Комбинаторика и теория графов

Первая часть экзамена 22. 12. 2021

#1 Определение бинарного отношения. Свойства бинарных отношений: рефлексивность - определение. Приведите пример и контрпример.

def Б.о

Пусть задано мн-во M. Рассмотрим декартово произведение этого множества на себя $M \times M$. Подмножество R мн-ва $M \times M$ называют бинарным отношением R на мн-ве M ($M \neq 0$ - мн-во; $f: M^2 \to \mathbb{B}$; f - бин. отнош. на M)

def Рефлексивность

Б.о f(R) на мн-ве M называется рефлексивным, если каждый элемент этого мн-ва находится в отношении с самим собой: $\forall m \in M \ \ f(m,m) = 1$ ($mRm = 1$)

Пример: ① Отношение сравнимости рефлексивно (при любом натуральном m и на любом мн-ве целых чисел)

② Отношение делимости рефлексивно (на любом мн-ве целых чисел, не содержащих нуля)

Контрпример: ① Отношение строгого неравенства на мн-ве вещественных чисел не рефлексивно

#2 Определение б.о. Св. б.о: арефлексивность - определение. Пр. и кпр

def Арефлексивность

Б.о f(R) на мн-ве M называют арефлексивным, если ни один элемент этого мн-ва не находится в отношении с самим собой: $\forall m \in M \ \ f(m,m) = 0$ ($mRm = 0$)

Пример: ① Отношение строгого неравенства на мн-ве вещ. чисел арефлексивно
② Отношение взаимной простоты арефлексивно на любом мн-ве целых чисел, не содержащем 1 и -1

Контрпример: ① Отношение взаимной простоты рефлексивно на множествах {1}, {-1}, {-1; 1}

#3 Определение б.о. Свойства б.о: симметричность - опр. Пр. и кпр.

def Симметричность

Б.о. (R) f на мн-ве M называют симметричным, если вместе с каждой парой (a;b) в отношение входит и симметричная пара (b;a): $\forall a, b \in M \quad f(a,b) = f(b;a)$ $(aRb = bRa)$

Пример: ① Отношение сравнимости симметрично при любом натуральном m и на любом мн-ве целых чисел
② Отношение взаимной простоты симметрично на любом мн-ве целых чисел.

Контрпример: ① Отношение строгого неравенства на мн-ве вещественных чисел не симметрично.

#4 Опр. б.о. Свойства б.о: антисимметричность - опр. Пр. и кпр

def Антисимметричность

Б.о. f(R) на мн-ве M называют антисимметричным, если никакая пара, состоящая из разных элементов не входит в отношение вместе с симметричной ей: $\forall a, b \in M \begin{cases} f(a,b) = 1 \\ f(b,a) = 1 \end{cases} \Rightarrow a = b$

Пример: ① Отношение нестрогого неравенства на множестве вещественных чисел антисимметрично.
② Отношение делимости является антисимметричным на любом мн-ве целых чисел

#5 Опр. б.о. Свойства б.о: асимметричность - опр. Пр. и кпр.

def Асимметричность

Б.о $f(R)$ на мн-ве M называют асимметричным, если ни одна пара не входит в отношение вместе с симметричной ей: $\forall a, b \in M \quad f(a,b)=1 \Rightarrow f(b,a)=0$

Пример: ① Отношение строгого неравенства на мн-ве вещественных чисел асимметрично

Контрпример: ① Отношение делимости не является асимметричным ни на каком мн-ве целых чисел, не содержащих нуля, вследствие рефлексивности

#6 Опр. б.о. Свойства б.о: транзитивность - опр. Пр. и кпр.

def Транзитивность

Б.о. $f(R)$ на мн-ве M называют транзитивным, если вместе с парами $(a;b)$ и (b,c) в отношение входит и пара $(a;c)$:
$$\forall a,b,c \in M \quad \begin{cases} f(a,b)=1 \\ f(b,c)=1 \end{cases} \Rightarrow f(a,c)=1$$

Пример: ① Отношение сравнимости транзитивно при любом натуральном m и на любом мн-ве целых чисел.
② Отношение делимости транзитивно на любом множестве целых чисел

Контрпример: ① Отношение взаимной простоты не является транзитивным на любом мн-ве целых чисел. Например, 2 взаимно просто с 3, 3 взаимно просто с 4, но 2 и 4 не взаимно просты.

#7 Опр. б.о. Способы задания б.о: аналитическое. Примеры

Дано множество и "формула" бинарного отношения

Пример: $M \in \mathbb{Z}$; $f(m_1, m_2) = \begin{cases} 1, & m_1 \equiv_3 m_2 \\ 0, & \text{иначе} \end{cases}$

#8 Опр. б.о. Способы задания б.о: задание матрицей. Примеры

Определим матрицу A размера $|M| \times |M|$, где $|M|$ - кол-во элементов мн-ва M (дискретного и конечного). Тогда $a_{ij} = 1$, если элемент с номером i находится в отношении с элементом с номером j: $f(m_i, m_j) = 1$; а $a_{ij} = 0$ - иначе. $M = \{m_1, \dots, m_n\}$

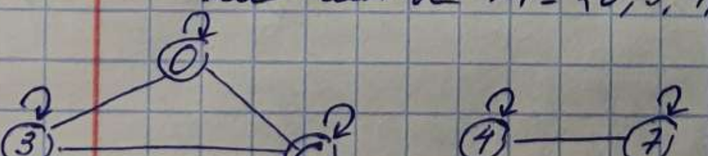
Пример: Матрица смежности для отношения делимости на мн-ве $M = \{1, 2, 3, 4, 5\}$

	1	2	3	4	5
1	1	0	0	0	0
2	1	1	0	0	0
3	1	0	1	0	0
4	1	1	0	1	0
5	1	0	0	0	1

#9 Опр. б.о. Способы задания б.о: задание графом. Примеры

Элементы мн-ва изображаются точками плоскости и образуют мн-во вершин графа. Отношение представляют дугами (ребрами) графа: если (a, b) входит в отношение, то у вершины a проводится ориентированная дуга в b

Пример: Граф для отношения сравнимости по модулю 3 на мн-ве $M = \{0, 3, 4, 6, 7\}$



#10 Определение б.о. Типы бинарных отношений: отношение толерантности - определение. Свойства матрицы и графа такого б.о.

det **Отношение толерантности** $M \neq \varnothing,\ f: M^2 \to \mathbb{B}$

Отношение толерантности на мн-ве M называется б.о, удовлетворяющее свойствам рефлексивности и симметричности, но не обязат. обладающее транзитивностью

На содержательном уровне толерантность означает следующее:
- любой объект неразличим сам с собой - рефлексивность:
  для матрицы - единицы на главной диагонали;
  для графа - петля в каждой вершине
- сходство двух объектов не зависит от порядка сравнения - симметр:
  для матрицы - симметричность относит. главной диагонали
  для графа - все стрелки парные

#11 Опр. б.о. Типы б.о: отношение эквивалентности - опр. Свойства

det **Отношение эквивалентности** $M \neq \varnothing,\ f: M^2 \to \mathbb{B}$

Бин. отн., обладающее свойствами рефлексивности, симметричности и транзитивности, называют отношением эквивалентности

M, f - экв ; $M = M_1 \sqcup \ldots \sqcup M_k$, M_i - классы эквивалентности

$\forall m, n \in M_i : f(m,n) = 1$

$\forall m \in M : \exists i\ M_i;$

$\forall m \in M_i,\ n \in M_j : f(m,n) = 0,\ i \neq j$

$\forall\ 1 \leq i, j \leq k : M_i \cap M_j = \varnothing$ или $M_i = M_j$

Граф транзитивного отношения с каждой парой стрелок, идущих от x к y и y к z, содержит стрелку, идущую от x к z.

#12 Опр. б.о. Типы б.о: отношение предпорядка - опр. Свойства

def **Отношение предпорядка**

Б.о называют отношением предпорядка, если оно рефлексивно и транзитивно.

#13 Опр. б.о. Типы б.о: отношение частичного (нестрогого) порядка. Свойства

def **Отношение частичного порядка**

Отношение предпорядка (рефл. и транзит.), обладающее свойством антисимметричности, называют отношением частичного (нестрогого) порядка.

ex $M = \{1, 2, 3, 4\}$, отношение делимости - отношение част. порядка, т.к ни 3 не делится на 4, ни 4 на 3

#14 Опр. б.о. Типы б.о: отношение линейного порядка. Свойства

def **Отношение линейного порядка**

Отношение порядка, обладающее следующим свойством: $\forall a, b \in M: f(a,b) = 1$ и/или $f(b,a) = 1$, называют отношением линейного порядка.

ex $M = \{1, 2, 4, 8\}$, отношение делимости - отношение лин. порядка

(14) Линейный порядок. $M \neq \varnothing \quad f: M^2 \to \mathbb{B}$
Частичный порядок, где $\forall a, b \in M: f(a,b) = 1$ или/и $f(b,a) = 1$, называется линейным порядком (можно сравнить любые 2 элемента).

Примеры: $\mathbb{N}, \leq$; $\mathbb{Z}, \geq$

Матрица имеет 1 на главной диагонали, а также a_{ij} или/и $a_{ji} = 1$ $\forall i \neq j$.
Граф полный (без учёта ориентации рёбер), с петлями.

