Глава 4.
Цикломатика графов

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

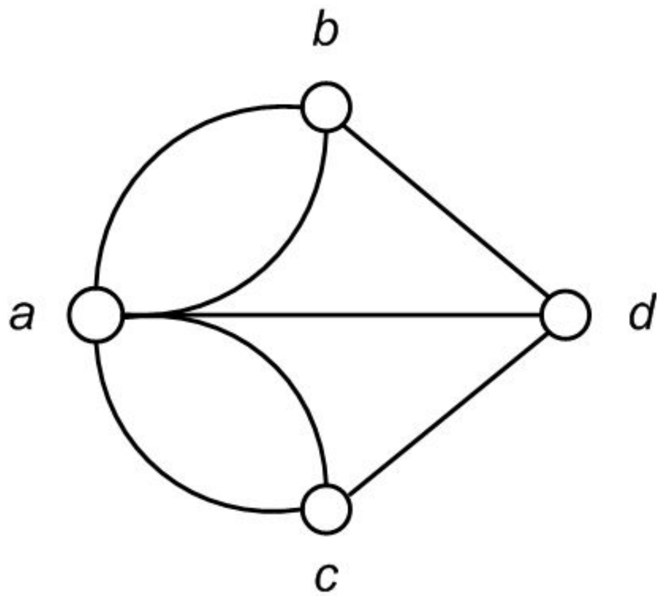
4.1. Цикломатическое число

Определение

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

$$\boxed{\lambda(G) = m(G) - n(G) + \varkappa(G),}$$

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



Пример.

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

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

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

Д о к а з а т е л ь с т в о .

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 $\varkappa(G_0) = \varkappa(G)$.

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 $\varkappa(G_0) = \varkappa(G) + 1$.

Свойства цикломатического числа

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

Действительно, если u – цикловое ребро, то

$$\lambda(G_0) = m(G_0) - n(G_0) + \varkappa(G_0) =$$
$$= m(G) - 1 - n(G) + \varkappa(G) = \lambda(G) - 1,$$

а если u – перешеек, то

$$\lambda(G_0) = m(G_0) - n(G_0) + \varkappa(G_0) =$$
$$= m(G) - 1 - n(G) + \varkappa(G) + 1 = \lambda(G).$$

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

Теорема.

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

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

Д о к а з а т е л ь с т в о .

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 $\lambda(G_0) = m(G_0) - n(G_0) + \varkappa(G_0) = 0 - n(G) + n(G) = 0$. Отсюда $\lambda(G) \geq \lambda(G_0) = 0$.

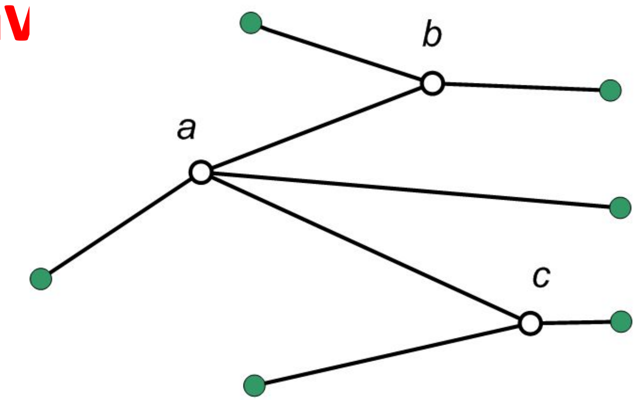
2. Пусть в графе нет циклов (все ребра – перешейки). Тогда, удаляя их, мы не изменяя λ, придем к пустому графу, для которого $\lambda(G_0) = 0$. Наоборот, если $\lambda(G) = 0$, удалять мы можем только перешейки, так как иначе λ стало бы отрицательным.

