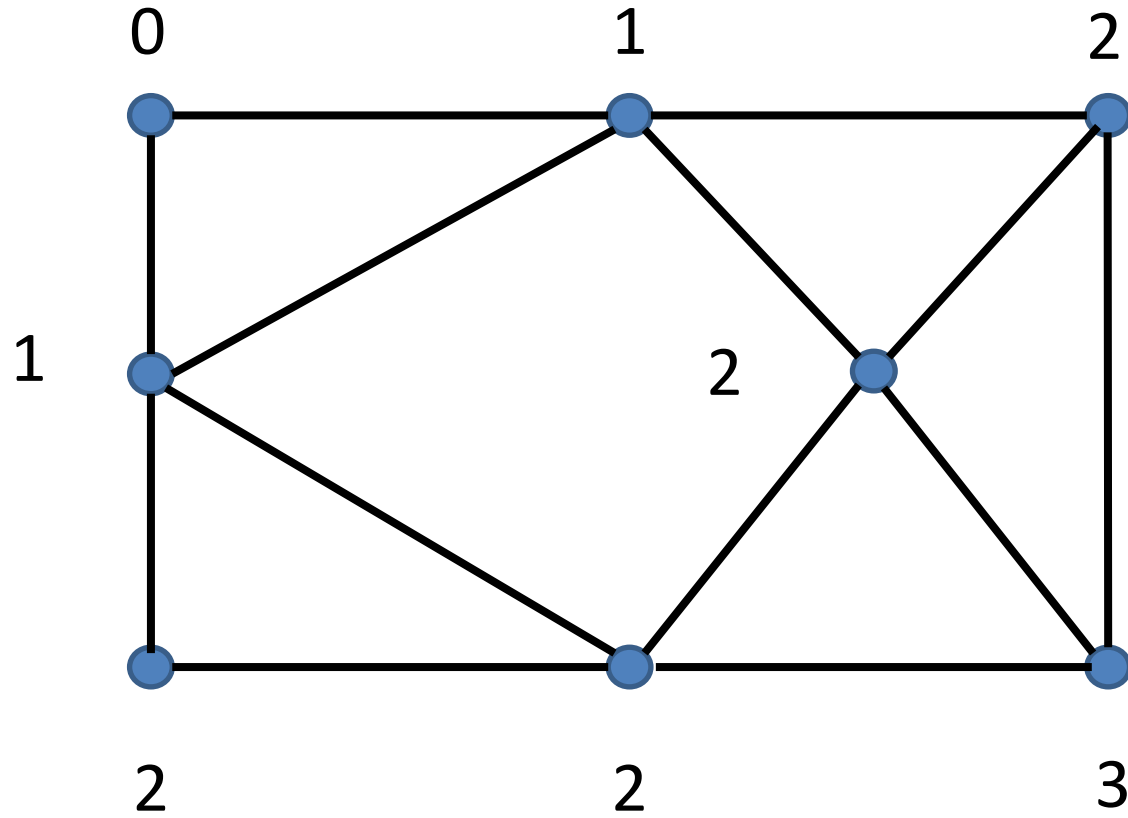
Покажем, что суграф  $T$ , оставшийся после такого удаления ребер, является каркасом графа  $G$ .

1.  $\kappa(T) = \kappa(G)$ , так как правила удаления таковы, что разметка сохраняется и для суграфа  $T$ . Поэтому число вершин, соединимых с вершиной, имеющей метку 0, одинаково в  $T$  и в  $G$ .

2. Разметка такова, что ребро между двумя вершинами с одинаковыми метками – цикловое. Далее, мы отыскиваем для каждой вершины единственную цепь, соединяющую эту вершину с вершиной, имеющей метку 0, отсекая все другие возможные цепи.



# Пример



## Кратчайший каркас графа

Рассмотрим связный обыкновенный граф со взвешенными ребрами  $G = (V, E)$ ; вес ребра  $\{x_i, x_j\}$  обозначим через  $c_{ij}$ . Если ребра нет, то  $c_{ij} = \infty$ .

