ЛЕКЦИЯ 07-09

Бинарные отношения

<u>Опр.</u>: M - множество $\neq \varnothing$

$R \subset M \times M$ - бинарное отношение

<u>Пояснение</u>

$M \times M$ - множество пар из элементов R

Допустим: $M = \{a, b, c\}$

$M \times M = \{(a,a); (a,b); (a,c); (b,a); (b,b); (b,c); \dots\}$

или: $M = \mathbb{N}$

$M \times M = \mathbb{N} \times \mathbb{N} = \{(1,1), (1,2), (2,1), (1,3), (2,2), \dots\}$

Отношение R - это подмножество пар

<u>Обозначение</u>

Вместо $(x, y) \in R$ - пара (x, y) принадлежит отношению мы будем писать xRy

Вместо $(x, y) \notin R$ мы будем писать $x \not R y$

<u>Примеры</u>

1. $M = \mathbb{R}$; $R = \{(x, y) : x > y\}$

$(3, 2) \in R$ $\quad 3R2$

$(3, 4) \notin R$ $\quad 3 \not R 4$

2. $M = \mathbb{R}$. Отношение $\geq$

$7 \geq 6$ $\quad 7 \geq 7 \quad \cancel{7 \geq 8}$

3. $M = \mathbb{R}$. Отношение $=$

$7 = 7 \quad \cancel{7 = 8}$

$(7,7) \in \; = \qquad (7,8) \notin \; =$

4. $M = \mathbb{R}$, $\approx$: $x \approx y \Leftrightarrow |x-y| < 1$

5. $M = \mathbb{R}$, $\#$: $x \# y \Leftrightarrow x^2 > y$

$2 \# 2 \quad \cancel{1 \# 2} \quad 7 \# 8 \quad \cancel{7 \# 100}$

6. $M = \mathbb{N}$ или $M = \mathbb{Z}$, $\vdots$: $x \vdots y \Leftrightarrow \exists k \in \mathbb{Z} : x = ky$

$4 \vdots 2 \quad 10 \vdots 5 \quad \cancel{7 \vdots 0}$

$\cancel{2 \vdots 4} \quad \cancel{10 \vdots 3} \quad 0 \vdots 0$

7. $M = \mathbb{Z}$, $\equiv_3$

$0 \equiv_3 3 \quad 1 \equiv_3 4 \quad \cancel{0 \equiv_3 2} \quad 1 \equiv_3 7 \quad \cancel{1 \equiv_3 8}$

8. $M = \mathbb{N}$, ц

a ц b если в числе "a" есть "b" цифр.

100 ц 3 $\quad$ 238 ц 3

~~238 ц 8~~

9. $M =$ прямые на $\mathbb{R}^2$; $//$

$l_1 // l_2$ если l_1 не пересекать l_2 или $l_1 = l_2$

10. $M =$ прямые на R^2 ; $\perp$

$l_1 \perp l_2$

11. $M =$ Студент ЛЭТИ

$x > y$: средний балл за последнюю сессию x больше чем y

12. $M =$ пользователи одноклассники

$x \to y$, если "y" в друзьях у "x"

Иванов $\to$ Петров

~~Петров $\to$ Носов~~

Свойства бинарных отношений

1. Опр. Бинарное отношение R на M называют рефлексивным, если

$\forall x \in M : xRx \qquad ((x,x) \in R)$

Замечание: Отношение не рефлексивно $\Leftrightarrow \exists x : \cancel{xRx}$

Примеры: $=$ рефлексивно $\forall x : x = x$

$\geq$ " $\forall x : x \geq x$

$\approx$ " $\forall x : x \approx x ; |x-x| = 0 < 1$

$\vdots$ " $\forall x : x \vdots x$

$>$ не рефлексивно $\cancel{2>2}$

$\vdots$ не рефлексивно $\cancel{3 \vdots 3}$

$\perp$ "

2. Опр: Бинарное отношение R на множестве M называют антирефлексивным, если $\forall x : \cancel{xRx}$

Замечание: R - не антирефлексивно $\Leftrightarrow \exists x : xRx$

Примеры: $>$ антирефлексивно $\forall x : \cancel{x > x}$

$\perp$ " $\forall$ прямые $\cancel{l \perp l}$

$\rightarrow$ " нельзя быть в друзьях у себя

$\vdots$ не антирефлексивно $1 \vdots 1$

Замечание:

1) $\vdots$ - не рефлексивно и не антирефлексивно

2) не бывает R, которое и рефлексивно, и антирефлексивно

(рассмотрим $a \in M \rightarrow aRa$ не ар

$\cancel{aRa}$ не р.)