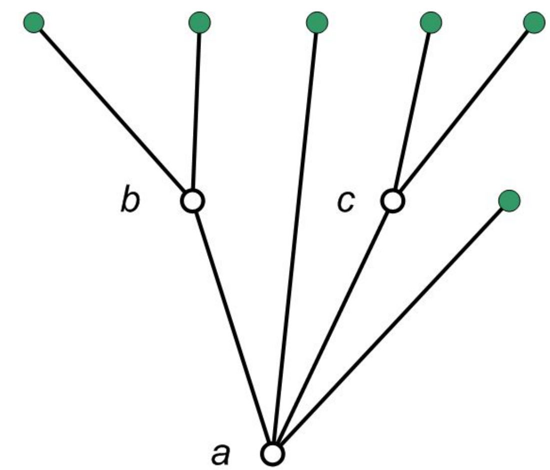
4.2. Деревья

Определения

Связный граф без циклов называется **дерево** **(tree)**. Висячие вершины дерева называются **листьями**
(leaf - leav

$$T = (V, E)$$





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

Свойства дерева

1. $\varkappa(T) = 1$ & $\lambda(T) = 0$ (определение дерева).
2. $\lambda(T) = 0$ & $m = n - 1$ (из определения).
3. Каждое ребро в T – перешеек (т.е. при удалении любого ребра $\varkappa$ 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

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

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 $j = 1, 2, \ldots, k$, то положим $i = k + 1$), является цикловым, так как после его удаления из G вершины x_{i-1} и x_i останутся соединенными маршрутом

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

Следовательно, граф имеет циклы, что противоречит определению дерева.

Свойство 5. Добавление ребра (разумеется, без добавления
вершины) приводит к увеличению λ 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 ≥2 есть по крайней
мере две вершины, не являющиеся точками сочленения.
Для дерева это висячие вершины.

4.3. Каркасы

Определение

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

Каркас связного графа является деревом.

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

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

$$\lambda(T) = 0 = m(T) - n(T) + \varkappa(T).$$

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;

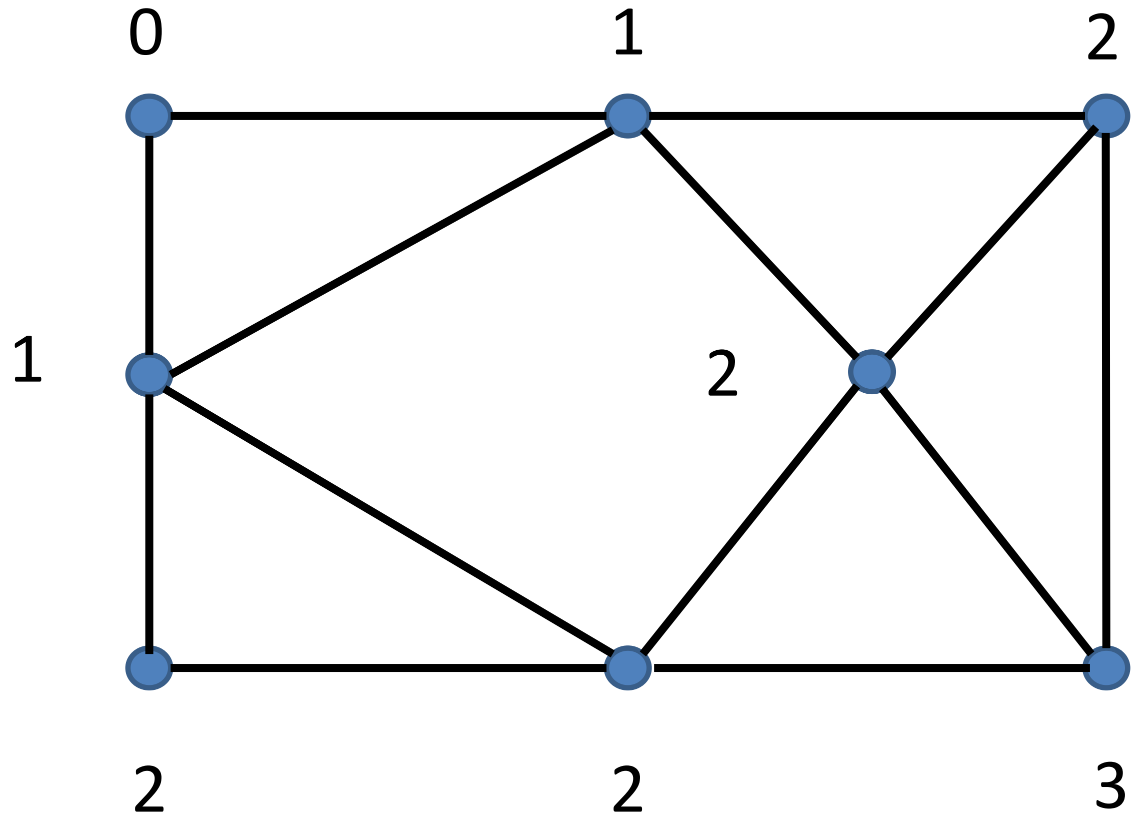
• из ребер, соединяющих x с вершинами, имеющими метку $l(x) - 1$, удаляем все, кроме любого одного.

