27 Понятие степени вершины для ориент. и неор. графа.

def $G(V,E)$ $v \in V$

Степень вершины $\deg(v)$ — # ребер, инцидентных ей.

NB При подсчете степени ребро-петля учитывается 2 раза.

def

Ребро (дуга), инцидентное вершине — если эта вершина является началом или концом ребра.

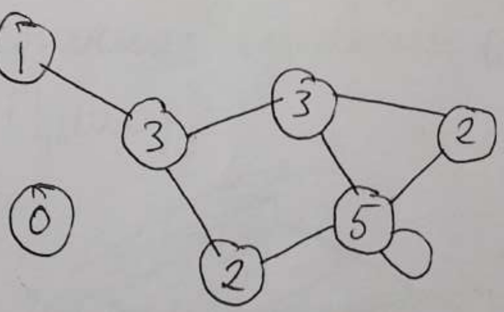
Примеры:

Для ориент. графа различают:

$\delta^+(v)$ — # входящих ребер.

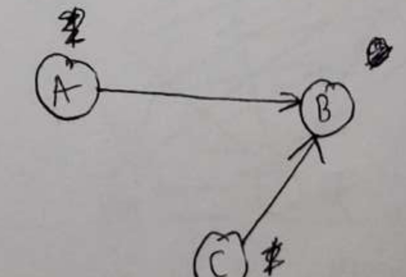
$\delta^-(v)$ — # исходящих ребер.

1)



№ вершины $= \deg(v)$

2)



$\delta^+(A)=0$ $\delta^+(B)=2$

$\delta^-(A)=1$ $\delta^-(B)=0$

$\delta^+(C)=0$

$\delta^-(C)=1$

(28) Определение пути в графе. Типы путей.
— открытый, замкнутый
(29) — простой путь, цепь, цикл.

def G(V,E) - ПУТЬ

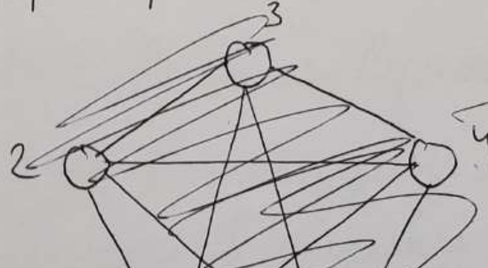
Путь - непрерывная последовательность ребер, согласованная с ориентацией

Типы путей:

① Открытый путь - путь, в котором начальная и конечная вершины не совпадают.

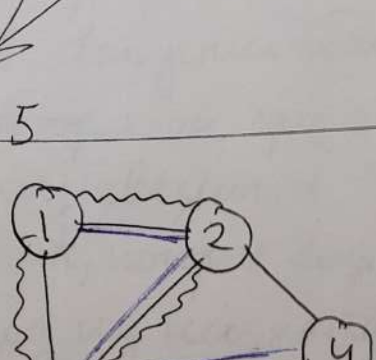
Замкнутый путь - путь, в котором начальная и конечная вершины совпадают.

Пример:



~~открытый путь: 2412341~~

Пример:



123453 - открытый путь.

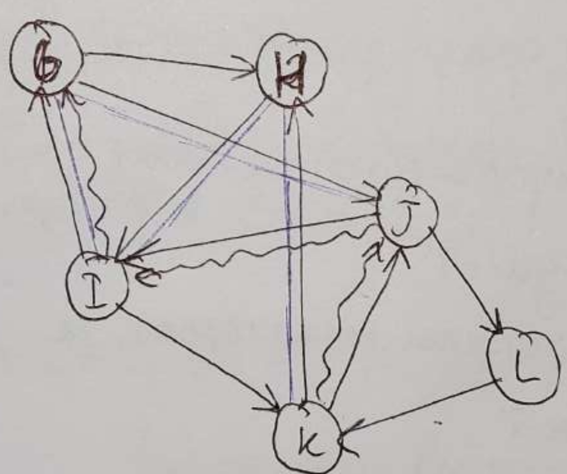
1234531 - закрытый путь

② Простой путь — путь, в котором ни одно ребро не повторяется.

Цепь — простой открытый путь, в котором не повторяются вершины.

Цикл — замкнутый путь, в котором не повторяются вершины.

Пример:



Простой путь: KHIGJ

Цепь: KJIG

Цикл: KJLK.

(30) Связность для неориент. графа — определение. Алгоритм выделения компонент связности в неориентированном графе.

<u>def</u> $G^{(V,E)}$ — неор.

Связанный граф — граф, в котором есть путь между двумя любыми вершинами.

G — связный $\Longleftrightarrow \forall u, v \in V$: $\exists$ путь.

def $G(V,E)$ — неор. граф

Компонента связности — $\forall$ max по включению связн. подграф $\Longleftrightarrow G_1(V_1, E_1)$: $\begin{cases} V_1 \subseteq V \\ E_1 \subseteq E \end{cases}$, G_1 — связный.