3. Опр: Бинарное отношение R на множестве M симметричное, если $\forall x, y : xRy \Leftrightarrow yRx$

Замечание: R - не симметрично $\Leftrightarrow \exists x, y : xRy, \cancel{yRx}$

Примеры: $=$ симметрично $x = y \Leftrightarrow y = x$

$\approx$ " $x \approx y \Leftrightarrow y \approx x$

$|x-y| = |y-x| < 1$

$\vdots$ не симметрично $x \vdots y$ $\cancel{y \vdots x}$

$//, \perp$ симметрично $a // b \Leftrightarrow b // a$

$a \perp b \Leftrightarrow b \perp a$

$\vdots$ не симметрично $100 \vdots 3$ $\cancel{3 \vdots 100}$

4. Опр: Бинарное отношение R на множестве M антисимметричное, если $\forall x \neq y$ $xRy \Rightarrow \cancel{yRx}$

Замечание: R - не антисимметрично, если

$\exists x \neq y : xRy$ ~~⇔~~ yRx

Примеры: $>$ антисимметрично

Попробуем построить контрпример

$x \neq y$, $x > y$, $y > x$ - не возможно

$\Rightarrow$ нет контрпримера $\Rightarrow$ антисимметрично

$\geqslant$ антисимметрично

$x \neq y$. $x \geqslant y$, $y \geqslant x$ - нет к. пр.

$=$ антисимметрично

$x \neq y$ $x = y$, $y = x$ - нет к. пр.

$\equiv_3$ не антисим. $1 \neq 4$ $1 \equiv_3 4$ $4 \equiv_3 1$ - к.пр.

$\vdots$ над N антисим $x \neq y$ $x \vdots y$ $y \vdots x$ - нет.

$\vdots$ над Z не антисим $4 \neq y-4$ $4 \vdots -4$ $-4 \vdots 4$

(Лекция 14-09)

5. Опр: R - бин. отн. на M - асимметричность, если

$\forall x, y \quad xRy \Rightarrow \cancel{yRx}$

($x \neq y$ - антисимметричность)

Контрпример: xRy, yRx

Утверждение.

R - симметрично $\Leftrightarrow$ R - антисимметрично и антирефлексивно

Пример:

$>$ асимметрично $\forall x, y: x > y \Rightarrow \cancel{y > x}$

$\square$ (пустое) асимметрично (пустое, когда $R = \varnothing$)

"выше" асимметрично

"начальник" " $\forall x, y: x$ "нач" $y \Rightarrow$ ~~y нач x~~

ЛЕКЦИЯ 14-09

6. Опр: R - бин. отн. транзитивно, если

$\forall x, y, z \quad xRy, yRz \Rightarrow xRz$

Примеры

$>$ транзитивно $x > y, y > z \Rightarrow x > z$

$\geqslant$ " $x \geqslant y, y \geqslant z \Rightarrow x \geqslant z$

$\vdots$ " $x \vdots y, y \vdots z \Rightarrow x \vdots z$

$\perp$ не " $x \perp y, y \perp z \Rightarrow \cancel{x \perp z}$

ц не " 100 ц 3 3 ц 1 $\Rightarrow$ ~~100 ц 1~~

7. Опр: R - бин. отн. эквивалентность, если

R рефлексивно, симметрично, транзитивно.

Примеры:

$=$ эквивалентно $\forall x: x = x$ - рефлек.

$\forall x, y: x = y \Rightarrow y = x$ - симметрично

$\forall x, y, z: x = y, y = z \Rightarrow x = z$ - тран.

// эквивалентно

$\equiv_3$ "

$\geqslant$ не " т.к. не симметрично.

$\approx$ н " т.к. не транзитивно.