(15) Строгий порядок. $M \neq \varnothing \quad f: M^2 \to \mathbb{B}$.

Асимметричное и транзитивное бин. отн. называется отношением строгого порядка. Примеры: $\mathbb{N}, <$; $\mathbb{Z}, >$.

Матрица имеет 0 на главной диагонали, а также a_{ij} или (не и) $a_{ji} = 1$ $\forall i \neq j$.
Граф полный, все рёбра однонаправлены, нет циклов, нет петель.

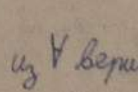
(16) Замыкание относительно свойства. Рефлексивное замыкание.
Замыкание б.о. f относительно свойства $*$ — б.о. $f^*: M^2 \to \mathbb{B}$ т.ч.
1) f^*, f согласованы ($f(a,b) = 1 \Rightarrow f^*(a,b) = 1$)
2) Для f^* выполнено св-во $*$.
3) f^* — min со св-вами 1) и 2).

Рефлексивное замыкание. Примеры: $M = \mathbb{N}$, $f = <$, $f^* = \leq$; $M = \mathbb{Z}$, $f = >$, $f^* = \geq$

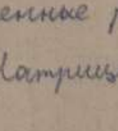
В матрице смежности появляются 1 на гл. диагонали, в графе — петли.

(17) Симметричное замыкание. Примеры: $M = \mathbb{N}$, $f = \leq$, $f^* = $ "$=$" ; $M = \mathbb{Z}$, $f = \geq$, $f^* = $ "$=$"

Матрица становится симметричной ($A^T = A$), рёбра графа становятся двунаправленными.



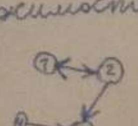
(18) Транзитивное замыкание. Примеры:



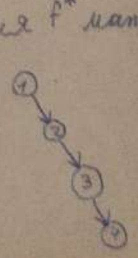
В графе появляются направленные рёбра из $\forall$ вершины a в те вершины, до которых из a есть путь. Матрица становится матрицей достижимости.

(19) Матрица достижимости б.о. $f: M^2 \to \mathbb{B}$ — $P_f = \begin{array}{c} \\ i \end{array}\begin{pmatrix} & j & \\ \cdot & 1 & \end{pmatrix}$ такая, что если существует путь $m_i \to \ldots \to m_j$, то $f(m_i, m_j) = 1$.
Для транзитивного замыкания f^* матрицей является матрица достижимости.

Примеры:



	1	2	3	4
1	1	1	1	0
2	1	1	1	0
3	0	0	0	0
4	0	0	1	0



	1	2	3	4
1	0	1	1	1
2	0	0	1	1
3	0	0	0	1
4	0	0	0	0

(20) Применим ли к данному б.о. алгоритм topsort? Обосновать.

Topsort – дополнение частичного порядка до линейного.

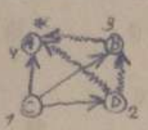
1) Выбор вершины

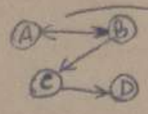
2) Поиск в глубину от выбранной вершины с нумерацией с конца.

3) Провести рёбра от вершин с меньшим номером к вершинам с бо́льшим.

Вершина с бо́льшим номером – min для отношения частичного порядка f.

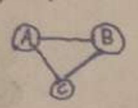
Алгоритм применим только для отношения частичного порядка.

Примеры:  – применим

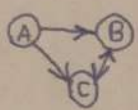
 не антисимметр. ⇒ не частичный порядок ⇒ не применим.

(21) Граф $G(V, E)$ – объект, состоящий из множества вершин E и множества рёбер E, соединяющих эти вершины.

Задание графа бинарным отношением $f: V^2 \to \mathbb{B}$, $f(u,v)=1 \Leftrightarrow$ ребро $uv \in E$. $\forall u, v \in V$

Примеры: 

$f(A,A)=0$ $f(A,C)=1$ $f(C,B)=1$
$f(A,B)=1$ $f(C,A)=1$ $f(B,B)=0$ – неор. граф.
$f(B,A)=1$ $f(B,C)=1$ $f(C,C)=0$

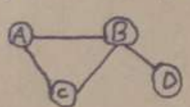


$f(A,B)=1$ $f(B,A)=0$ $f(C,A)=0$
$f(A,C)=1$ $f(B,B)=0$ $f(C,B)=1$ – ор. граф.
$f(A,A)=0$ $f(B,C)=1$ $f(C,C)=0$

(22) Задание графа матрицей смежности. $V=\{v_1, v_2, \dots, v_n\}$.

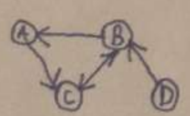
$$\begin{array}{c|c} & 1 \dots j \dots n \\ \begin{matrix}1\\ \vdots\\ i\\ \vdots\\ n\end{matrix} & \begin{pmatrix} & & \\ - & - & 1 \\ & & \end{pmatrix}\end{array} \Leftrightarrow v_i \text{ и } v_j \text{ смежны}$$

$1 \le i, j \le n$

Примеры: 

	A	B	C	D
A	0	1	1	0
B	1	0	1	1
C	1	1	0	0
D	0	1	0	0

– неор. граф.



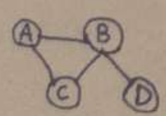
	A	B	C	D
A	0	0	1	0
B	1	0	1	0
C	0	1	0	0
D	0	1	0	0

– ор. граф.

(23) Задание графа матрицей инцидентности. $V=\{v_1, \dots, v_n\}$, $E=\{e_1, \dots, e_m\}$

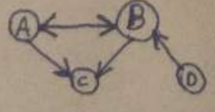
$$\begin{array}{c|c} & 1 \dots j \dots m \\ \begin{matrix}1\\ \vdots\\ i\\ \vdots\\ n\end{matrix} & \begin{pmatrix} & & \\ - & - & 1 \\ & & \end{pmatrix}\end{array} \Leftrightarrow e_j \text{ инцидентно вершине } v_i$$ (–1, если ор. граф, и ребро e_j входит в v_i)

$1 \le i \le n$, $1 \le j \le m$.

Примеры: 

	AB	AC	BC	BD
A	1	1	0	0
B	1	0	1	1
C	0	1	1	0
D	0	0	0	1

– неор. граф.



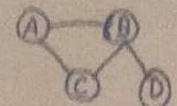
	AB	AC	BA	BC	DB
A	1	1	-1	0	0
B	-1	0	1	1	-1
C	0	-1	0	-1	0
D	0	0	0	0	1

– ор. граф

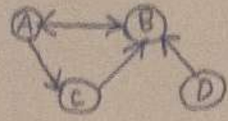
(24) Задание графа списком инцидентности. $V=\{v_1, v_2, \dots, v_n\}$

Списки вида $v_i : v_{j_1}, v_{j_2}, \dots, v_{j_m}$, $\forall k$ v_{j_k} смежны с v_i.

Примеры:



A: B, C
B: A, C, D — неор. граф.
C: A, B
D: B



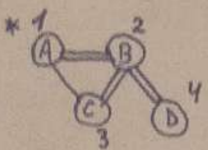
A: B, C
B: A
C: B — ор. граф.
D: B

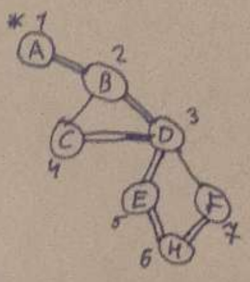
(25) Обход в глубину — алгоритм обхода графа $G(V,E)$.

DFS: start: вершина $v \in V$

Step: помечаем вершину v;
для всех смежных с v неотмеченных вершин $u \in ch(v)$:
запуск DFS(u);

Примеры:



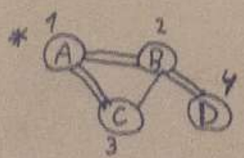


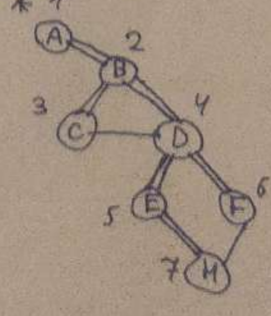
(26) Обход в ширину — алгоритм обхода графа $G(V,E)$

BFS: start: вершина $v \in V$ — помечаем, кладём в очередь

step: ~~помечаем~~ достаём вершину v из очереди;
проход по всем неотмеченным смежным с v вершинам $u \in ch(v)$;
добавление u в очередь, помечаем каждую вершину u.

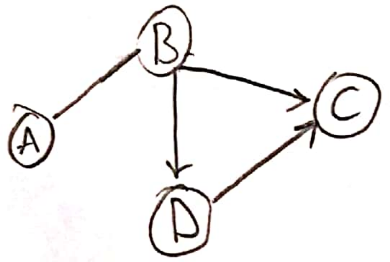
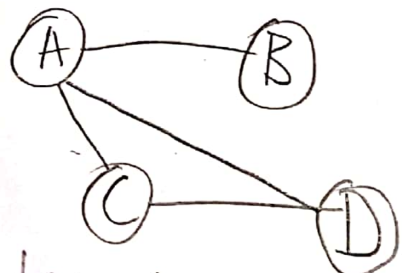
Примеры:





27. Понятие степени вершины для ориентированного и неориентированного графов.