$\forall v \in V_1, \forall u \in V \setminus V_1$ нет пути из u в v.

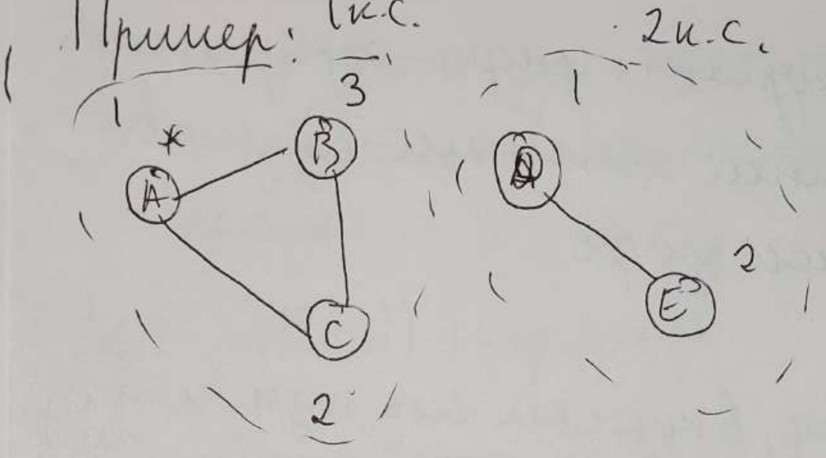
§ Алгоритм выделения компонент связности в неор. графе.

<u>start</u>: выберем вершину $v \in V$

<u>step</u>: воспользуемся либо обходом в глубину, либо обходом в ширину.

Серия обходов: запустим обход из первой вершины и все вершины, которые он при этом обошел — образуют первую компоненту связности. Затем найдем первую из оставшихся вершин, которые еще не были посещены, и запустим обход из неё, найдя тем самым вторую компоненту связности. И так далее, пока все вершины графа не станут окрашенными.

Пример:



1 итерация поиска в глубину: AC

2 итер: AB ~~...~~ } 1 к.с.

3 итер. DE. } 2 к.с.

=> две к.с.

31 К-связность для неориентированного графа — определение. Алгоритм выделения мостов в неор. графе.

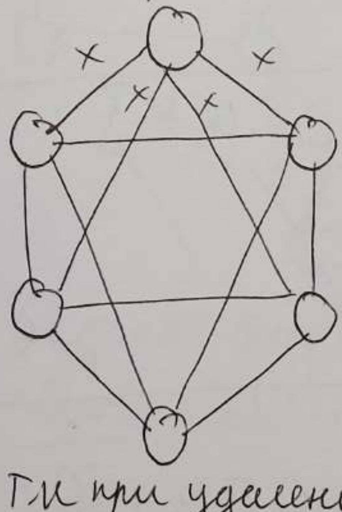
def $G(V, E)$ - неор. граф.

[G - связный $\Longleftrightarrow \forall u, v \in V$ $\exists$ путь.

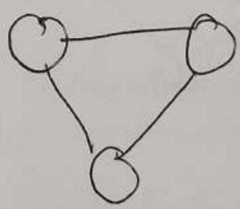
def $G(V, E)$ - неор. граф

[G - к-связный $\Longleftrightarrow \forall e_{i1}, \dots, e_{ik-1}$
$G(V, E \setminus \{e_{i1}, \dots, e_{ik-1}\})$ - связан

Примеры:



4-связный граф. ~~при удалении~~



2-связный граф.
т.к при удалении 1-го ребра, граф остается связным.

т.к при удалении 3-х ребер, - граф ~~д~~ связный.
однако, если удалить 4, то:
граф перестает им быть.

Алгоритм выделения мостов в неор. графе.

$G(V,E)$ — неор. граф.

① Серия поисков в глубину.

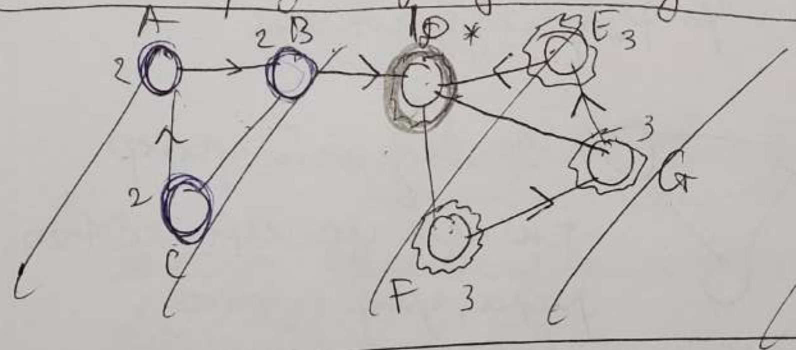
а) I посещение — вершина в список L.