Задача заключается в том, чтобы из всех каркасов графа найти такой, что сумма весов его ребер – наименьшая.

Такой каркас называется **кратчайшим (shortest spanning tree – SST)**. Задача нахождения SST возникает, если какие-либо пункты нужно связать кратчайшей сетью коммуникаций (трубопроводов, электрических проводов, дорог и т.д.). Задача нахождения кратчайшего каркаса – это одна из немногих задач теории графов, которую можно считать полностью решенной.

## Алгоритм Прима

Идея алгоритма состоит в постепенном выращивании кратчайшего стягивающего дерева из любой начальной вершины.

Пусть на некоторой итерации имеется дерево  $T_k$ . Множество его вершин обозначим  $V_k$ . Тогда  $V \setminus V_k$  - множество вершин графа, не принадлежащих дереву  $T_k$ . Найдем кратчайшее ребро, соединяющее множества  $V_k$  и  $V \setminus V_k$  и добавим это ребро к дереву  $T_k$ .

Многократно выбирая кратчайшие ребра, в конце концов получим SST.

Robert C. Prim (b. 1921) along with coworker [Joseph Kruskal](#) developed two different algorithms (see [greedy algorithm](#)) for finding a [minimum spanning tree](#) in a weighted [graph](#), a basic stumbling block in [computer network design](#). His self-named algorithm, [Prim's algorithm](#), was originally discovered in 1930 by mathematician [Vojtěch Jarník](#) and later independently by Prim in 1957. It was later rediscovered by [Edsger Dijkstra](#) in 1959. It is sometimes referred to as the *DJP algorithm* or the *Jarník algorithm*

[en.wikipedia.org/wiki/Robert\\_C.\\_Prim](https://en.wikipedia.org/wiki/Robert_C._Prim)



Алгоритм Прима нахождения SST очень похож на алгоритм Дейкстры нахождения кратчайшей цепи в графе. Точно так же происходит разметка вершин временными и постоянными метками, однако правило обновления меток несколько отличается: при обнаружении вершины  $u$ , для которой  $l(y) > l(x)$ ,  $l(y) = c(x, y)$ . В результате постоянная метка при вершине дает длину кратчайшего ребра, соединяющего данную вершину с уже построенным каркасом.

Еще одно отличие состоит в том, что построение кратчайшего каркаса происходит не после окончания, а в процессе разметки. При превращении временной метки в постоянную сразу же происходит добавление очередного ребра к растущему каркасу. На это ребро указывает обратная навигационная ссылка внутри метки.

## Шаг 0. Инициализация.

Назначить любой исходной вершине  $s$  метку  $l(s) = 0$  и считать ее постоянной. Вершины с постоянными метками считаются обработанными и к ним алгоритм не возвращается. Для всех  $x \neq s$  назначить метки  $l(x) = \infty$  и считать эти метки временными.

Текущая вершина  $p = s$ .

## Шаг 1. Обновление соседних временных меток из текущей вершины.

Взять текущую вершину  $p$ .

Рассмотреть все смежные с ней вершины  $x$  с временными метками и, если  $l(x) > c(x, p)$ , то обновить их по правилу:  $l(x) = c(x, p)$ . При обновлении меняется и навигационная метка: она дает ссылку на вершину  $p$ , из которой произошло обновление.