Степень вершины:

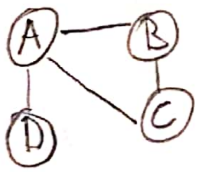
Ориентированный граф	Неориентированный граф
$\delta^+(v)$ — кол-во ребер, входящих в v $\delta^-(v)$ — кол-во ребер, исходящих из v	$\forall v \in V: deg(v) = \delta(v)$ — кол-во ребер, инцидентных v. $deg(v) = 0$ — изолированная вершина $deg(v) = 1$ — висячая вершина (лист)
 1) $\delta^+(A) = 1$ $\delta^-(A) = 1$ 2) $\delta^+(B) = 1$ $\delta^-(B) = 3$	 1) $deg(A) = 3$ 2) $deg(B) = 1$

28. Определение пути в графе. Типы путей: открытый, замкнутый, каждый тип проиллюстрируйте примерами.

Путь — непрерывная последовательность ребер, согласованная с ориентацией:

Открытый путь — начальная и конечная вершины не совпа

Закрытый (замкнутый) путь — начальная и конечная вершины совпадают



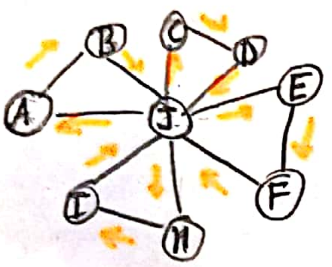
Открытый путь: DABC

Замкнутый путь: DABCAD

29. Определение пути в графе. Типы путей: простой путь, цепь, цикл. Каждый тип проиллюстрируйте примером.

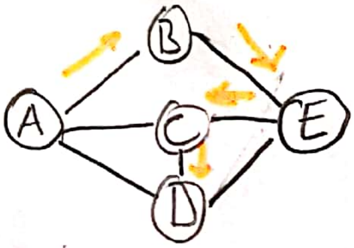
Путь – непрерывная последовательность ребер, согласованная с ориентацией.

Простой путь – путь без повторяющихся ребер.



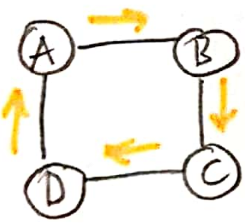
A B J C D J E F J H I J A

Цепь – путь без повторяющихся вершин и ребер.



ABECD

Цикл – замкнутый простой путь, кроме начальной нет повторяющихся вершин.

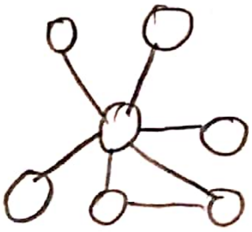


ABCDA

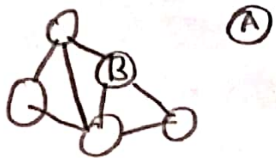
30. Связность для неориентированного графа – определение. Алгоритм выделения компонент связности в неориент. графе.

$G(V, E)$ – неор. граф

G – связный $\iff \forall\, v, u \in V\ \exists$ путь



← связный



← не связный, т.к. $\exists\, A \in V$: A и B $\nexists$ пути

$G(V,E)$ – неор. граф.

<u>Компонента связности</u>: $\forall$ max по включению связный подграф $G_1(V_1,E_1)$: $\begin{cases} V_1 \subseteq V \\ E_1 \subseteq E \end{cases}$, G_1 – связный.

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

Алгоритм выделения к.с. в неориентированном графе.

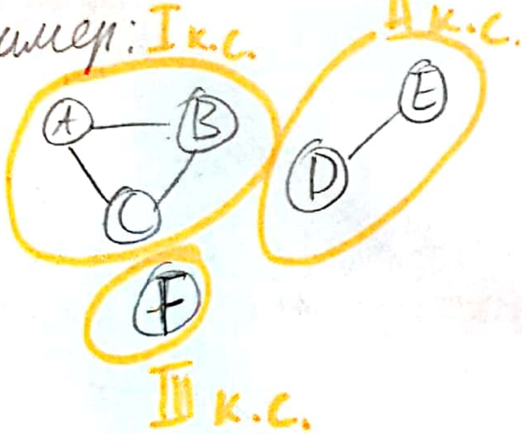
(1) Составляем список вершин L

(2) Для невычеркнутой вершины в списке L запускаем поиск в глубину и кладем полученные вершины в список L_i

(3). Вычеркиваем из списка L вершины списка L_i

(4) Если в списке L остались незачеркнутые вершины $\Rightarrow$ выполняем (2)–(4), иначе получено разбиение на к.с. (списки L_i)

Пример:



0) $L = \{A, B, C, D, E, F\}$

1) $L_1 = \{A, B, C\}$

$L = \{\cancel{A}, \cancel{B}, \cancel{C}, D, E, F\}$

2) $L_2 = \{D, E\}$

$L = \{\cancel{A}, \cancel{B}, \cancel{C}, \cancel{D}, \cancel{E}, F\}$

3) $L_3 = \{F\}$

$L = \{\cancel{A}, \cancel{B}, \cancel{C}, \cancel{D}, \cancel{E}, \cancel{F}\}$

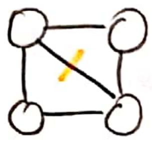
L_1, L_2, L_3 – вершины, содержащиеся в i-ой к.с.

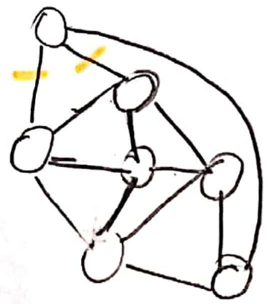
31. k-связность для неориентированного графа – определение. Алгоритм выделения мостов в неориентированном графе.

def
$G(V, E)$ – неор. граф, $k \in \mathbb{N}$
G – k-связный $\Leftrightarrow \forall e_{i_1}, \dots, e_{i_{k-1}}$:
$G(V, E \setminus \{e_{i_1}, \dots, e_{i_{k-1}}\})$ – связный.

(Можно выкинуть $k-1$ ребро, но граф все равно останется связным)

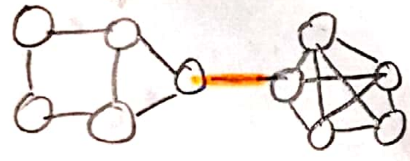
Примеры:

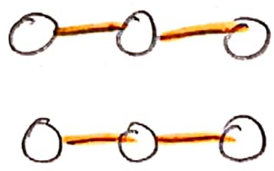

– 2 связный


3 связный

def
$G(V, E)$ – неор. граф.
$e \in E$ – **мост** $\Leftrightarrow$ # комп. связности в $G_1(V, E \setminus e)$ больше, чем в $G(V, E)$.

Примеры:

1) 

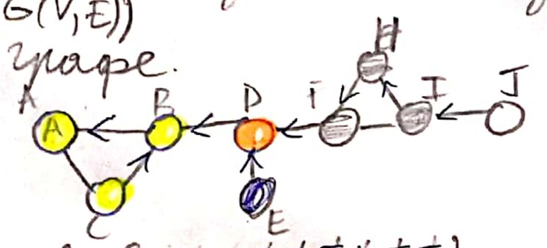
2) 

(Граф не обязательно связный) ($G(V,E)$)

Алгоритм поиска мостов в неор. графе.

① Серия поисков в глубину
а) I посещение – вершина в $\mathcal{L}$
б) Проходя по ребру на нем – ориентация, противоположная движению

② Вторая серия поисков в глубину:
а) согласно ориентации ребер; б) Порядок запусков из списка $\mathcal{L}$



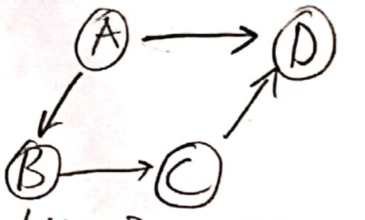
$\mathcal{L} = \{A\ B\ C\ D\ E\ F\ H\ I\ J\}$
Мосты $\{BD, DE, DF, IJ\}$

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

def $G(V,E)$ – ор. граф.

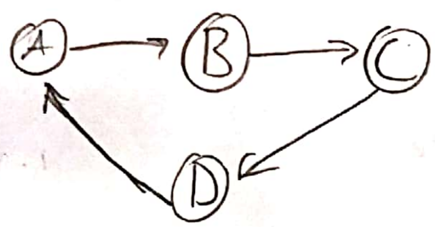
G – слабосвязный $\Leftrightarrow$ связный, без учёта ориентации рёбер

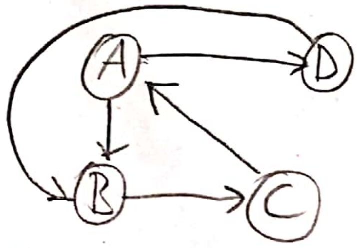
(из C нет пути в A)



(из D нет пути в A)

G – сильносвязный $\Leftrightarrow \forall\, v,u \in V\ \exists$ ор. путь





Алгоритм Косарайю и Шарира
(Выделение компонент сильной связности).