б) Проходя по ребру $e \in E$, на нем — ориентация, противоположная нашему движению.

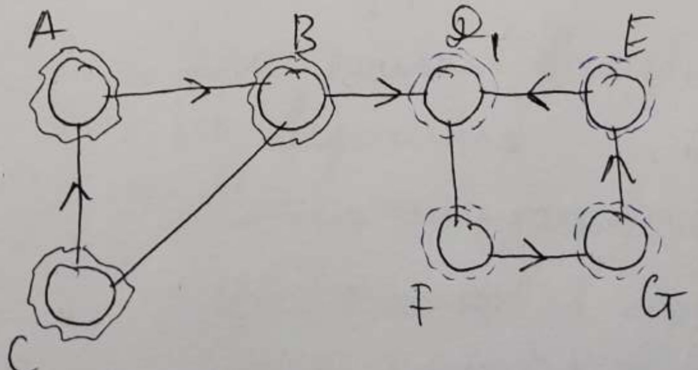
② Вторая серия поисков в глубину.

а) сог. с ориент. ребер.

б) порядок запусков — из списка L.



$\{D, B, A, C, E, G, F\}$



$\{D, B, A, C, E, G, F\}$

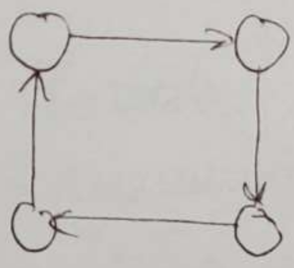
$\Rightarrow$ BD — мост.

32) Связность в ориентированном графе: слабая и сильная связность. Примеры. Алгоритм Косарайю и Шарира.

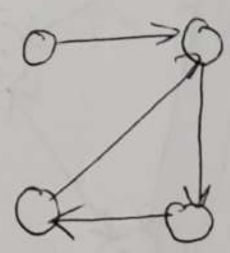
def $G(V,E)$ - ор. граф.

G - слабосвязный $\Leftrightarrow$ связный без учета ориентации ребер.
G - сильносвязный $\Leftrightarrow$ $\forall u,v \in V$ $\exists$ ориент. путь.

Примеры:



- сильно связный граф.



- слабо связный граф.

Алгоритм Косарайю

Задача: выделение компонент сильной связности;

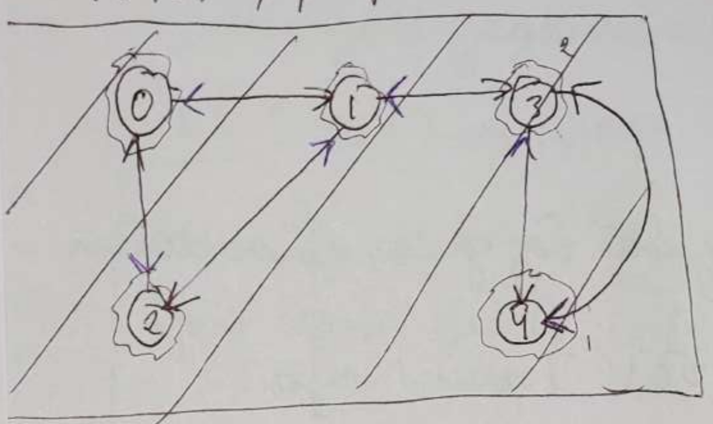
start $v \in V$ - выбираем вершину v.

step 1) Серия поисков в глубину, результат - нумерация v на вершинах.
2) Меняем ориентацию всех ребер (транспонир. графа)
3) Серия поисков в глубину:
а) согласована с новой ориент.
б) нач. с вершины с max № из найденных.

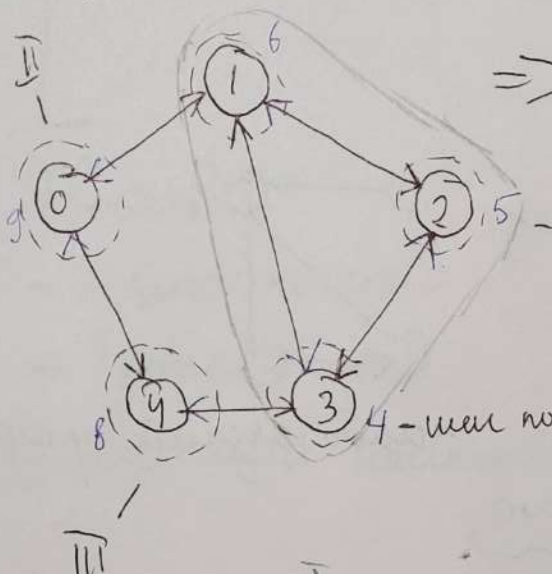
Итог: компоненты сильной связности.

Пример:

G(V,E) - орграф.