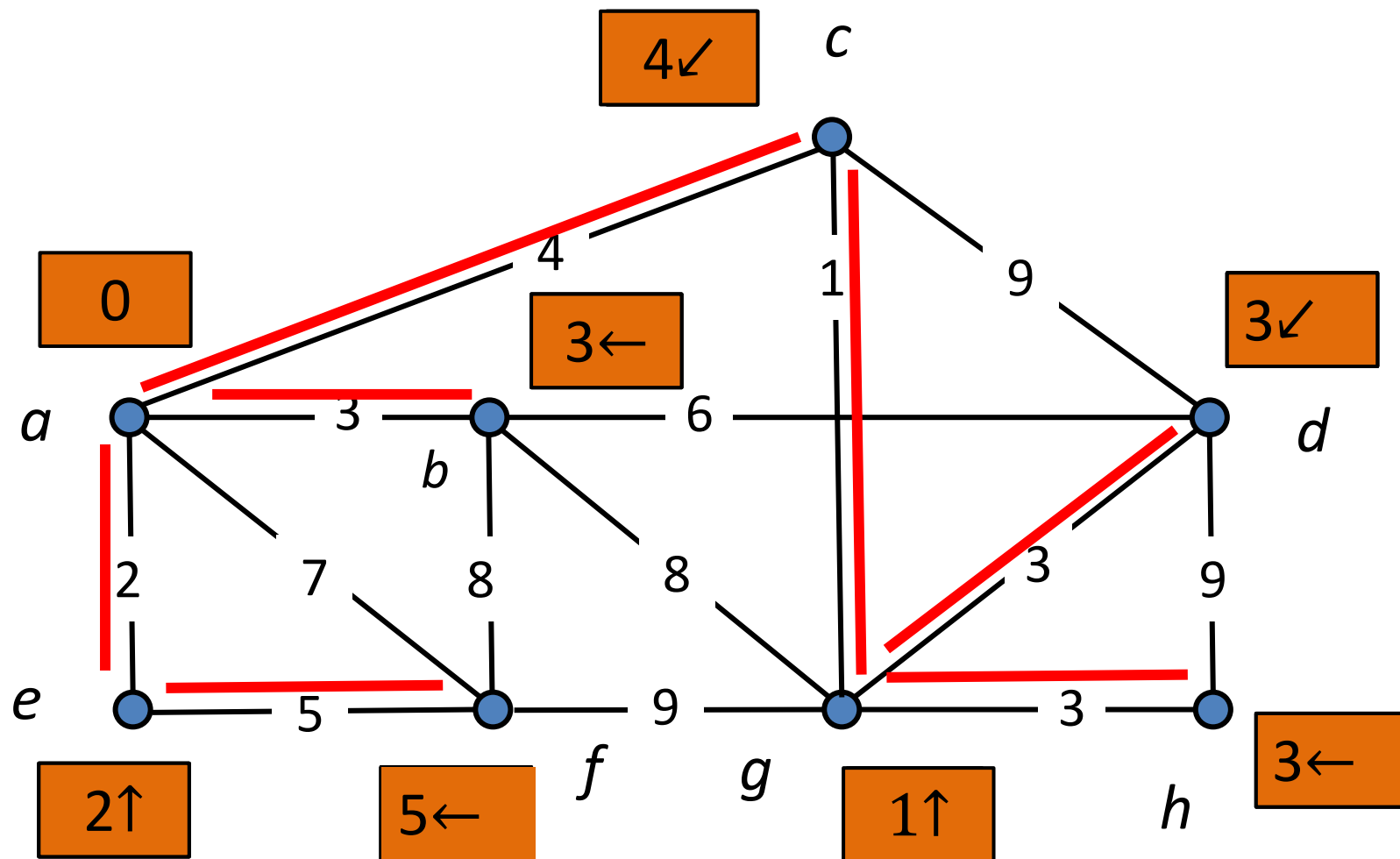
## Шаг 2. Превращение метки в постоянную.