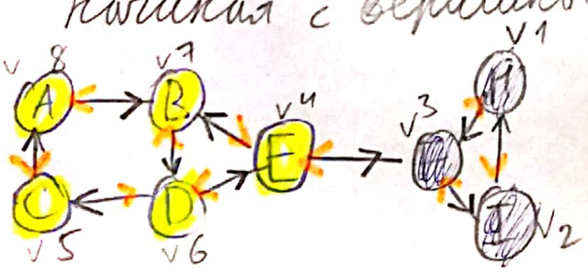
① Серия поисков в глубину. Результат – нумерация на вершинах (у max по глубине вершине, min номер)

② Меняем ориентацию у всех рёбер (транспонирование графа)

③ Серия поисков в глубину:

а) согласно с новой ориентацией

б) начиная с вершины с max N_2 из необработанных



33. Граф Герца – определение, св-ва. Приведите пример графа, граф Герца для которого будет обладать заданным набором характеристик.

<u>def</u> $G(V, E)$ – ор. граф.

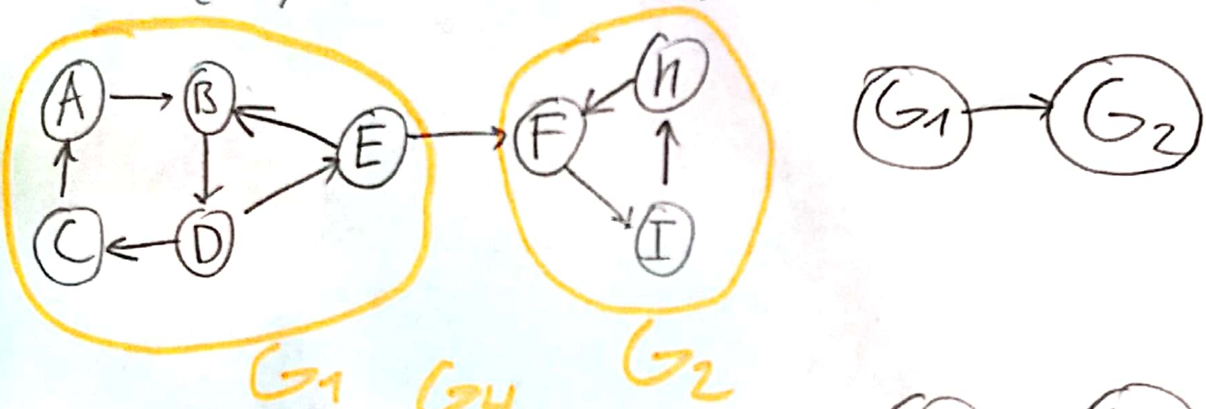
$G_1, \dots, G_k$ – к.с.с.
(все)

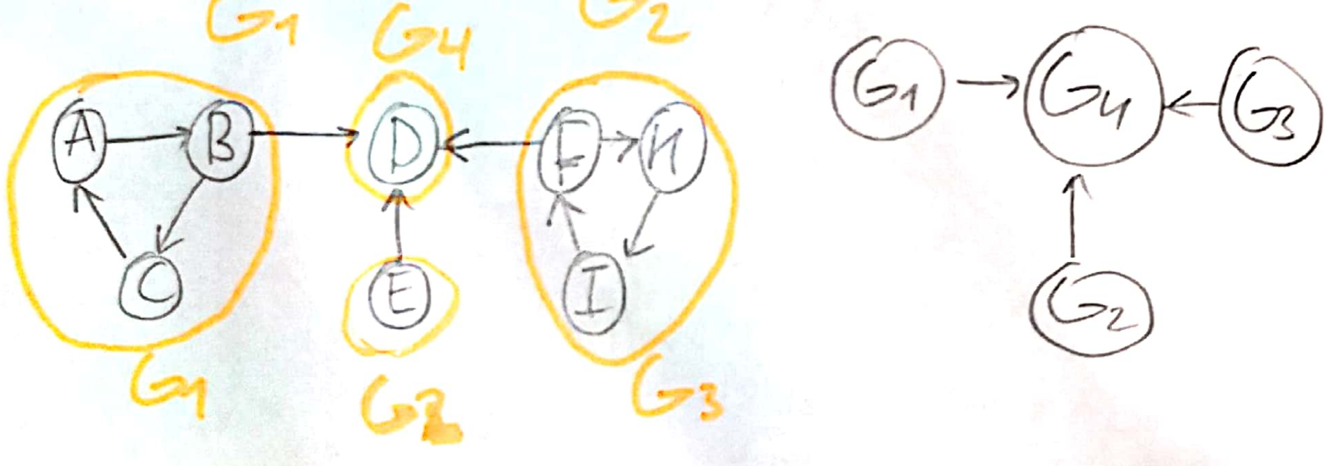
Граф Герца для G (граф концентрации, граф конденсации).

$\tilde{G}(G) = (V_G, E_G)$

$V_G = \{G_1, \dots, G_k\}$

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



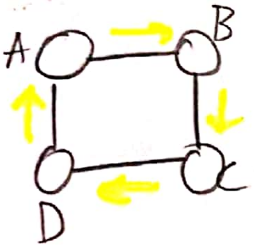


Св-ва графа Герца:

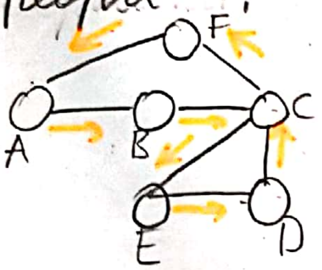
1. не содержит ориентированных циклов
2. Единственный для данного $G(V, E)$.

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

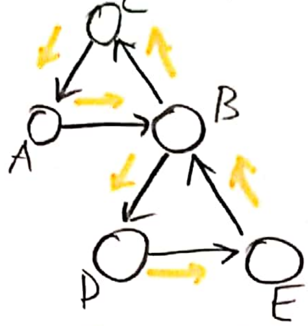
Эйлеров цикл – замкнутый простой путь, содержащий все рёбра



A B C D A



A B C E D C F A



A B D E B C A

Критерий эйлеровости:

1) $G(V,E)$ – неор. граф:

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

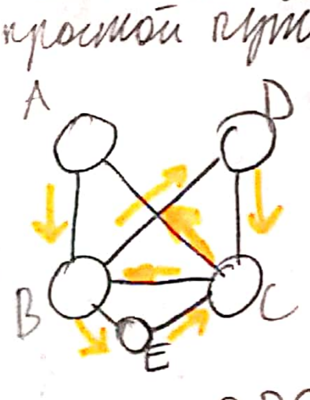
2) $G(V,E)$ – ориент. граф:

$\begin{cases} \text{а) } \exists \text{ ровно 1 к.с.с., содержащая ребра (компонента сильной связности)} \\ \text{б) } \forall v \in V: \delta^+(v) = \delta^-(v) \end{cases}$

---

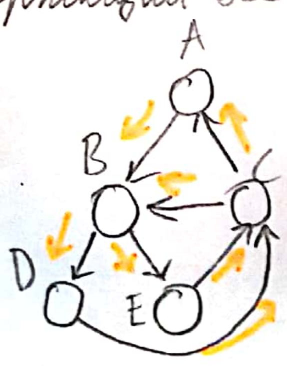
35. Эйлеров граф – определение. Алгоритм Флери.

def G – эйлеров $\Leftrightarrow$ $\exists$ эйлеров цикл (замкнутый простой путь, содержащий все ребра).



A B E C B D C A

G – эйлеров



A B D C B E C A

G – эйлеров

Алгоритм Флери. (построение эйлерова пути (цикла))

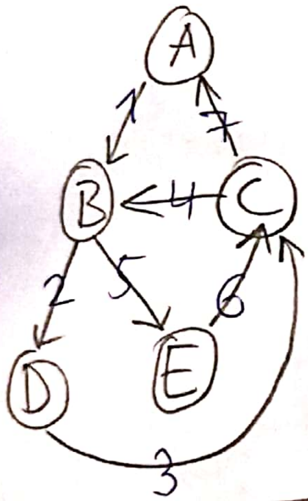
NB Есть дешевый способ проверить является ли $e \in E$ мостом? (алг. Косарайю-Шарира).

start. эйлеров - любая вершина
полуэйлеров неор. - с любой из двух вершин нечетной степени.
Полуэйлеров ор. - с вершины: $\delta^-(v) = \delta^+(v) + 1$

step Проходим по ребрам, не являющимися мостами (в приоритете).

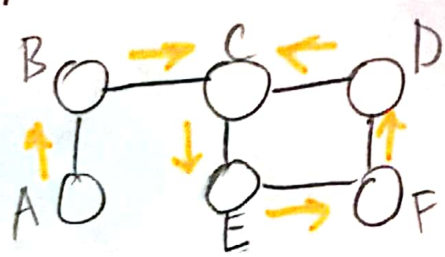
$\Downarrow$

обеспечивает целостность графа (не разваливается) (отваливаются отдельные изолированные вершины

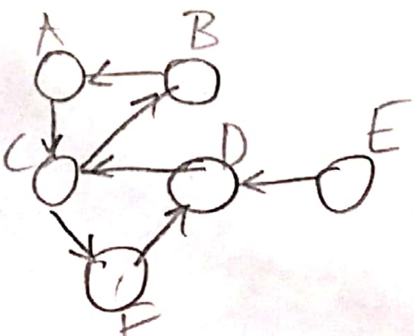


36. Эйлеров путь определение. Критерий полуэйлеровости для орграфа и для неорграфа.