Scanned with CamScanner

$\uparrow$ на $\mathbb{N}$: $x \uparrow y$, если у "x" и "y" поровну цифр (đều nhau)
(số c/s = nhau)

$2 \uparrow 5$ $\quad$ $35 \uparrow 100$ $\quad$ $\cancel{29 \uparrow 7}$

$\left.\begin{array}{l} \forall x: x \uparrow x \text{ - рефлек.} \\ \forall x,y: x \uparrow y \Rightarrow y \uparrow x \text{ - симме.} \\ \forall x,y,z: x \uparrow y, y \uparrow z \Rightarrow x \uparrow z \text{ - тран.} \end{array}\right\} \Rightarrow$ эквива.

8. Опр.: R - отн. эквивалентности на множестве M; $x \in M$

Класс элемента x: $M_x = \{y \mid xRy\}$

Примеры:

$=$ $\quad$ $M_5 = 5$

$\equiv_3$ $\quad$ $M_2 = \{2, 5, 8, 11, \dots\}$

$\parallel$ $\quad$ $M_a = \{b, c, d \mid b \parallel c \parallel d \parallel a\}$

Утверждение

R - отн. эквива. на M

$\forall x, y \in M$: $M_x = M_y$ или $M_x \cap M_y = \emptyset$

Доказательство:

$\Rightarrow$ $M_x \cap M_y \neq \emptyset \Rightarrow \exists z \in M_x; z \in M_y \Rightarrow xRz, yRz$

R - симметрично $\Rightarrow zRy$

R - транзитивно $\Rightarrow xRy$ (xRz, zRy)

Теперь проверим, что класс $M_x = M_y$

Возьмем $u \in M_x$, проверим что $u \in M_y$

$u \in M_x \Rightarrow xRu$

$xRy \Rightarrow yRx \Rightarrow yRu \Rightarrow u \in M_y$

Следствие:

R - отн. экв. на M, тогда M разбито на несколько классов эквивалентности

$M = M_1 \cup \dots \cup M_n$

$M_i \cap M_j = \emptyset$

Примеры:

$=$ на N $\quad$ 1 и 2 и $3 \dots$

$\equiv_3$ на N $\quad$ $0, 3, 6, 9 \dots$

$1, 4, 7, 10 \dots$

$2, 5, 8, 11 \dots$

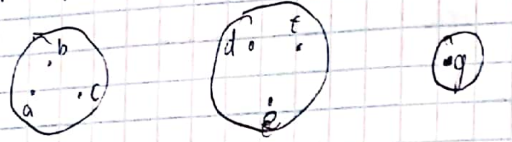
Замечание: Если есть $M = \emptyset$ разбитое на $M_i = \emptyset$

$M = M_1 \cup \dots \cup M_n$ и $M_i \cap M_j$

Тогда можно ввести отношения R

xRy, если $\exists M_i: x, y \in M_i$

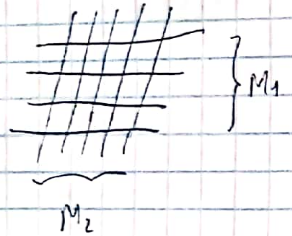
Примеры:

+) 

a, b, c, d, e, f, g

aRb bRc $\cancel{gRd}$ $\cancel{gRf}$ gRg $\cancel{gRa}$

+) Для // классы эквивалентности



+) Отношения порядка (выше, лучше, сильнее, важнее)

9. Опр: R - бин. отн.

R - транзитивно, антисимметрично:

1) рефлексивно - нестрогий порядок $\succcurlyeq$
2) антирефлексивно - строгий порядок. $\succ$

$a \succ b$ $b \succ c \Rightarrow a \succ c$ (тран.)

$a \succ b \Rightarrow b \not\succ a$ (антисим.)

Примеры:

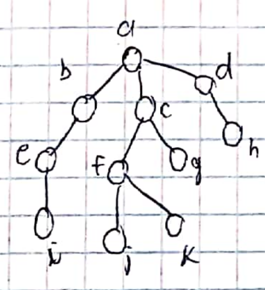
$>$ на R строгий порядок

$\geqslant$ на R не   "

$\vdots$ на R не   "

нач

a нач b a нач c b нач f c нач f



10. Опр: $\exists\, R$ - строгий порядок или нестрогий порядок

R - линейный, если $\forall x \neq y$ xRy или yRx

R - частичный иначе $\exists\, x \neq y$ $\cancel{xRy}$ $\cancel{yRx}$

Примеры:

$>, \geqslant$ · линейный порядок

$\vdots$ частичный порядок $\cancel{2 \vdots 3}$ $\cancel{3 \vdots 2}$

нач   "

Утверждение

R - порядок ~~(строгий или нестрогий)~~ (строгий или нестрогий) на

M - конечное $|M| < \omega$. Тогда $\exists\, x$ - минимальный, т.е. $\forall y: x \not\succ y$

ЛЕКЦИЯ 21-09 (продолжение)

Утв. R - отношение порядка на $|M| < \infty$ (строгий или не строгий) тогда $\exists\, x$ - минимальный, т.е. $\forall y \neq x$ $\cancel{xRy}$

Пример:

$\geq$ на $\{1, 2, 3, 4, 5\}$

1 - мин т.к. $\forall y : \cancel{1 \geq y}$

$\vdots$ на $\{2, 3, 4, 5, 6\}$

2 - мин, $\forall y : \cancel{2 \vdots y}$

3 - мин $\forall y : \cancel{3 \vdots y}$

5 - мин $\forall y : \cancel{5 \vdots y}$

Д-во: Берём x_1 - $\forall$ элемент множества

1 Если он не мин $\Rightarrow \exists\, x_2 \neq x_1$ $\quad x_1 > x_2$ (это R)

Если x_2 не мин $\Rightarrow \exists\, x_3 \neq x_2$ $\quad x_2 > x_3$

Если x_3 " $\Rightarrow \exists\, x_4 \neq x_3$ $\quad x_3 > x_4$

Если мы не можем найти минимальный элемент.

$\Rightarrow$ т.к. множество M конечно, в какой-то момент $x_i = x_j$ (первый повтор)

$$x_i > x_{i+1} > x_{i+2} > \ldots > x_{j-1} > x_j = x_i$$

$>$ - транзитивно $\Rightarrow \{ x_i > x_{j-1}$ и $x_{j-1} > x_i \Rightarrow$ невозможно. если антисимм.

Опр. Отношение R_1 на множество M расширяет R_2 на M, если

$$R_2 \subset R_1$$

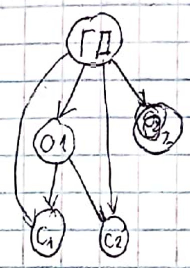
Замечание R_1 "доставляет" пары, где xRy

Замечание $xR_2y \Rightarrow xR_1y$

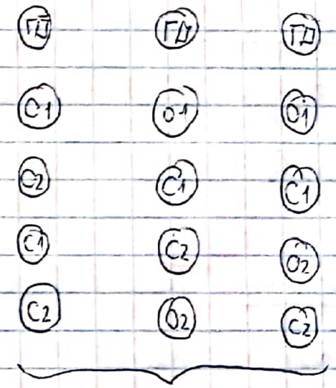
Th о топологической сортировке

Если $>$ - отн. порядка (стр. или нестр.) на M, то $\forall$ то $\exists \gg$ - отн. линейного порядка на M, и $\gg$ расширяет $>$

Пример: ...



- не линейный порядок



топологическая сортировка

Д-во:

Находим минимальный элемент отношения $>$. $\exists$ это $x_1 \in M$.

Удалим x_1 из M, теперь имеем $>|_{M \setminus \{x_1\}}$

Очевидно, новое отношение тоже антирефл., антисимм., транз.

в нём тоже есть мин. элемент, это x_2.

Scanned with CamScanner

Удалим x_2 из M и продолжаем

Итого имеем последовательность $x_1, x_2, \ldots, x_n$ $(n = |M|)$

Вводим новый порядок $x_i \ll x_j$ для $i < j$

$x_1 \ll x_2 \ll \ldots \ll x_n$

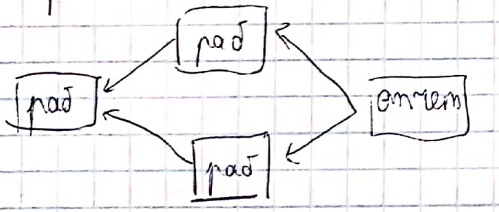
Почему $\ll$ расширять $<$

Если $x < y \Rightarrow x$ был удален раньше y

$\Rightarrow x \ll y$.