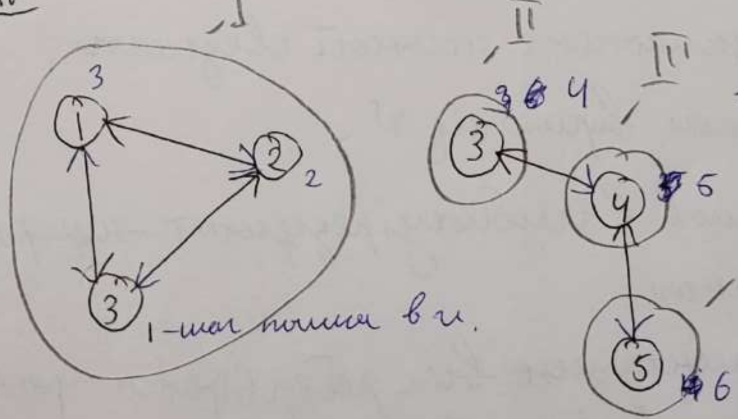
1



=> по завершению алгоритма имеем 3 к.с.с.

4 - шаг поиска в глубину.

2



=> 4 к.с.с.

1 - шаг поиска в г.

(33) Граф Герца - определение, свойства. Приведите пример графа, граф герца для которого будет обладать заданным набором характеристик.

def $G(V,E)$ - ор. граф.

$G_1, \dots, G_k$ - к.с.с. (все).

Граф герца для G: $\widetilde{G}(G) = (V_G, E_G)$, $V_G = \{G_1, \dots, G_k\}$

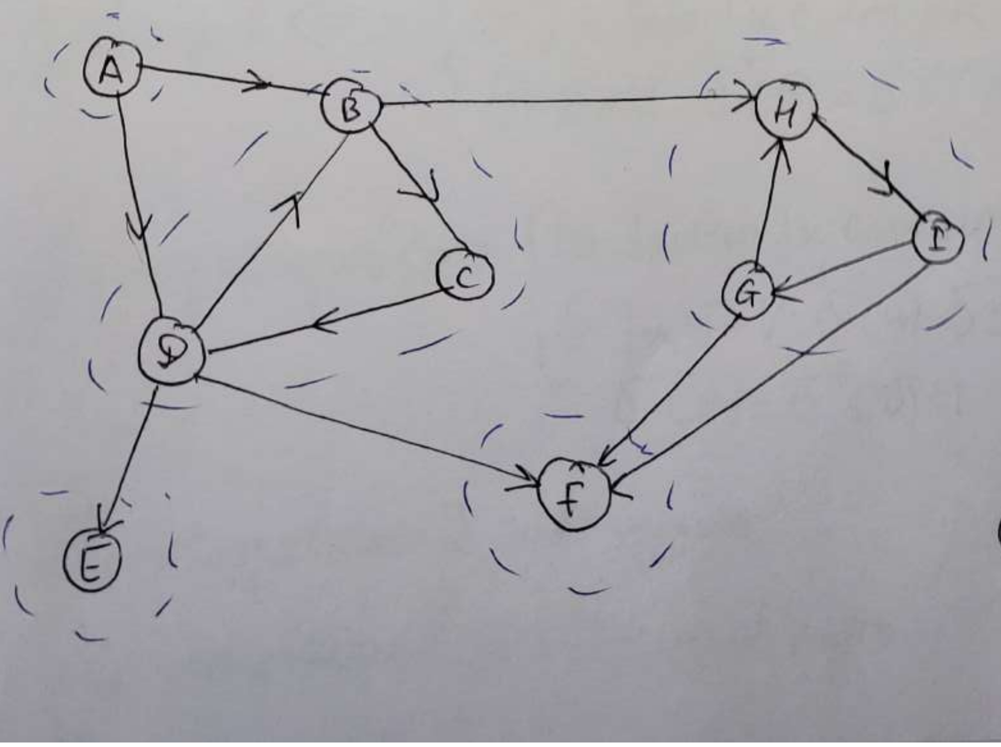
$E_G = \{\tilde{e} \dots \dots\}$

$\tilde{e} \Leftrightarrow \exists u \in G_i, v \in G_j$, т.ч. $\exists e = \overline{uv}$.

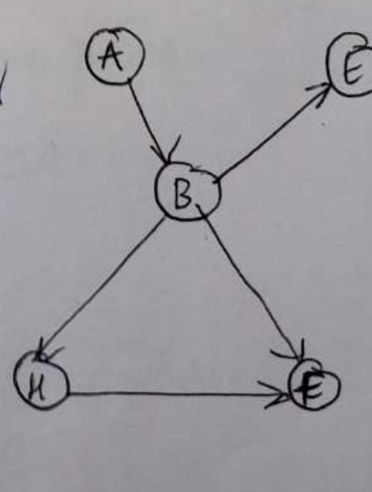
Свойства:

- ацикличный
- ? рёбра-мосты
- вершины-компоненты сильной связанности.