Эйлеров путь - незамкнутый простой путь проходящий по всем ребрам.



A B C E F D C



EDCBACFD

Критерий полуэйлеровости:

1) $G(V, E)$ – неор. граф.

$$\begin{cases} \text{а) } \exists \text{ ровно 1 к.с., содерж. ребра} \\ \text{б) } \exists \text{ ровно 2 вершины неч. степени} \end{cases}$$

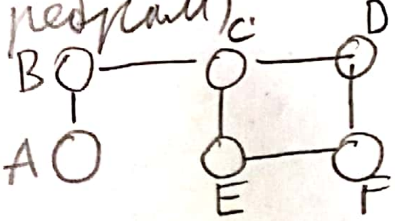
2) $G(V, E)$ – ор. граф:

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

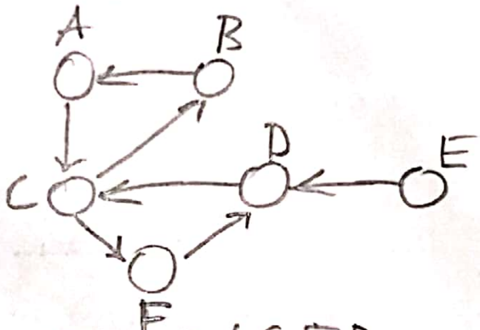
---

37. Полуэйлеров граф – определение. Алгоритм на списках инцидентности.

def G – полуэйлеров $\Leftrightarrow \exists$ эйлеров путь (незамкнутый простой путь, проходящий по всем ребрам)



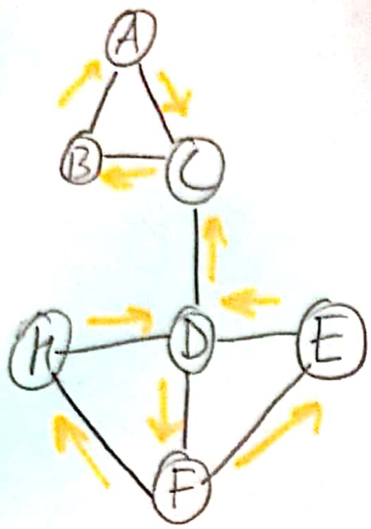
ABCEFDC



EDFBACFD

Алгоритм на списках инцидентности (построение эйлерова пути/цикла)

1) Создаём список инцидентности (для удобства лексикографически)
2) Создаём 2 стека temp и result
3) Добавляем вершины в temp, вычеркивая их из списка инцидентности
4) Если вершин для вычеркивания на текущем шаге нет, то из temp вершину вычеркиваем и добавляем в result.
5) Повтор (3) – (4) до тех пор, пока все вершины не окажутся в стеке result.



A	~~B~~ ~~C~~
B	~~A~~ ~~C~~
C	~~A~~ ~~B~~ ~~D~~
D	~~C~~ ~~E~~ ~~F~~ ~~H~~
E	~~D~~ ~~F~~
F	~~D~~ ~~E~~ ~~H~~
H	~~D~~ ~~F~~

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

start: эйлеров – любая вершина
полуэйлеров неор. – с любой из двух вершин нечет. степ.
полуэйлеров ор. – с вершины $\delta^-(v) = \delta^+(v) + 1$

38. Граф де Брюйна – определение, примеры,

$\forall$ Алфавит $T = \{\alpha_1, \dots, \alpha_n\}$, $|T| = n$.

T^* – все слова в алфавите T

T^k – все слова длины k в алфавите T,

(слово – последовательность символов в алфавите).

$|T^k| = n^k \quad (\underline{n}\,\underline{n}\,\underline{n} \dots \underline{n})$

Граф де Брюйна для слов длины k в алфавите T:

$B(T, k)$, $V = T^k$

$u, v \in V \Rightarrow \exists e = \overline{uv}$

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

$v = \boxed{u_2 \dots u_k} v_{k+1}$

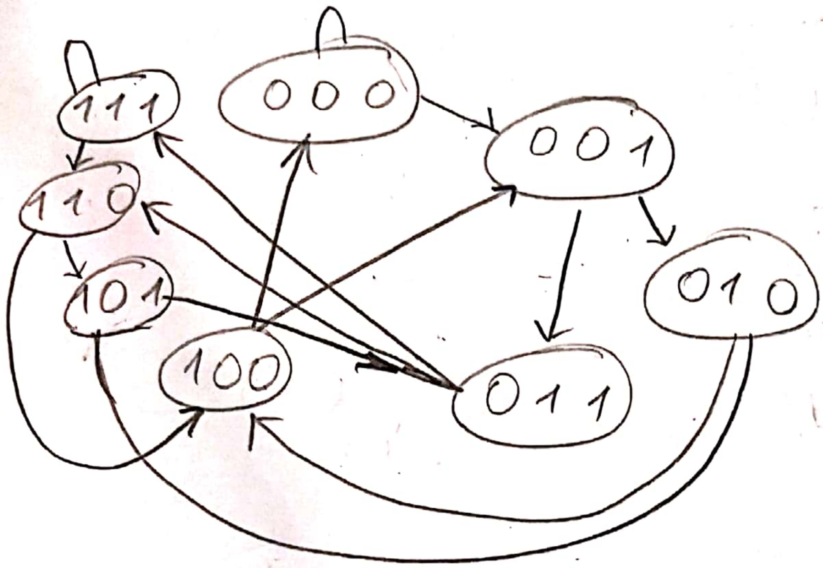
$u \xrightarrow{u_1 \boxed{u_2 \dots u_k} v_{k+1}} v$

Св-ва:

1. Могут быть петли
2. Ребра могут быть одно- и дву- направленными
3. Граф эйлеров
4. Биекция между парами вершин и ребрами $|T^k| = n^k$, $|E| = n^{k+1}$

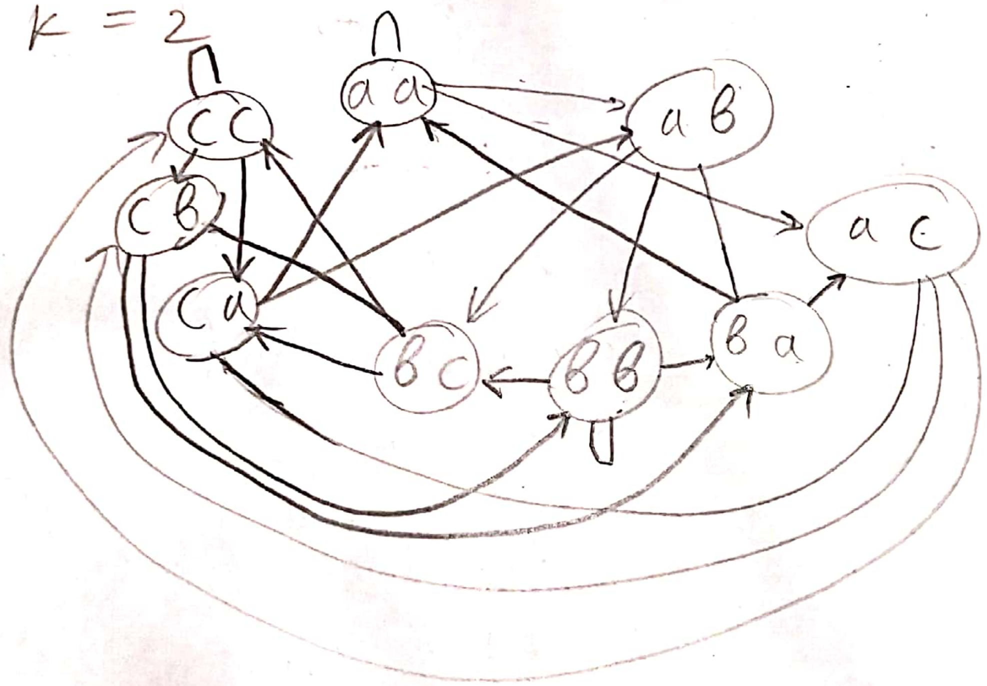
$T = \mathbb{B}$

$k = 3$



$T = \{a, b, c\}$

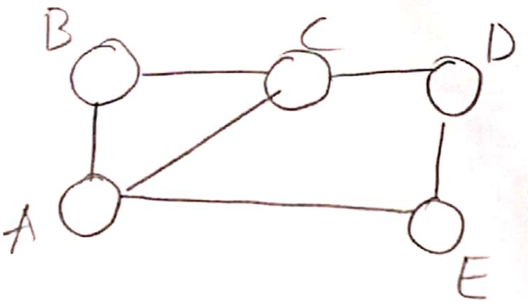
$k = 2$



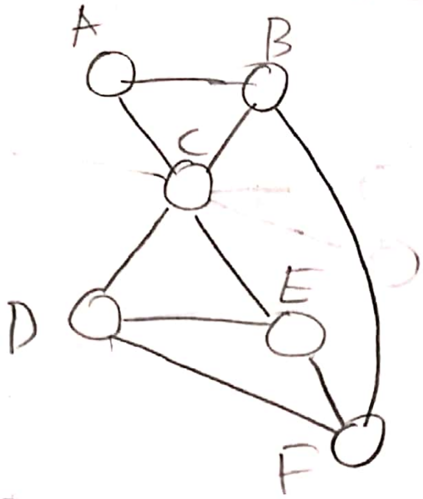
39. Гамильтонов цикл – определение. Теорема Оре.