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

1. $\varkappa(T) = \varkappa(G)$, 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 spannig 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

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 y, для которой $l(y) > l(x), \quad l(y) = c(x, y)$. 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 $p := x^*$.

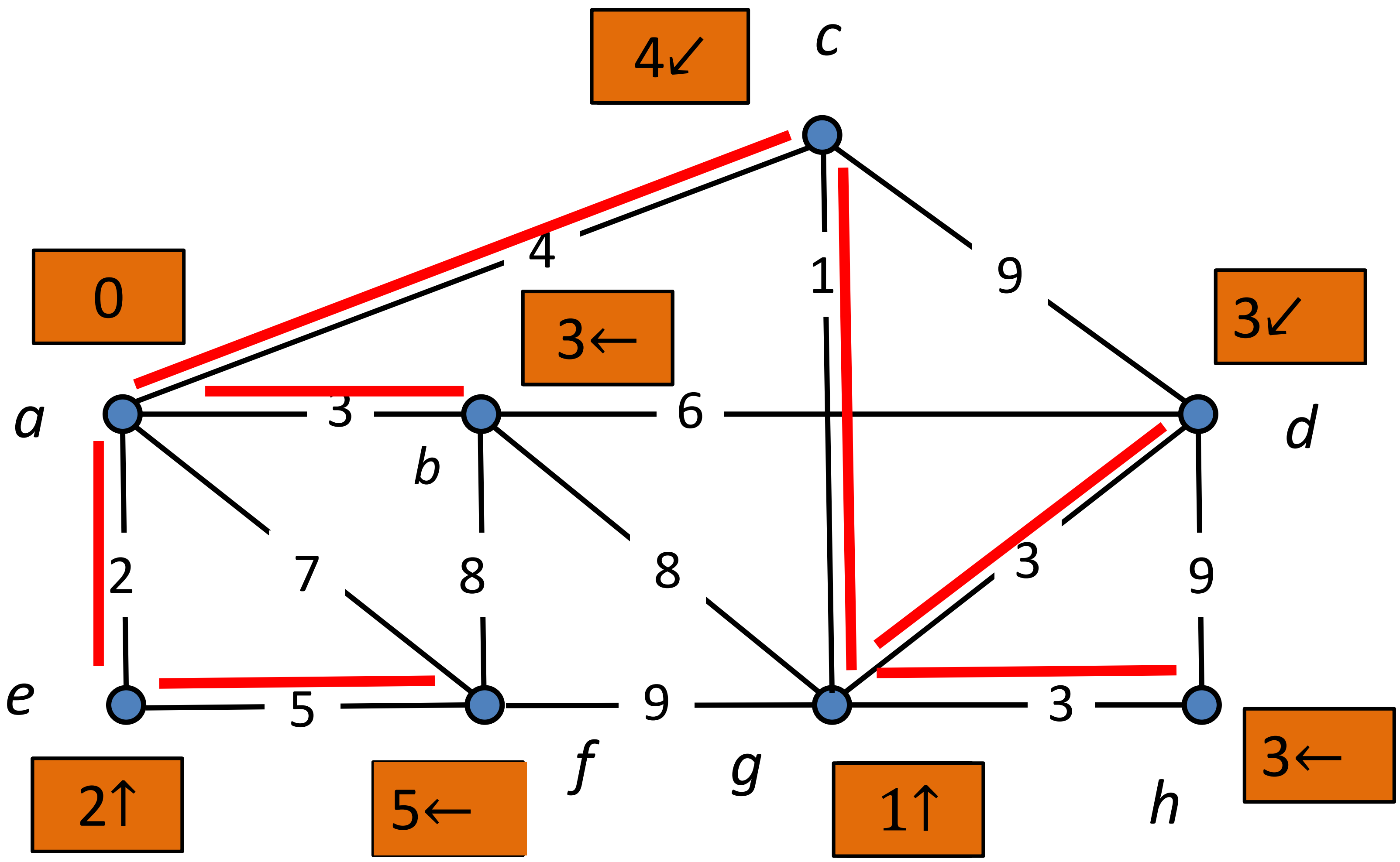
г) Ребро, указанное навигационной меткой, включить в каркас.