<u>Замечание</u>. Этот алгоритм (поиска мин. и удаления) не самый эффективный. Лучше - сделать поиск в глубину с обратной

<u>Замечание</u> Топологическая сортировка - практически важная задача порядок работ



p_1 p_2 p_3

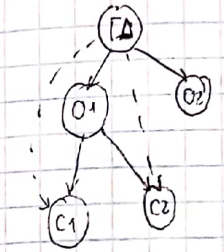
p_1 p_2 p_3 } топ. сорт.

Транзитивное замыкание

был порядок - расширяем до линейного (Топ. Сорт)

было отношение - расширяем до транзитивного (Тр. замк.)

Пример: поод



ГД R О1 ~~ГД R С1~~

О1 R С1

Если добавить в отношение, что ГД R С1, ГД R С2 - станет транзитивным.

$\underline{\text{Th}}$ $\exists R$ - бин. отношение на M.

$\exists \bar{R}$ - отношение на M

1) $\bar{R}$ расширяет R $(R \subset \bar{R})$

2) $\bar{R}$ транзитивно

3) $\bar{R}$ - мин. транзитивное расширение, т.е если $\bar{\bar{R}}$ - тр. расширение R, то $\bar{\bar{R}} \supset \bar{R}$

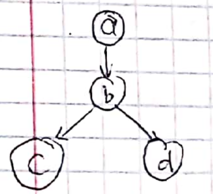
Д-во (не для алгоритма)

Рассмотрим все транзитивные расширения

$\{\bar{R}_i\}$ возьмем $\bar{R} \cap \bar{R}_i$

т.е. берём только те ~~подмн.~~ пары, которые есть во всех транзитивных расширений

Пример: $M = \{a, b, c, d\}$

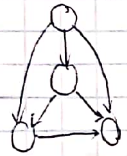


aRb

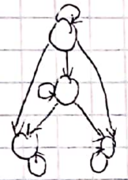
bRc

bRd

$\bar{R}_1$ -

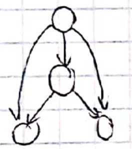


$\bar{R}_2$ -



$aRa \quad bRb$

$\bar{R}$ =



- только те пары, которые есть везде

Проверим, что $\bar{R}$ подходит под условия

1) $\bar{R}$ - расширяет

$\exists\, xRy \Rightarrow \forall \bar{R}_i : x\bar{R}_i y \Rightarrow x\bar{R}y$

2) $\bar{R}$ - транзитивно

$\exists\, x\bar{R}y,\ y\bar{R}z \Rightarrow \forall \bar{R}_i : x\bar{R}_i y,\ y\bar{R}_i z \Rightarrow x\bar{R}_i z \Rightarrow x\bar{R}z$

3) $\bar{\bar{R}} = \bar{R}_i \supset \bar{R}$ т.е. $R = \bar{R}_i \cap \dots$

4) $\bar{R}_i$ существует? $\bar{R}_i$ = полное отн. $= M \times M$

⊗ Графы

Опр. неориентированный граф.

$G = (V, E)$, где V - множество (вершины)

$E \subset \{(u, v)$, где $u, v \in V\}$

пара неупорядоченная

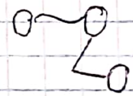
Замечание Как рисовать.

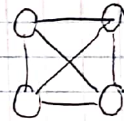
Вершины • или O

Ребра - линии между O

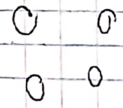
Важно только .

Примеры

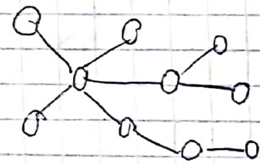




- полный



- пустой



Опр. G - полный, если $\forall u, v \in V \quad (u, v) \in E$

Замечание: V = vertex $\quad E$ = edge

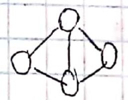
Опр. $|G|$ - размер (порядок) графа $= |V|$ - количество вершин

$|V| = n$ (обычно)

$|E| = m$ (обычно) } - количество ребер

G - это (n, m) граф

Пример:

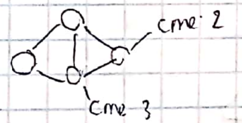


- $(4,5)$ граф.