def

<u>Гамильтонов цикл</u> – замкнутый путь, содержащий $\forall v \in V$ ровно 1 раз, кроме start $v \in V$ = finish $v \in V$



A B C D E A



$G(V,E)$ – неор, $n \geq 3$, $|V| = n$

B A C D E F B

<u>Теорема Оре</u> (достаточное условие гамильтоновости)

$\forall\ u, v \in V$: u, v – не смежные

$$\boxed{\deg u + \deg v \geq n}$$

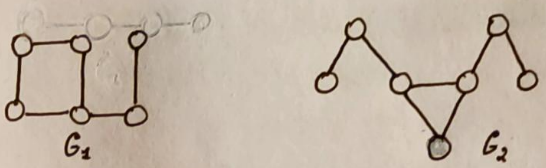
Вопросы КиТГ

(40.) Полугамильтонов граф — определение. Теорема Дирака. ⌈G(V,E) — неориентированный⌋

def⌋ Гамильтонов путь — открытый путь, содержащий каждую вершину графа, причем ровно один раз.

def⌋ Полугамильтонов граф — неориент. граф G, содержащий гамильтонов путь, называется полугамильтоновым

Примеры полугамильтоновых графов:



G_1 G_2

Теорема Дирака (достаточное условие гамильтоновости). Если степень любой вершины графа G больше либо равна половине количества вершин в этом графе, то граф является гамильтоновым.

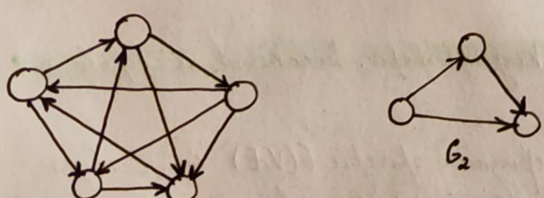
[$G(V,E)$ — неор. граф, $|V| = n \geq 3$. Если $\forall u \in V$ $\deg u \geq \frac{n}{2}$ $\Rightarrow$ G — гамильтонов]

(41.) Гамильтонов путь — определение. Турнир — определение, пример.

Определение 1 см. в п. 40 (с примерами)

def⌋ Турнир — полный граф $G(V,E)$, в котором все ребра направленные.

Примеры турнира:



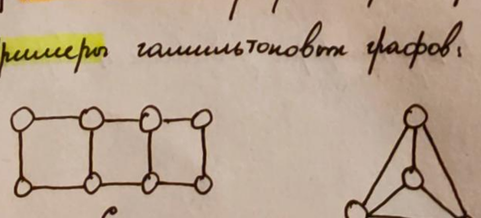
G_1 G_2

(42.) Гамильтонов граф — определение. Теорема Редеи-Камиона.

def⌋ Гамильтонов цикл — замкнутый путь, содержащий любую вершину, кроме начальной, ровно один раз (начальную — два раза).

def⌋ Гамильтонов граф — неориентированный граф G, содержащий гамильтонов цикл

Примеры гамильтоновых графов:



G_1 G_2

def| Турнир – полный граф $G(V,E)$, в котором все рёбра направленные

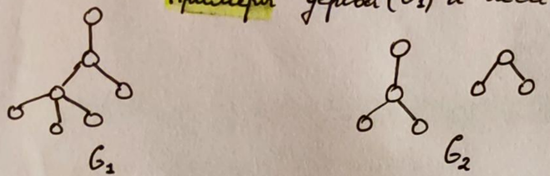
Теорема Редеи-Камиона. Любой сильно связный турнир – гамильтонов.

43. Дерево, лес – определение. Свойства деревьев (не меньше 4)

def| Дерево – связный ациклический граф.

def| Лес – граф, являющийся набором непересекающихся деревьев.

Примеры дерева (G_1) и леса (G_2)



Свойства деревьев: $G(V,E)$ - дерево

- G – связный, $|V| = n \Rightarrow |E| = n-1$
- Цикломатическое число дерева:
  $\mu(G) = |E| - |V| + 1 = 0$
- Любое дерево является двудольным графом
- Каждый связный граф $\tilde{G}(V,\tilde{E})$ допускает остовное дерево, которое является деревом, содержащий каждую вершину $\tilde{G}$ и чьи рёбра являются рёбрами $\tilde{G}$
- Любые две вершины графа G соединены единственным простым путем
- При добавлении в G любого ребра для несмежных вершин появляется один простой цикл

Г • ацикличный – это по определению,
• связный ┘ – но, возможно, подойдет и как свойство

44. Остовное дерево, остовный лес – определение. Код Прюфера. Восстановите дерево по данному коду Прюфера.

def| Остовное дерево – для произвольного связного графа $G(V,E)$ существует дерево $T(V,\tilde{E})$, содержащее все вершины графа G и чьи рёбра являются рёбрами G.

def| Остовный лес – для произвольного несвязного графа $G(V,E)$ для каждой компоненты связности можно построить остовное дерево T_i. $T_1 \cup \ldots \cup T_k$ – остовный лес.

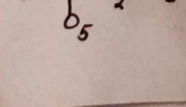
def| Код Прюфера – способ взаимно однозначного кодирования деревьев с n вершинами (помеченных) с помощью последовательности из $n-2$ чисел.

Алг. построения кода Прюфера:

Дано дерево $T(V,E)$

step: Срезаем лист с минимальным номером, в код Прюфера добавляем вершину, смежную с этим листом.

Пример построения $Pruf(T)$:



$\Rightarrow Pruf(T) = \{3,2,3,2\}$

Алг. восстановления дерева по коду Прюфера:

Дано: $Pruf(T)$ – код Прюфера, A – список всех вершин T

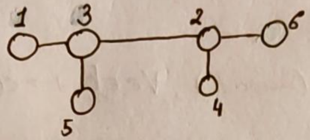
step: Берем первую невычеркнутую вершину из $Pruf(T)$ и минимальную вершину (невычеркнутую) из списка A, которой в коде Прюфера на данном шаге нет

Когда $Pruf(T)$ опустеет, соед. оставшиеся 2 вершины из A

Пример построения дерева по его коду Прюфера:

$Pruf(T) = \{3,2,3,2\}$

$A = \{1,2,3,4,5,6\}$ $\Longrightarrow$



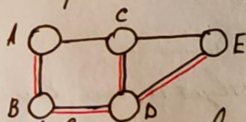
(45.) Главный цикл – определение. Разложите данный замкнутый путь в данном графе в сумму главных циклов.

$G(V,E)$ – неориентированный граф, $T(V,\tilde{E})$ – остовное дерево графа G

def **Хорда** (соответствующая T) – ребро графа G, не являющееся ветвью остовного дерева T: $\hat{e} \in E \setminus \tilde{E}$

def **Главный цикл** – цикл, получающийся при добавлении в остовное дерево T какой-либо хорды $\hat{e}$

Примеры главных циклов:



граф G_1 и его остовное дерево T_1

$\Longrightarrow$ гл. циклы:
$C_1 = (ABCDA)$
$C_2 = (CDEC)$

Теорема. $G(V,E)$, $T(V,\tilde{E})$. $C_1, C_2, \dots, C_k$ – набор главных циклов, порожденных T.

Любой замкнутый простой путь в G можно представить в виде суммы главных циклов. ($z = C_{i_1} \oplus \dots \oplus C_{i_s}$)

Алгоритм разложения:

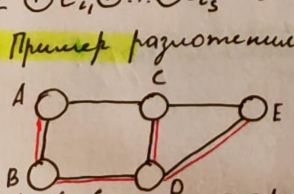
Если z – уже главный цикл, то разложение уже построено.

Пусть z – не главный цикл.

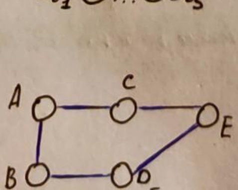
Рассматриваем хорду e_i, входящую в z и главный цикл, содержащий e_i. Когда все такие хорды рассмотрены, получится равенство

$$z \oplus C_{i_1} \oplus \dots \oplus C_{i_s} = 0 \Longrightarrow z = C_{i_1} \oplus \dots \oplus C_{i_s}$$

Пример разложения:



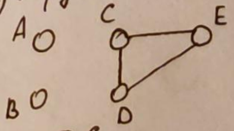
граф G и его ост. дерево T



замкнутый путь

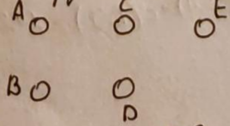
главные циклы:
$C_1 = (ABCDA)$
$C_2 = (CDEC)$

1) хорда AC



$z \oplus C_1$

2) хорда CE



$z \oplus C_1 \oplus C_2$

$\Longrightarrow z = C_1 \oplus C_2$

46. Задача о максимальном потоке в сети – постановка задачи. Постройте максимальный поток в данной сети при помощи данного алгоритма.

def| **Сеть:**

$G(V,E)$ – ориентированный граф, $S, F \in V$:

S – исток, $\delta^+(S)=0$,

F – сток, $\delta^-(F)=0$.