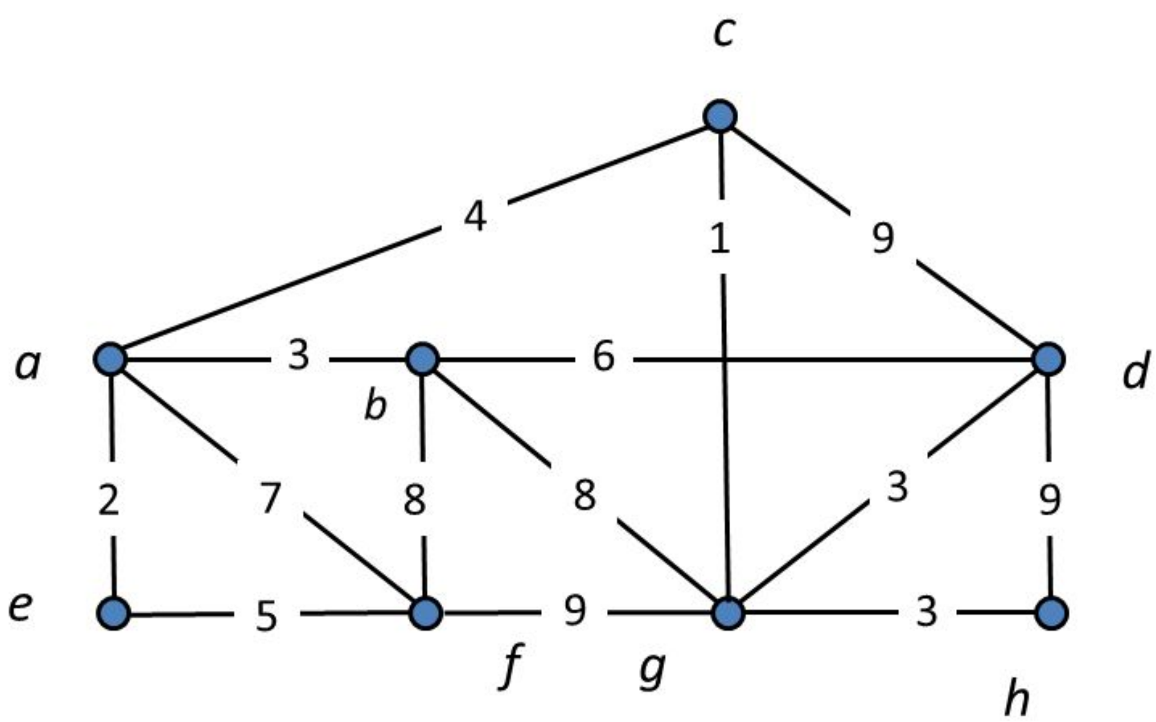
г) Если есть еще временные метки, возврат на **шаг 1**.

д) Если все метки постоянные – переход на **шаг 3.**

Шаг 3. Найден кратчайший каркас.