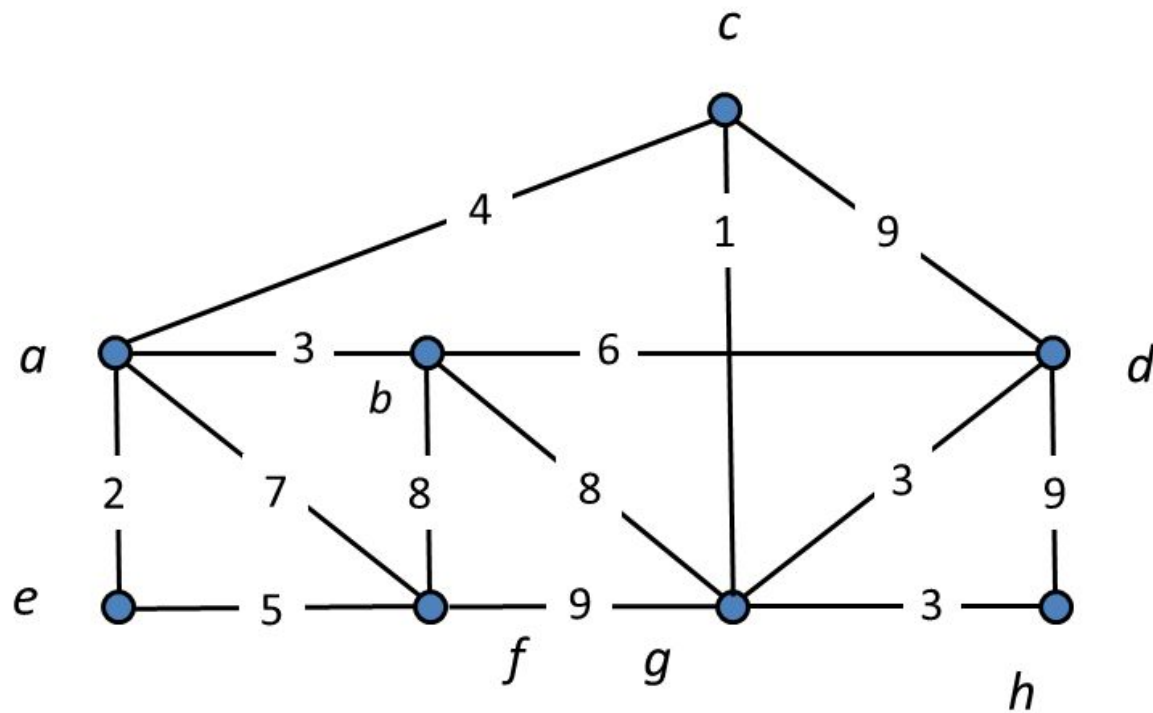
- а) Среди всех вершин с временными метками найти вершину  $x^*$  с наименьшей меткой.
- б) Считать эту метку постоянной.
- в) Считать эту вершину текущей  $p := x^*$ .
- г) Ребро, указанное навигационной меткой, включить в каркас.
- г) Если есть еще временные метки, возврат на **шаг 1**.
- д) Если все метки постоянные – переход на **шаг 3**.

**Шаг 3.** Найден кратчайший каркас.



# Пример





	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>h</i>	<i>g</i>	<i>f</i>	<i>e</i>	<i>E</i>	<i>C</i>
<i>a</i>	$[0, a]$	$[3, a]$	$[4, a]$	$[\infty, d]$	$[\infty, h]$	$[\infty, g]$	$[7, a]$	$[2, a]$	$(a, e)$	2
<i>e</i>	$\times$	$[3, a]$	$[4, a]$	$[\infty, d]$	$[\infty, h]$	$[\infty, g]$	$[5, e]$	$\times$	$(a, b)$	3
<i>b</i>	$\times$	$\times$	$[4, a]$	$[6, b]$	$[\infty, h]$	$[8, b]$	$[5, e]$	$\times$	$(a, c)$	4
<i>c</i>	$\times$	$\times$	$\times$	$[6, b]$	$[\infty, h]$	$[1, c]$	$[5, e]$	$\times$	$(c, g)$	1
<i>g</i>	$\times$	$\times$	$\times$	$[3, g]$	$[3, g]$	$\times$	$[5, e]$	$\times$	$(g, d)$	3
<i>d</i>	$\times$	$\times$	$\times$	$\times$	$[3, g]$	$\times$	$[5, e]$	$\times$	$(g, h)$	3
<i>h</i>	$\times$	$\times$	$\times$	$\times$	$\times$	$\times$	$[5, e]$	$\times$	$(e, f)$	5

## Теорема о хорде каркаса

Ребра графа  $G$ , не принадлежащие его каркасу  $T$ , называются **хордами** каркаса  $T$  в  $G$ .  
Число хорд каркаса равно цикломатическому числу графа. Действительно:

$$m(G) - n(G) + \kappa(G) = \lambda(G)$$

$$m(T) - n(T) + \kappa(T) = 0$$

Вычитая из первого равенства второе и учитывая, что  $n(G) = n(T)$ ,  $\kappa(G) = \kappa(T)$ , получаем