Пример: G



Граф Герца для G:



34 Эйлеров цикл — определение. Критерий эйлеровости для орграфа и для неорграфа.

def $G(V,E)$-неор.

Эйлеров цикл в графе $G \Leftrightarrow$ замкн. путь, прох. по всем $e \in E$ ровно по 1 разу.

Критерий эйлеровости для орграфа и для неографа.

$G(V,E)$-неор. граф.

① эйлеров $\Leftrightarrow$ $\begin{cases} \text{а) } \exists \text{ ровно 1 к.с., содерж. ребра} \\ \text{б) } \forall v \subseteq V: \deg(v) \vdots 2 \end{cases}$

② полуэйлеров $\Leftrightarrow$ $\begin{cases} \text{а) } \exists \text{ ровно 1 к.с. содерж ребра.} \\ \text{б) в графе ровно две вершины нечетной степени.} \end{cases}$

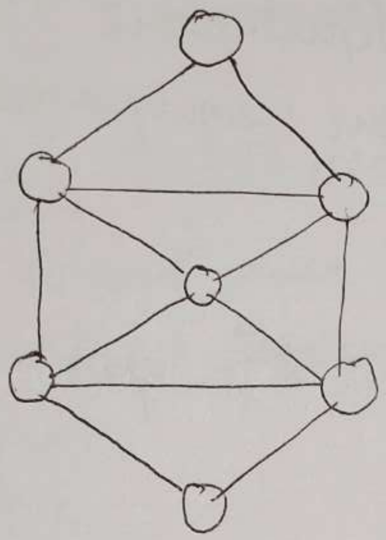
$G(V,E)$-ор. граф.

① эйлеров $\Leftrightarrow$ $\begin{cases} \text{а) } \exists \text{ ровно 1 к.с., содерж. ребра.} \\ \text{б) } \forall v \in V \quad \delta^+(v) = \delta^-(v) \end{cases}$

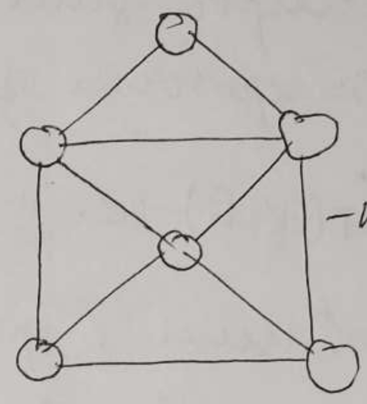
② полуэйлеров $\Leftrightarrow$ $\begin{cases} \text{а) } \exists \text{ ровно 1 к.с., содерж ребра.} \\ \text{б) } \exists u, v \in V: \delta^+(u) = \delta^-(u) + 1 \\ \qquad \delta^-(v) = \delta^+(v) + 1 \end{cases}$

G-эйлеров $\Leftrightarrow \exists$ эйл. цикл.

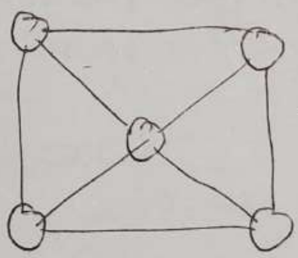
G-полуэйлеров $\Leftrightarrow \exists$ эйлеров путь.



— эйлеров.



— полуэйлеров



— не эйлеров граф

35) Эйлеров граф – определение. Алгоритм Флери.

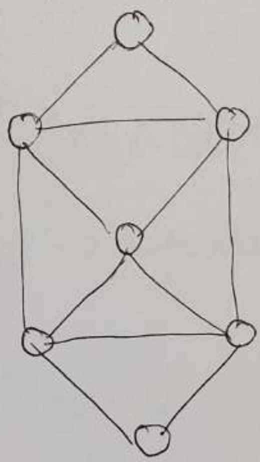
def G(V,E)-неор.

Эйлеров цикл в графе G <=> замкн. путь, прох. по всем ~~вершин~~ $e \in E$ ровно по 1 разу.

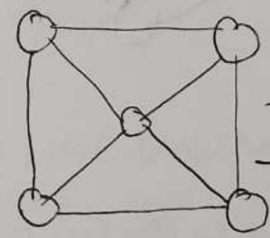
def

G – эйлеров <=> $\exists$ эйл. цикл.
G – полуэйлеров <=> $\exists$ эйл. путь.

Пример:



– эйлеров.



– не эйлеров.

Алгоритм Флери.