Пример





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

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

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

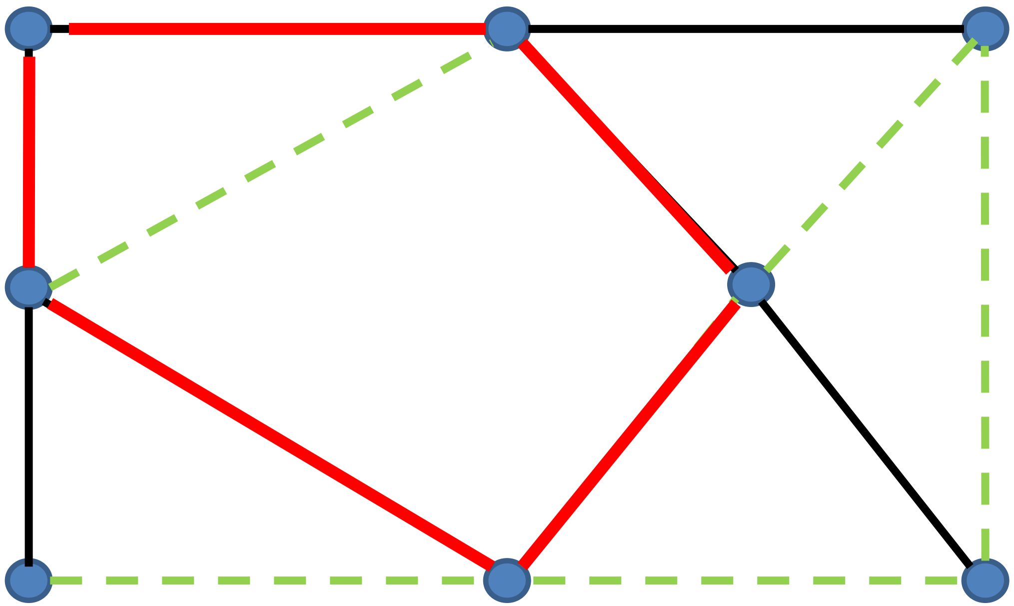
Вычитая из первого равенства второе и учитывая, что $n(G) = n(T)$, $\varkappa(G) = \varkappa(T)$, получаем

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

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.

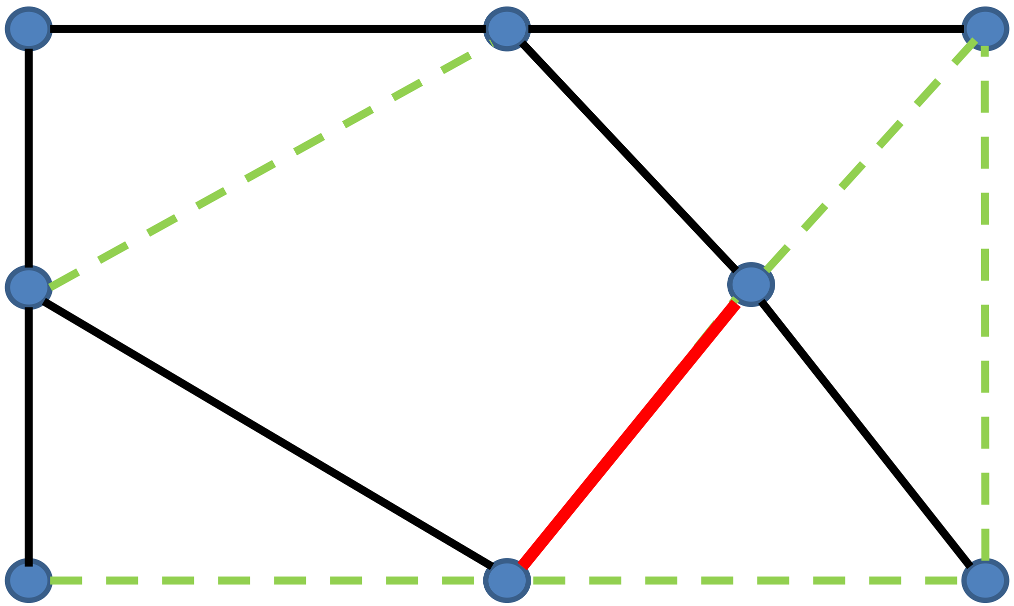
Д о к а з а т е л ь с т в о . Доказательство проводится для связного графа, т.к. для несвязного графа можно провести доказательство для каждой из компонент. Пусть x, y – те вершины графа G, которые соединяет ребро u. Так как T – дерево, то в нем имеется единственная цепь, притом простая, соединяющая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