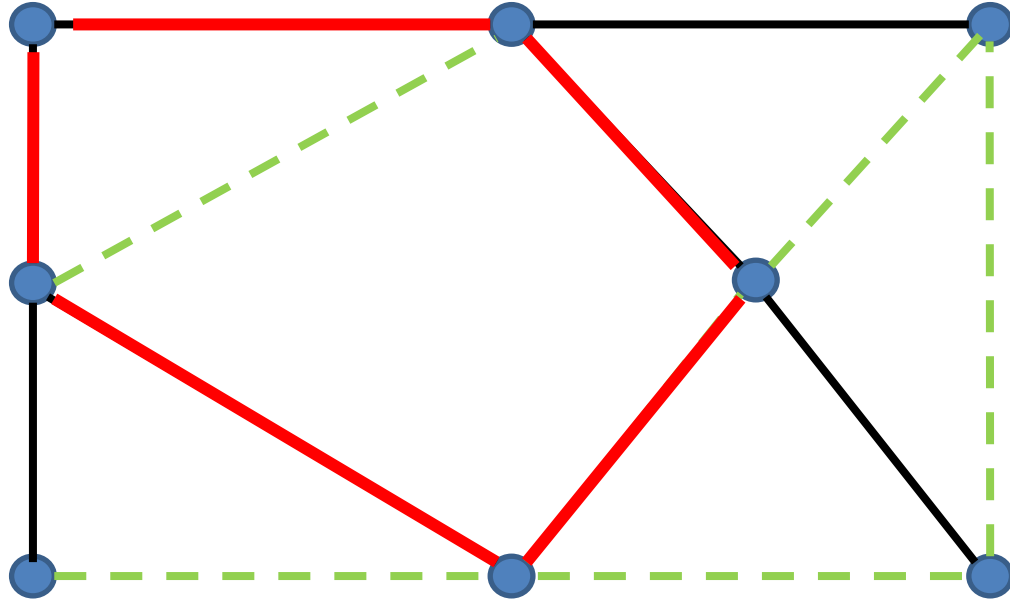
$$m(G) - m(T) = \lambda(G)$$

**Теорема.** Пусть  $T$  – некоторый произвольный каркас графа  $G$ ,  $u$  – произвольная хорда этого каркаса. Тогда в графе  $G$  существует цикл, содержащий  $u$  и не содержащий других хорд каркаса  $T$ , причем этот цикл простой и единственный.

**Доказательство.** Доказательство проводится для связного графа, т.к. для несвязного графа можно провести доказательство для каждой из компонент. Пусть  $x, y$  – те вершины графа  $G$ , которые соединяет ребро  $u$ . Так как  $T$  – дерево, то в нем имеется единственная цепь, притом простая, соединяющая  $x$  с  $y$  (свойство дерева) и эта цепь вместе с хордой  $u$  образует тот самый единственный простой цикл.