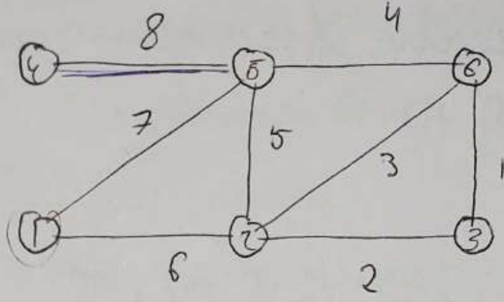
NB Есть дешевый способ проверить, явл. ли $e \in E$ мостом?

start: в качестве начальной вершины – любая, с нечёт. степенью.

step: выстраиваем последов ребер, проходим по мосту только тогда, когда у нас нет возможности пройти не по мосту.

Пример:

①

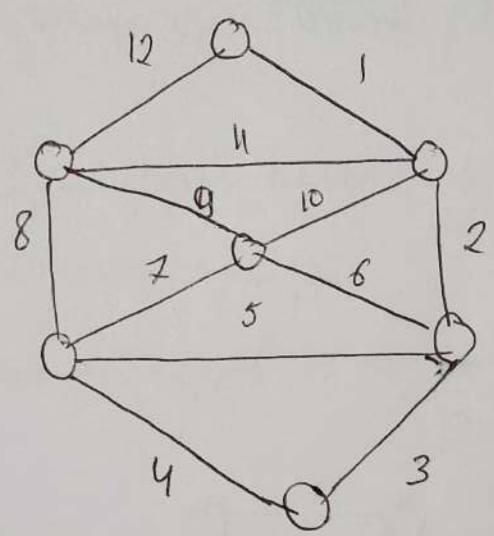


Res: 1 2 3 4 5 6 7 8

45-мост. получим.

Эйлеровый путь.

②



1 2 3 4 5 6 7 8 9 10 11 12 1 -

Эйлеров цикл.

36 Эйлеров путь - определение. Критерий полуэйлеровости для орграфа и для неорграфа.

def

Эйлеров путь в гр. G <=> незамкн. простой путь, проходящий по всем ребрам.

Thm Критерий полуэйлеровости:

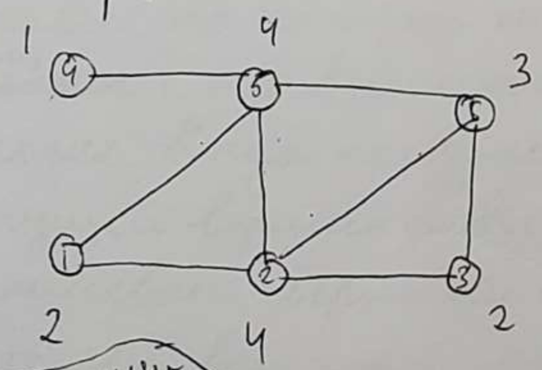
G(V,E) - неор. граф.

G - полуэйлеров <=> { а) ∃ ровно 1 к.с. содерж. ребра.
б) в графе ровно две вершины нечетной степени.

G(V,E) - ор. граф

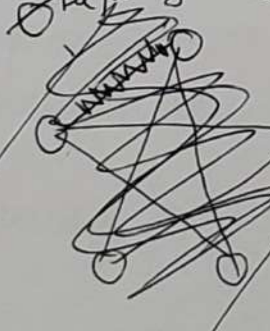
G - полуэйлеров <=> { а) ∃ ровно 1 к.с. содерж. ребра
б) $\exists u, v \in V: \delta^+(u) = \delta^-(u) + 1$
$\delta^-(v) = \delta^+(v) + 1$

Пример:

① 

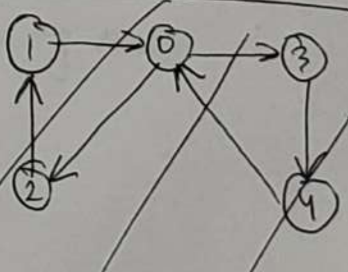
а) 1 к.с.

б) в графе всего две вершины нечетной степени: 4;6 => граф полуэйлеров.

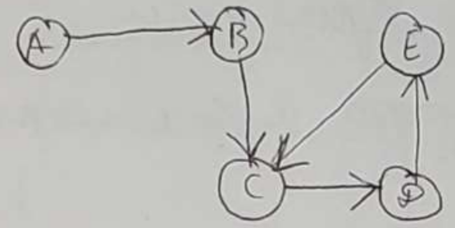
② 

а) 1 к.с.

б)



(2)



Эйлеров путь:
ABCED.

а) 1 к.-с. содерж. ребра.

б) $\exists C: \delta^+(C)=2=\delta^-(C)+1$

$\exists A: \delta^-(A)=1=\delta^+(A)+1$

(37) Полуэйлеров граф — определение. Алгоритм на списках инцидентности.

def

G — полуэйлеров $\Longleftrightarrow$ $\exists$ эйлеров путь

def

Эйлеров путь в гр. G $\Longleftrightarrow$ незамкн. простой путь, проходящий по всем ребрам.