Пример



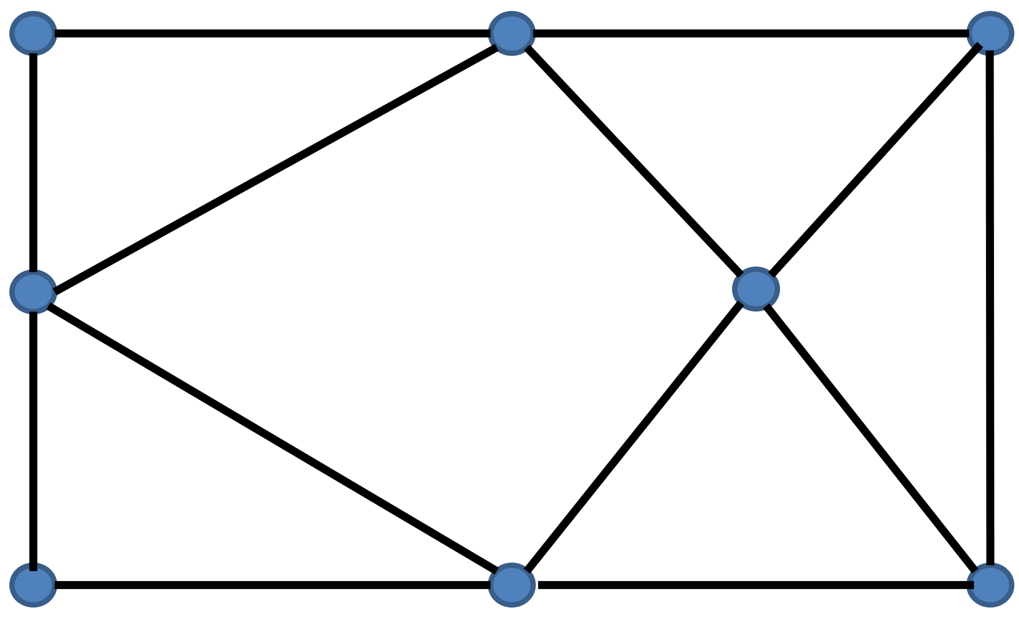
$$\lambda = m - n + \varkappa = 13 - 8 + 1 = 6$$

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



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

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



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

Каждому суграфу $G'=(V,E')$ графа $G=(V,E)$ взаимно однозначно соответствует подмножество ребер $E' \subseteq E$; каждое E' задается вектором $\mathbf{a}(G') = \mathbf{a}(E') = (a_1, a_2, \ldots, a_m)$, у которого

$$a_i = \begin{cases} 1, & \text{если} \quad u_i \in E' \\ 0, & \text{если} \quad u_i \notin E' \end{cases}$$

Далее суграфом называют и E', и $\mathbf{a}(E')$.

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)

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

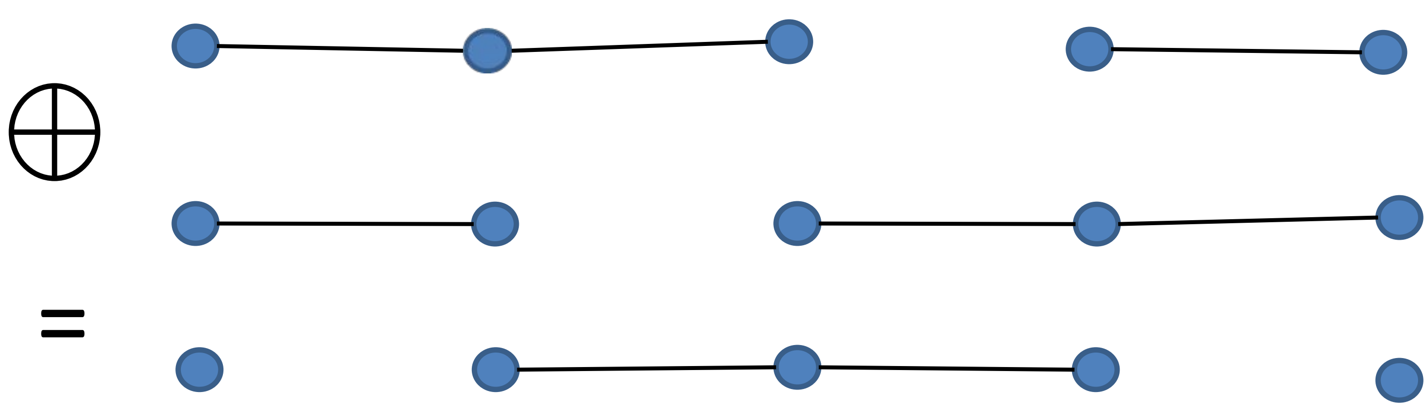
$0 \oplus 0 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$, $1 \oplus 1 = 0$