Все ребра – направленные

Граф G – взвешенный; $c: E \to \mathbb{Z}$ (либо $\mathbb{R}$), $\forall e \in E \mapsto c(e) \geq 0$

def| **Поток** в сети G: $f: E \to \mathbb{Z}$ (либо $\mathbb{R}$), такая что:

1) $\forall e \in E \quad 0 \leq f(e) \leq c(e)$

2) $\forall u \in V \setminus \{S,F\} \quad \sum\limits_{\substack{e_1 \in E \\ \overrightarrow{e_1}\bullet u}} f(e_1) - \sum\limits_{\substack{e_2 \in E \\ \overset{\bullet}{u}\overrightarrow{e_2}}} f(e_2) = 0$

Задача о максимальном потоке состоит в определении максимального количества, которое можно провезти через сеть из S в F.

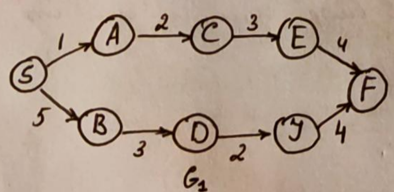
def| **Разрез** в сети $G(V,E)$: $\tilde{E} \subseteq E$, такой, что

$G(V, E \setminus \tilde{E})$ – перестает быть связным

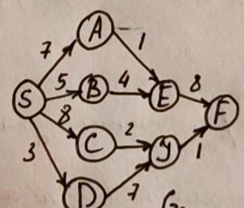
$\tilde{E}$ – минимальный по включению набор ребер

Теорема Форда-Фалкерсона. Максимальный поток в сети существует и его величина совпадает с суммой весов ребер $\sum\limits_{e \in \tilde{E}} c(e)$, где $\tilde{E}$ – минимальный разрез, отделяющий S и F

Примеры сетей:



G_1

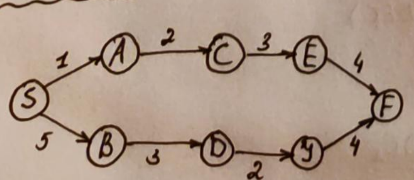


G_2

P.S. Алгоритм описывать не стану, где посмотреть знаете

47. Как связаны задача о максимальном потоке и минимальный разрез. Проиллюстрируйте примером.

см. п. 46.



G_1

min разрез: SA, DJ

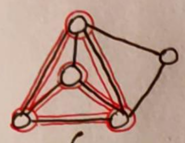
MAX поток: $c(SA) + c(DJ) = 1+2=3$

48. Клика — определение. Приведите пример графа с данной максимальной кликой.

def| **Клика** — полный подграф $\tilde{G}$ графа G.

$\tilde{G} \subseteq G$ - клика $\Longleftrightarrow$ $\tilde{G}$ - полный подграф в G

Примеры графов, содержащих клику:



G_1, MAX клика $\tilde{G}_1$ (K_4)

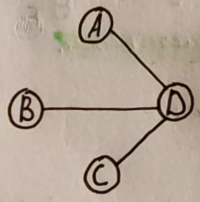
G_2, MAX клика $\tilde{G}_2$ (K_5)

49. Независимое множество — определение. Приведите пример графа с данным максимальным независимым множеством.

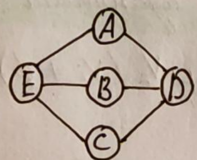
def| $W \in V$ графа $G(V,E)$ — **независимое множество** $\Longleftrightarrow \forall x,y \in W \; \nexists e = xy \in E$

Примеры:

$W = \{A, B, C\}$



$W = \{A, B, C\}$



50. Как связаны понятие клики и независимого множества? Приведите пример графа с данными максимальной кликой и независимым множеством или докажите, что это невозможно.

Определения клики и независимого множества см. в п. 48 и п. 49

def| $G(V,E)$, $|V| = n$

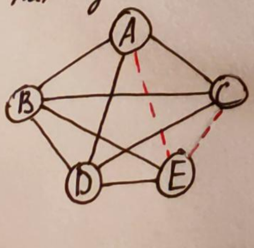
Граф-дополнение к G: $\bar{G}(V, \tilde{E})$, где $K_n = (V, \hat{E})$ и $\tilde{E} \sqcup E = \hat{E}$

[т.е. граф-дополнение содержит ребра, которых не хватает в G для получения K_n. Имея максимальное независимое множество и множество всех вершин в графе, можно построить граф-дополнение. Можно построить полный граф, содержащий вершины клики и независимого мн-ва, после чего выкинуть ребра, которые входят в граф-дополнение]

Примеры:

Клика: $\{A, B, C, D\}$

max нез. мн-во: $\{A, E\}$, $\{E, C\}$



51. Плоский граф — определение. Приведите примеры: плоского графа, планарного, но не плоского графа, непланарного графа, обоснуйте.

def | $G(V,E)$ - **плоский** $\Longleftrightarrow$ на плоскости нет самопересечений

def | $G_1(V_1,E_1)$ и $G_2(V_2,E_2)$ - **изоморфны**

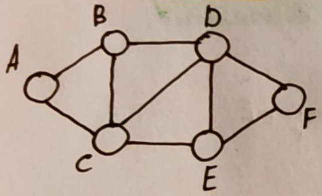
$\Updownarrow$

$\exists$ согласованные биекции $f_V: V_1 \to V_2$, $f_E: E_1 \to E_2$

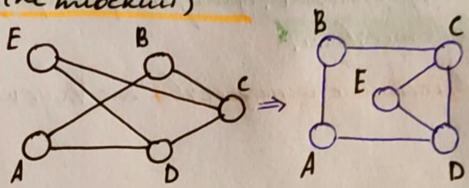
def | Граф - **планарный**, если он „изоморфен" плоскому

Примеры:

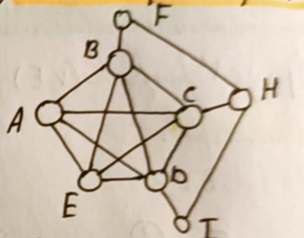
а) Плоский граф:



б) планарный граф (не плоский)



в) непланарный граф



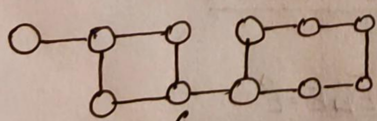
52. Двудольный граф — определение. Критерий двудольности. Приведите пример двудольного графа с заданными характеристиками.

def | Граф $G(V,E)$ - **двудольный** $\Longleftrightarrow$ $V = V_1 \sqcup V_2$ $(V_i \neq \varnothing)$, такие что:

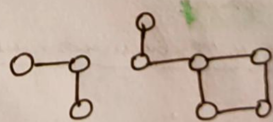
$\forall e \in E: e = v_1 v_2$, где $v_1 \in V_1, v_2 \in V_2$.

Теорема (критерий двудольности). Граф $G(V,E)$ - двудольный $\Longleftrightarrow$ граф не содержит циклов нечетной длины.

Примеры двудольных графов:



G_1



G_2

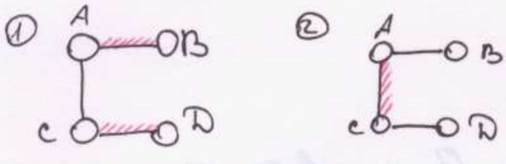
Вопросы к 1-й части экзамена по КиТГ.

53 „Максимальное и наибольшее паросочетание - определения, примеры, иллюстрирующие различие между этими понятиями. Приведите пример с заданными характеристиками."

Максимальное (насыщенное) паросочетание - это такое паросочетание, в которое невозможно добавить больше рёбер.

Наибольшее паросочетание - это паросочетание с наибольшим количеством рёбер.

Примеры:



1 и 2 - максимальные паросочетания, т.к. при добавлении любого ребра появится ребро, инцидентное вершине, являющейся концом другого ребра из паросочетания.

1- наибольшее паросочетание, т.к. в нём больше рёбер, чем в любом другом возможном паросочетании.

54 „Метрические характеристики графа - определения. Приведите пример графа с заданным набором метрических характеристик."

$G(V, E)$ - связный граф.

Эксцентриситет вершины - максимальное расстояние от вершины v до другой вершины графа ($exc(v) = \max dist(v, \omega)$)

Радиус графа ($r(G)$) - минимальный эксцентриситет графа.

Диаметр графа ($d(G)$) - максимальный эксцентриситет графа.

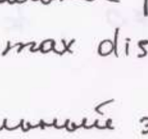
Центр графа ($Cent(G)$) - набор вершин с минимальным эксцентриситетом ($Cent(G) = \{v \in V \mid exc(v) = r(G)\}$).

55 „Правильная вершинная раскраска графа - определение. Хроматическое число - определение. Приведите пример графа с данным хроматическим числом (и некоторым заданным набором характеристик)."

Правильная вершинная раскраска графа - это такая раскраска вершин графа, при которой смежные вершины разного цвета.