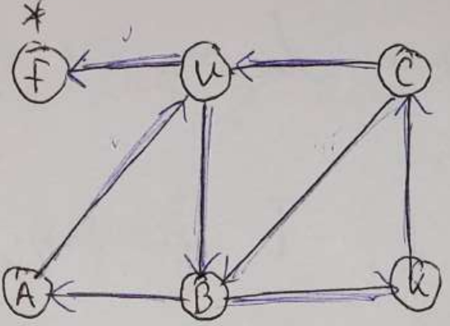
Алгоритм со списком инцидентных вершин.

start: выбираем нач. вершину. Если граф эйлеров — любую. Если граф полуэйлеров — любую вершину нечет. степени.

step: начиная со стартовой вершины строим путь, добавляя на каждом шаге еще не пройденное ребро, ~~смежное~~ инц. с тек. вершиной. Вершины накапливаются в стеке. Когда наступает такой момент, что для текущей вершины все инц. ребра уже пройдены, записываем вершины из стека в ответ, пока не встретим вершину, которой инцидентны еще не пройденные ребра. Далее продолжаем обход по непосещенным ребрам.

Пример:

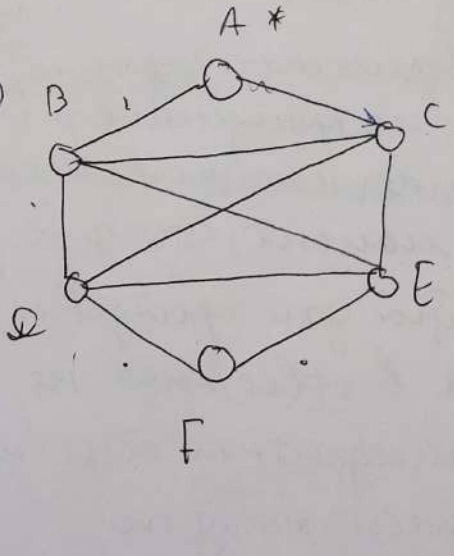
①



A: B,U
B: A,C,K,U
C: B,K,U
F: U
K: B,C
U: A,B,C,F

temp	result
F	C
U	U
A	B
B	K
C	C
K	B
B	A
U	U
C	F

②



A : BC
B: AC
D: BCEF
E: DBCF
F: DE

temp	result
A	A
B	C
D	E
F	D
E	C
B	B
C	E
D	F
E	D
C	B
A	A

(38) Граф де Брюина – определение, примеры.

Пусть алфавит $T = \{\alpha_1, \ldots, \alpha_n\}$

T^k – все слова длины k в алф. T.

$|T^k| = n^k$

Граф де Брюина для слов длины k в алф. T:

$B(T,k) = (V,E)$, $V = T^k$, $u, v \in V \Rightarrow \exists\, e = \overline{uv}$

$u = u_1 \boxed{u_2 \ldots u_k}$

$v = \boxed{u_2 \ldots u_k} v$

$u_1 \boxed{u_2 \ldots u_k} v_{k+1}$

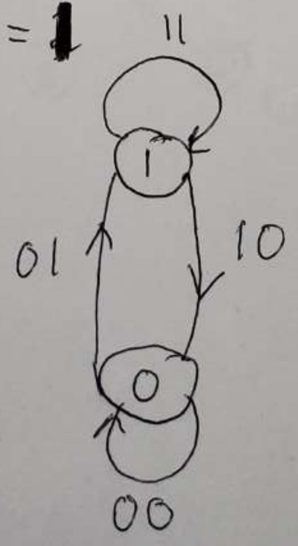
Свойства:

1) Могут быть петли.
2) Ребра могут быть напр. и двунапр.
3) Эйлеров.
4) Биекция между парами вершин и ребер: $|T^k| = n^k$, $|E| = n^{k+1}$

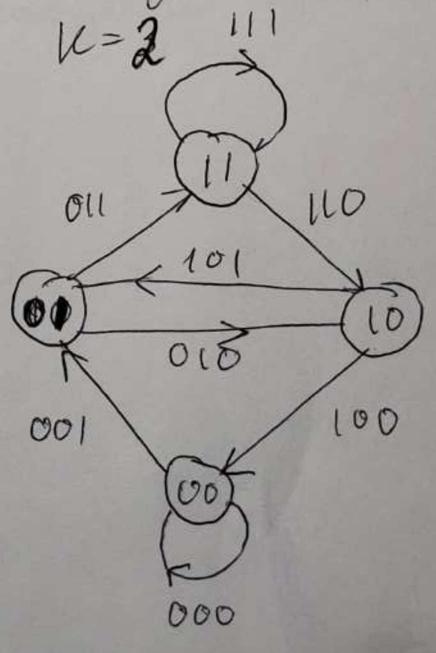
Примеры:

Граф для двоичного алфавита со словами длины

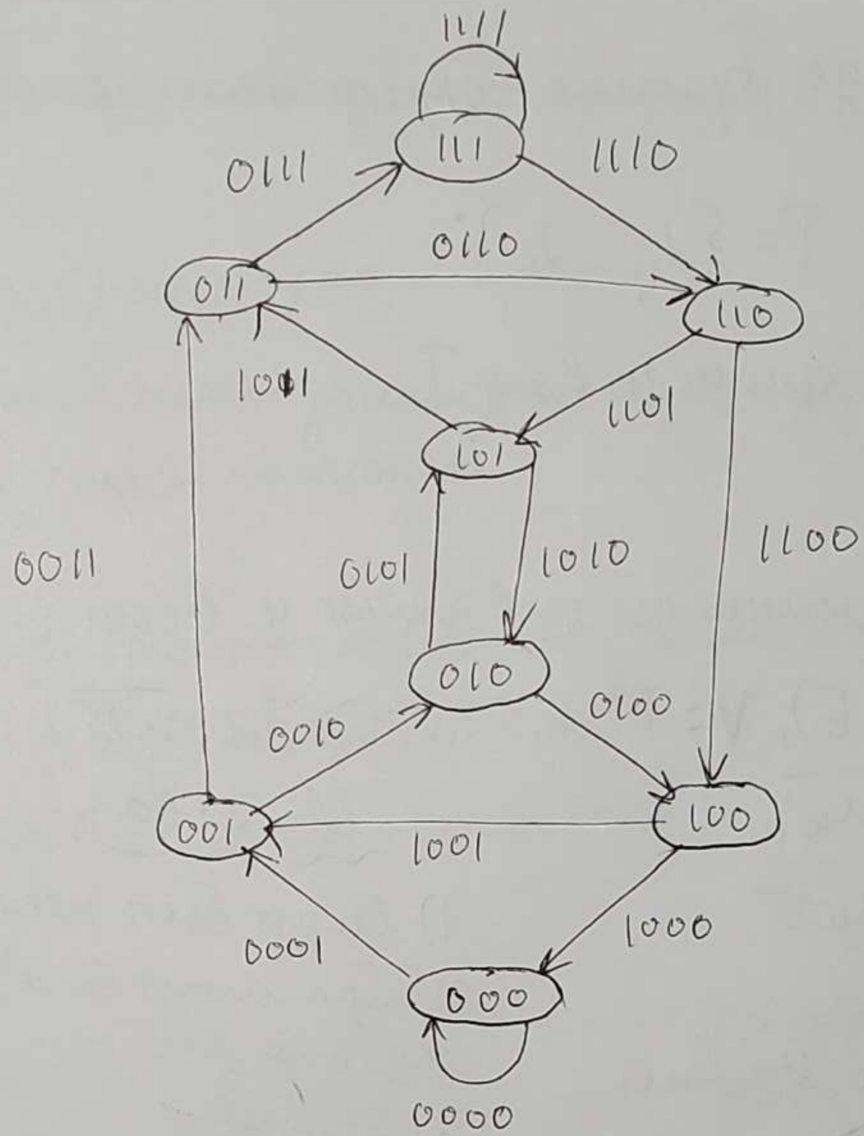
$k = 1$



$k = 2$



$K = 3$



1111
111
0111
1110
0110
011
110
1001
1101
101
0011
0101
1010
1100
010
0010
0100
001
1001
100
0001
1000
000
0000

39 Гамильтонов цикл — определение. Теорема Оре.

def $G(V, E)$ — неор. граф.

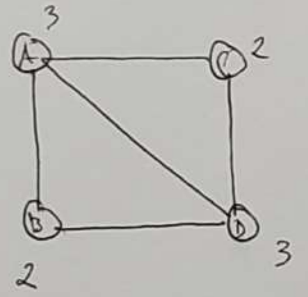
Гамильтонов цикл — замкнутый путь, содерж. $\forall v \in V$ ровно 1 раз, кроме начальных.

Теорема Оре: Достаточное условие гамильтоновости

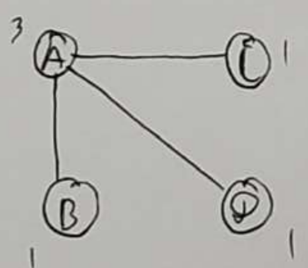
$\forall u, v \in V$, т.ч. u, v — не смежные

$\deg u + \deg v \geq n \Rightarrow G$ — гамильтонов.

Примеры:



$\deg C + \deg B \geq n$; $4 \geq 4 \Rightarrow$ граф гамильтонов + гамильтонов цикл ABDC.



$\deg B + \deg C \geq n$; $1+1 \ngeq 4 \Rightarrow$ граф не гамильтонов, + не существует гам. цикла.