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

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

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

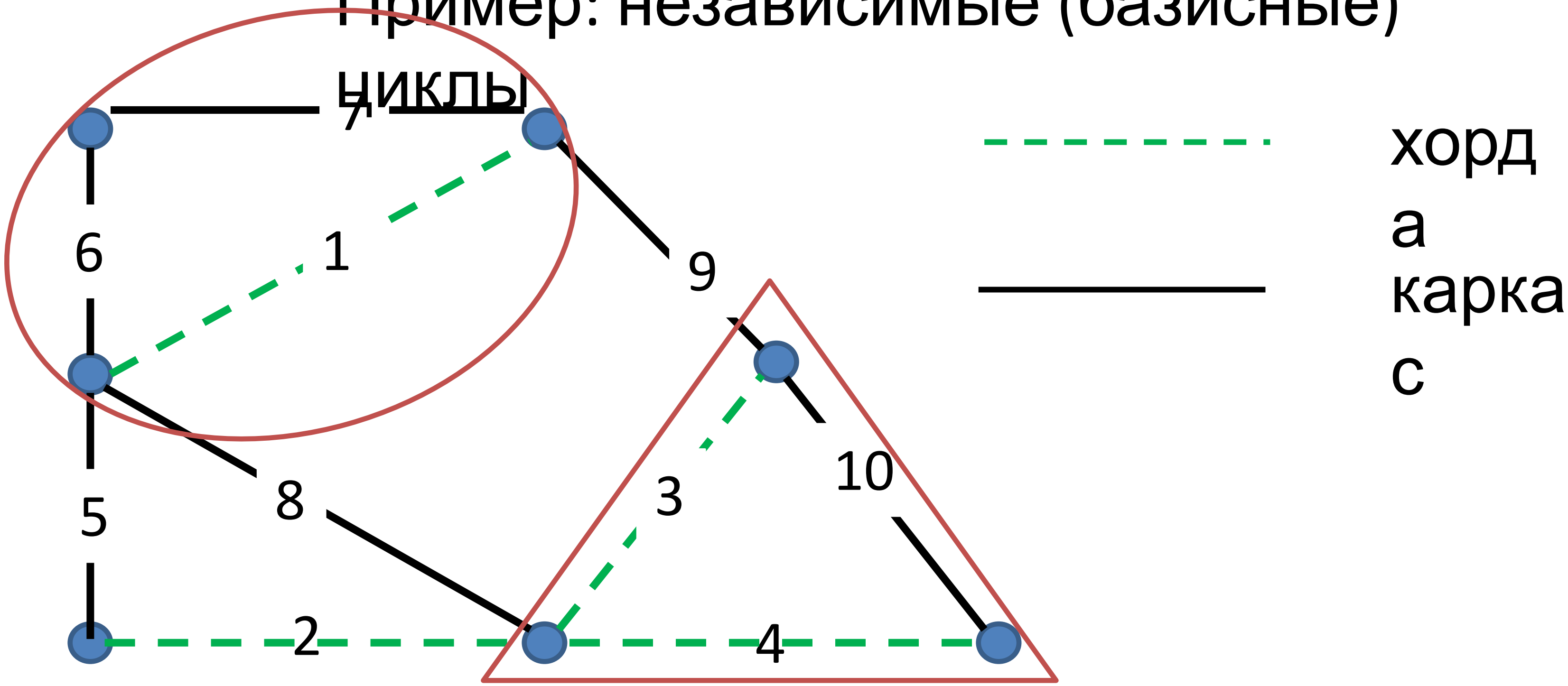
Пример: сумма суграфов (1,1,0,1) ⊕(1,0,1,1)=(0,1,1,0)



В этом разделе цикл определяется как суграф, все вершины ко-
торого имеют четные валентности. Сумма двух циклов есть цикл
(не обязательно цикл в обычном понимании: это могут быть два
цикла без общих вершин). Множество циклов образует линейное
пространство.

Пусть теперь T -- некоторый каркас графа G. Каждой хорде этого
каркаса поставим в соответствие тот единственный цикл, 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
независимых (базисных) циклов.

Пример: независимые (базисные) циклы



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

Цикломатическая матрица

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