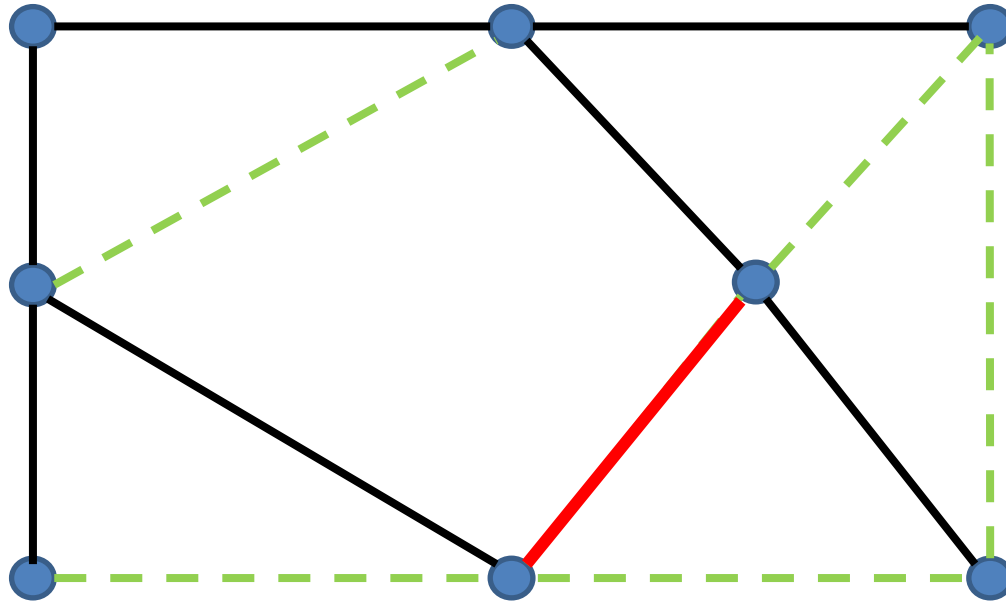


# Пример



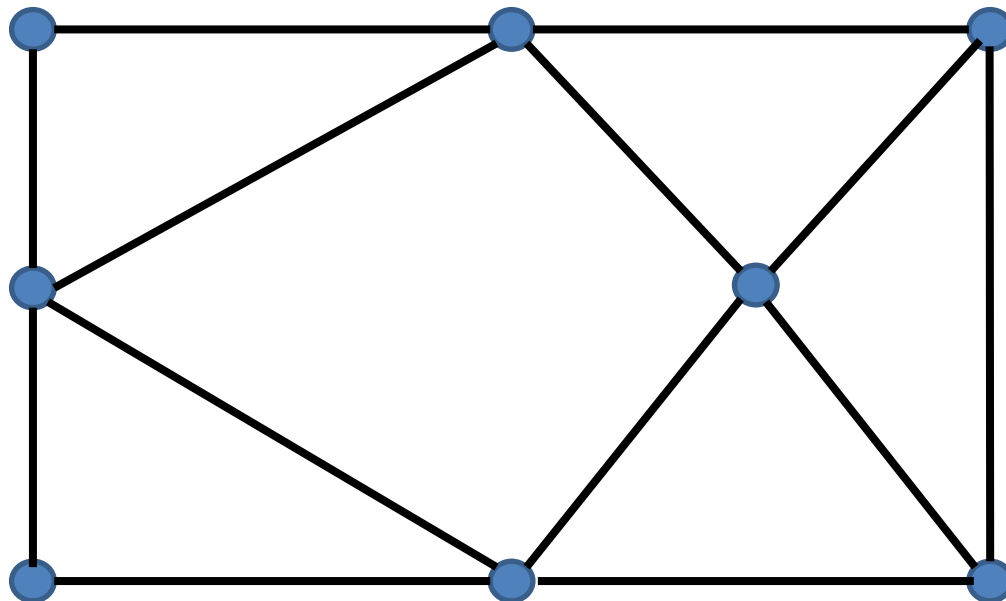
$$\lambda = m - n + \kappa = 13 - 8 + 1 = 6$$

Эта теорема имеет два полезных следствия.  
Во-первых, она дает возможность перебирать  
каркасы.



$$\lambda = m - n + k = 13 - 8 + 1 = 6$$

Во-вторых, она утверждает, что число **независимых** циклов в графе равно цикломатическому числу.



$$\lambda = m - n + k = 13 - 8 + 1 = 6$$



Каждому суграфу  $G'=(V,E')$  графа  $G=(V,E)$  взаимно однозначно

соответствует подмножество ребер,  $E' \subseteq E$ ; каждое  $E'$

задается вектором  $a_i = \begin{cases} 1 & \text{если } u_i \in E' \\ 0 & \text{если } u_i \notin E' \end{cases}$  которого

Далее суграфом называют и  $E \quad \vec{a}(E')$ .