Хроматическое число ($\chi(G)$) - минимальное кол-во цветов, в которые можно раскрасить граф.

Пример графа с $\chi(G)=4$:



56 „Хроматический многочлен графа - определение. Свойства (не менее 5)."

Хроматический многочлен - это функция $P_G: \mathbb{N} \to \mathbb{N}_0$, $\forall \lambda \in \mathbb{N}$: $P_G(\lambda)$ - количество различных правильных раскрасок графа G в λ цветов.

Свойства:

① $P_G(x)$ - многочлен.

② $\deg(P_G(x)) = n$, где $n = |V|$

③ $a_n = 1$ - старший коэфф.

④ $G = G_1 \sqcup \ldots \sqcup G_k$ - объединение компонент связности $\Rightarrow P_G(x) = \prod_{i=1}^{k} P_{G_i}(x)$

⑤ $P_G(0) = 0$

$P_G(x) = x^k \cdot g(x)$, где $g(x)$ - мн-н, $g(0) \neq 0$, k - количество к.с.

⑥ $|a_{n-1}| = |E|$

⑦ a_i - знакочередующиеся.

57 „Дан хроматический многочлен графа. Восстановите максимально возможное количество информации о графе."

$P_G(x)$ - хроматический многочлен.

$G(V,E)$ - граф.

$|V| = \deg(P_G(x))$

$|E| = |a_{n-1}|$

$x^k \cdot g(x) = P_G(x)$, где $g(0) \neq 0$, k - кол-во компонент связности.

58 „ Рёберно-взвешенный граф - определение. Минимальное остовное дерево - определение. Алгоритм Прима."

Рёберно-взвешенный граф - $G(V,E)$, $f: E \to M$, где M - мн-во упоряд.

Минимальное остовное дерево - это остовное дерево с минимально возможной суммой весов для графа.

Алгоритм Прима.

верш	ребро	Σ
v_i	—	0
⋮	⋮	⋮
все верш	⋮	⋮

Старт: выбирается любая вершина графа и записывается в столбец вершин, в столбец с суммой весов пишется 0.

Шаг: на каждом шаге выбирается вершина из ещё необработанных вершин, в которую ведёт ребро с минимальным весом. Вес ребра добавляется к Σ.

Шаги повторяются пока не будут обработаны все вершины.

Лемма: На любом шаге алг. Прима получается дерево.

59 „Рёберно-взвешенный граф - определение. Минимальное остовное дерево - определение. Алгоритм Краскала"

Определения см. в 58

Алгоритм Краскала.

верш	ребро	Σ
—	—	0
...	...	...

~~Шаги: Выбирается и добавляется ребро с минимальным весом, вершина . . Вес ребра добавляется к Σ.~~

Шаги: Выбирается ребро с минимальным весом из ещё необработанных рёбер так, чтобы не появился цикл. Вес ребра добавляется к Σ.

60 „Сколько минимальных остовных деревьев существует в данном графе? Ответ обоснуйте."

61 „Кратчайший путь в рёберно-взвешенном графе - определение. Критерий корректности задачи о нахождении кратчайшего пути."

$G(V,E)$ - ориентированный, рёберно-взвешенный. ($\varphi: E \to \mathbb{R}$)

$A, B \in V$, тогда путь $A \xrightarrow{e_{i_1}} \dots \xrightarrow{e_{i_k}} B$ называется кратчайшим, если $\sum_{j=1}^{k} \varphi(e_{ij}) \to \min$.

Критерий корректности:

В графе должны отсутствовать циклы такие, что $\sum_{e_{ij} \in e_n} (\varphi(e_{ij})) < 0$.

62 „Алгоритм Дейкстры - постановка задачи, описание, пример."

Задача: найти кратчайшие пути от вершины v до других вершин графа.

Старт: Выбираем начальную вершину. Ставим метки на вершинах, на начальной 0, на остальных ∞.

Шаг: Ω - множество ~~н~~обработанных вершин.

Выбираем вершину $\omega \in (V \setminus \Omega)$, такую, что метка ω минимальна.

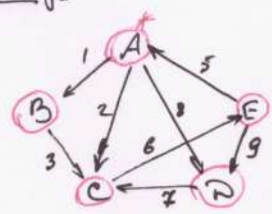
Просматриваем все рёбра, исходящие из ω в ещё необработанные вершины.

Сравниваем $\sigma(u)$ с $\sigma(\omega) + \varphi(\omega u)$, где $\sigma(\ )$ - метка, u - вершина, в которую ведёт ребро из ω.

Если $\sigma(u) > \sigma(\omega) + \varphi(\omega u)$, то $\begin{cases} \sigma(u) := \sigma(\omega) + \varphi(\omega u) \\ u \text{ получает предка } \omega. \end{cases}$

После просмотра всех путей из ω выбираем новую вершину.

Пример:



* - начальная вершина

	0	1	2	3	4	5
A	0, A	0, A	0, A	0, A	0, A	0, A
B	∞, -	1, A	1, A	1, A	1, A	1, A
C	∞, -	2, A	2, A	2, A	2, A	2, A
D	∞, -	8, A	8, A	8, A	8, A	8, A
E	∞, -			8, E	8, E	8, E

$$\begin{tikzcd} D^8 & A^0 \arrow[l] \arrow[r] \arrow[d] & C^2 \arrow[d] \\ & B^1 & C^8 \end{tikzcd}$$

63 „Алгоритм Форда - Беллмана - постановка задачи, описание, пример."

Задача: найти кратчайшие пути из вершины v в остальные вершины графа.

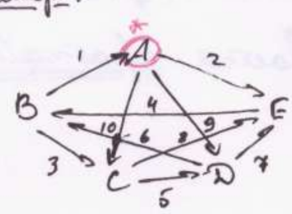
Старт: Выбираем вершину, ставим ей метку 0, остальным вершинам ∞.

Выписываем все рёбра в определённом порядке.

Шаг: Проходим по списку рёбер. Для ребра vu сравниваем $\sigma(u)$ с $\sigma(v) + \varphi(vu)$, если $\sigma(u) > \sigma(v) + \varphi(vu)$, то

$\begin{cases} \sigma(u) := \sigma(v) + \varphi(vu) \\ u \text{ получает родителя } v. \end{cases}$

Пример:



Список рёбер.

E	вес
BA	1
AC	10
AD	9
AE	2
BC	3
DB	6
EB	4
CD	5
CE	8
DE	7

	0	1		2	3
A	0, A			0, A	0, A
B	∞, -	~~15, D~~	6, E	6, E	6, E
C	∞, -	10, A		9, B	9, B
D	∞, -	9, A		9, A	9, A
E	∞, -	2, A		2, A	2, A

$$\begin{tikzcd} A^0 \arrow[r] \arrow[d] & E^2 \arrow[r] & B^6 \arrow[r] & C^9 \\ D^9 \end{tikzcd}$$

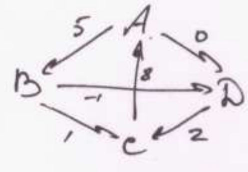
(64) „Алгоритм Флойда - постановка задачи, описание, пример."

<u>Задача</u>: найти кратчайшие пути между вершинами графа.

<u>Старт</u>: Выписываем веса рёбер в таблицу.

<u>Шаг</u>: ~~Пров~~ Выбираем столбец и строку вершины v. Сравниваем значение в ячейке с суммой значений ячеек в столбце и строке. Если в ячейке значение больше, то заменяем его на сумму, а также записываем рёбенка.

<u>Пример</u>:



$h^0 =$

	A	B	C	D
A	0	5	∞	0
B	∞	0	1	-1
C	8	∞	0	∞
D	∞	∞	2	0

$h^1 =$

	A	B	C	D
A	0	5	∞	0
B	∞	0	1	-1
C	8	13,A	0	8,A
D	∞	∞	2	0

$h^2 =$

	A	B	C	D
A	0	5	6,B	0
B	∞	0	1	-1
C	8	13,A	0	8,A
D	∞	∞	2	0

$h^3 =$

	A	B	C	D
A	0	5	6,B	0
B	9,C	0	1	-1
C	8	13,A	0	8,A
D	10,C	15,C	2	0

$h^4 =$

	A	B	C	D
A	0	5	2,D	0
B	9,C	0	1	-1
C	8	13,A	0	8,A
D	10,C	15,C	2	0

(65) „Алгоритм Джонсона - постановка задачи, описание, пример."

<u>Задача</u>: найти кратчайшие пути между вершинами.

<u>Старт</u>: $G(V,E)$, $p: E \to \mathbb{R}$

$\tilde{p}: E \to \mathbb{R} \geq 0$

$\forall v \in V: h(v)$ - Джонсоновский потенциал.

Вначале достраиваем к графу фиктивную вершину, а также рёбра из фиктивной вершины в каждую вершину графа с весом 0.

Выписываем список рёбер, выбираем в качестве начальной вершины фиктивную вершину. Далее проводим алгоритм Дейкстры. Полученные значения для вершин - $h(v)$. Заменяем веса рёбер по формуле $\tilde{p}(uv) = p(uv) + h(u) - h(v)$.

<u>Шаг</u>: аналогичен алг. Дейкстры.