И определяется операция сложения суграфов  $E_1, E_2$  :

$E_1 \oplus E_2 = (E_1 \cup E_2) \setminus (E_1 \cap E_2)$  (симметрическая разность).

Для векторов, соответственно, покомпонентное сложение по модулю два (исключающее **или**)

$$(a_1, a_2, \dots, a_m) \oplus (b_1, b_2, \dots, b_m) = (a_1 \oplus b_1, a_2 \oplus b_2, \dots, a_m \oplus b_m)$$

$$0 \oplus 0 = 0, \quad 1 \oplus 0 = 1, \quad 0 \oplus 1 = 1, \quad 1 \oplus 1 = 0$$

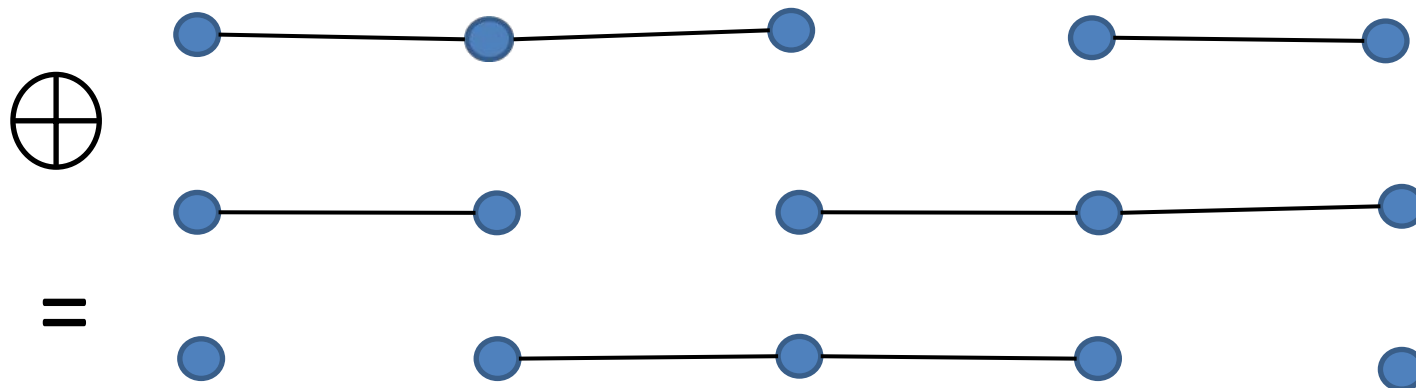
# Однорреберные

суграфы  $\vec{e}_1 = (1, 0, \dots, 0), \vec{e}_2 = (0, 1, \dots, 0), \dots, \vec{e}_m = (0, 0, \dots, 1)$

линейно независимы, а любой суграф есть их линейная комбинация  $(a_1, a_2, \dots, a_m) = a_1 \vec{e}_1 \oplus a_2 \vec{e}_2 \oplus \dots \oplus a_m \vec{e}_m$ .

Множество суграфов данного графа  $G$  образует линейное пространство со сложением  $\oplus$ , и обычным умножением; нуль-вектор  $\vec{0} = (0, 0, \dots, 0)$  соответствует  $E' = \emptyset$ ; обратный элемент для каждого суграфа – он сам:  $\vec{a} \oplus \vec{a} = \vec{0}$ . Базис пространства суграфов размерности  $m$  есть множество однорреберных суграфов.

Пример: сумма суграфов  $(1, 1, 0, 1) \oplus (1, 0, 1, 1) = (0, 1, 1, 0)$



В этом разделе цикл определяется как суграф, все вершины которого имеют четные валентности. Сумма двух циклов есть цикл

(не обязательно цикл в обычном понимании: это могут быть два

Пусть теперь  $T$  -- некоторый каркас графа  $G$ . Каждой хорде цикла без общих вершин). Множество циклов образует

этого линейное

каркаса поставим в соответствие тот единственный цикл, пространство, который

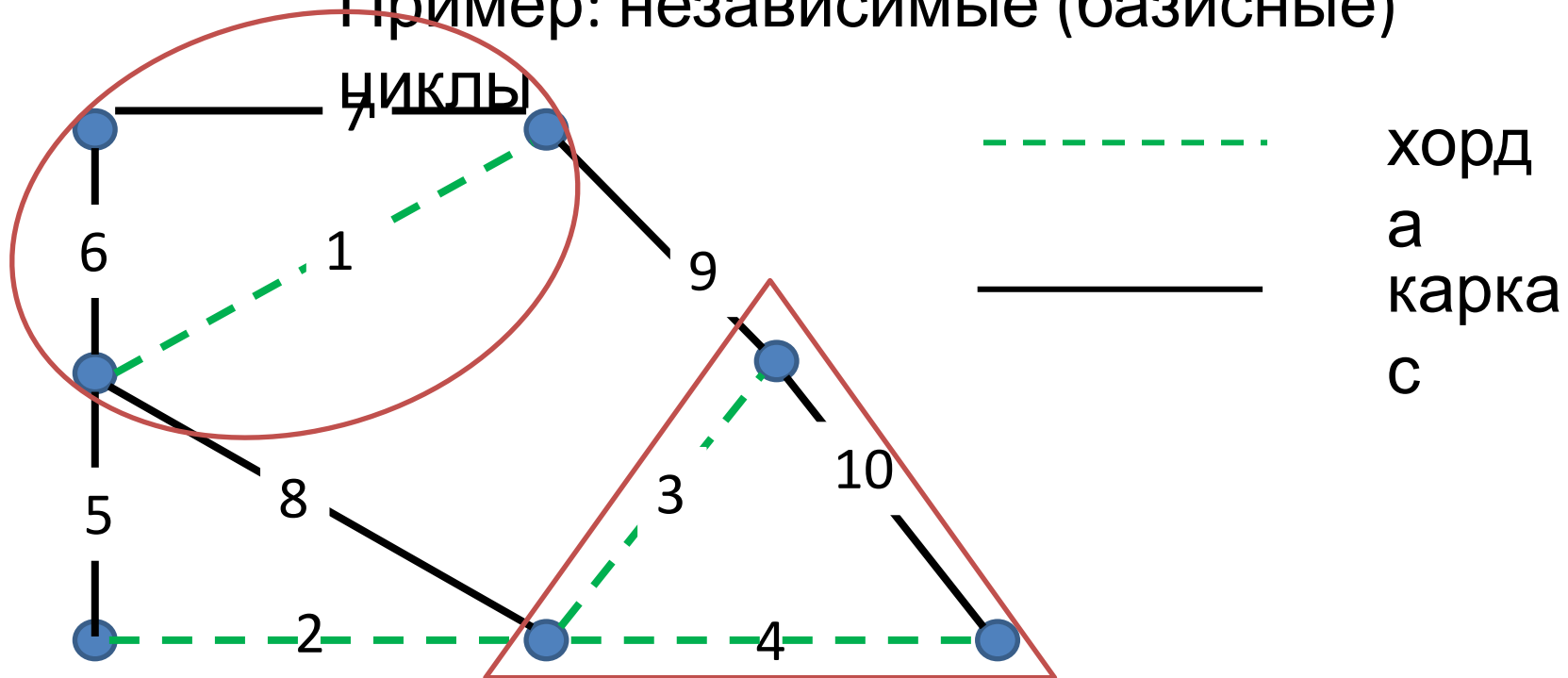
она образует вместе с некоторыми ребрами  $T$ , и тем самым полу-

чим систему из  $\lambda(G)$  циклов. Элементы этой системы линейно не-

зависимы и образуют базис пространства циклов размерности  $\lambda(G)$ .

Любой другой цикл может быть получен как линейная комбинация

# Пример: независимые (базисные)



	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	1	1	0	0	0
2	0	1	0	0	1	0	0	1	0	0
3	0	0	1	0	0	1	1	1	1	0
4	0	0	0	1	0	1	1	1	1	1

Цикломатическая матрица

$$C_1 \oplus C_3 \oplus C_4 = (1, 0, 1, 1, 0, 1, 1, 0, 0, 1)$$