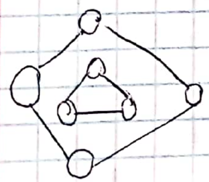
Опр. Степень вершины $v \in V$ - это $|\{(v,u) \mid (u,v) \in E\}|$ (количество ребер с этой вершиной)

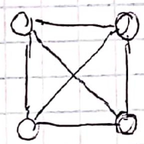
Обозначение. $\deg v$

Пример



Опр. k-регулярный граф, это граф, где $\forall v \in V \ \deg v = k$





3-рег

ЛЕКЦИЯ 28-09-2021

Напоминание

степени

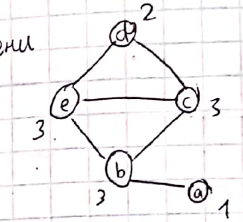


Рис. 1

$G(V,E)$

Опр. Путь в графе - последовательность

$v_1 e_1, v_2 e_2, \dots, v_{n-1} e_{n-1}, v_n$

$v_i \in V$; $e_i \in E$ $\quad e_i = (v_i, v_{i+1})$

Пример (Рис. 1)

1) a b c d

(a,b) (b,c) (c,d)

2) a b

3) a b a

4) a b c d e c d e b a → замкнут

(e,c) → не простой (2 раза de)

Опр. Замкнутый путь - если $v_1 = v_n$

Открытый путь (не замкнутый) - если $v_1 \neq v_n$

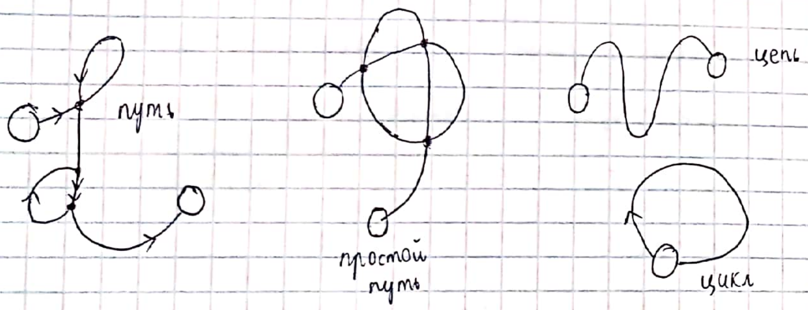
Опр. Простой путь, если $e_i \neq e_j$ при $i \neq j$

Пример (Рис 1)

5) b e d c e - простой, не замкнутый

Опр.

Пути	Все ребра разные (простые)	Все вершины разные
Замкнутый	Простой замкнутый путь	цикл
Открытый	Простой открытый путь	цепь



Теорема: Если $\exists$ путь между вершинами $u, v \Rightarrow$ есть цепь от u до v

Доказательство.

$\exists$ Путь $u\ e_1 v_1 e_2 v_2 \dots e_n v$

Рассмотрим все пути из этих ребер и выберем min, это будет цепь иначе $u \dots u_i \dots u_j \dots v$

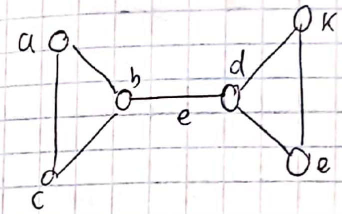
$\exists\ v_i = v_j$

укоротим $u \dots v_i = v_j \dots v$??

Теорема: Если есть простой замкнутый путь через ребро e $\Rightarrow$ есть цепи через e

Доказательство аналогично

Замечание



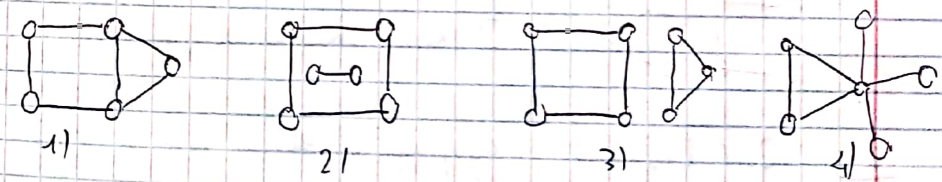
dbacbd - не простой путь (e - повторяется)

цикла через e нет

Связность графа

Опр: Граф связан, если $\forall\ u, v \in V$. $\exists$ цепь (путь) из u в v

Пример



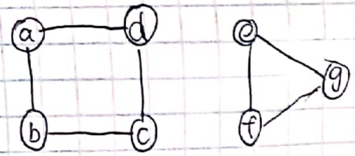
1, 4 - связаны

2, 3 - не связаны

Введем отношение $\equiv$ на вершинах графа:

$u \equiv v$, если $\exists$ путь из u в v

Пример



$a \equiv c$ $\quad e \equiv g$

$d \equiv d$ $\quad \cancel{a \equiv e}$

Проверим, что $\equiv$ - это отношение эквивалентности

1) рефлексивность $u \equiv u$ - верно, путь u

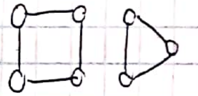
2) симметричность $u \equiv v \Rightarrow v \equiv u$ путь $u e_1 v_1 \dots v$

путь $v \dots v_1 e_1 u$

3) транзитивность $u \equiv v$, $v \equiv w$ путь $u e_1 v_1 \dots v v \dots w$

не повторяется, а входит в 2 пути.

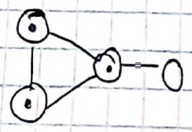
Опр: Классы эквивалентности $\equiv$ - это компоненты связности



- 2 компонента связности

Опр: $G_1 = (V_1, E_1)$ - подграф G, если $V_1 \subset V$, $E_1 \subset E$

Пример



Помеченные • являются подграфом

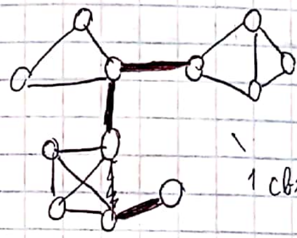
Замечание

G - свой подграф

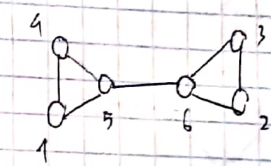
O - пустой граф - подграф чего угодно.

Опр: G ребро e называется мостом, если компонент связности G < количество компонентов связности (V, E_e)

Пример



1 связный граф



5 6 мост

▬ мост

Опр: Степень связности графа G - это количество ребер, которые надо выкинуть, чтобы G стал несвязным

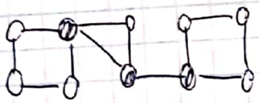
Опр: Двусвязный граф - надо выкинуть ≥ 2 ребер, чтобы он стал несвязным

Замечание: двусвязный $\Leftrightarrow$ нет мостов и связи

Опр: Вершина $v \in V$ называется точкой сочленения, если количество компонента связности^{G} < количества компонента связности$^{G'}$

$G' = (V_v, E'(u,v) \mid (v,u) \in E)$

Пример

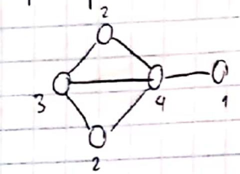


<u>Теорема</u>: В графе $G = (V, E)$, если $\deg(u)$ – степень вершины u

$$|E| = \frac{1}{2}\sum_{v \in V} \deg(v)$$

Пример

Ребер: $6 = \frac{1}{2}(3+2+2+4+1) \Rightarrow$ Верно



Доказательство

$\deg(v)$ = количество ребер, выходящих из вершин

$\sum_{v \in V} \deg(v)$ = все ребра посчитали дважды $= 2|E|$

<u>Следствие</u>:

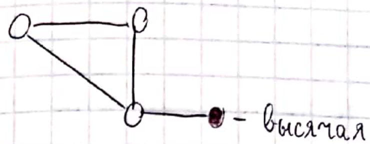
1) сумма степеней вершин всегда четна
2) вершин нечетной степени четно

Пример:

15 инопланетян, по 3 руки у каждого, могут ли они взяться за руки, чтобы не было свободных рук?

Решение: Нет, это граф из 15 нечетных вершин степени 3 (нечет)

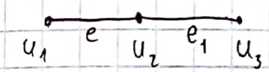
<u>Опр</u>: Висячая вершина – это вершина степени 1



<u>Теорема</u>: Если в графе есть ребра, но нет висячих вершин, то $\exists$ цикл.

Доказательство:

Берем ребро $e = (u_1, u_2)$



u_2 – не висячая вершина $\Rightarrow$ из неё есть ещё ребро (u_2, u_3)

u_3 – " $\Rightarrow$ " (u_3, u_4)

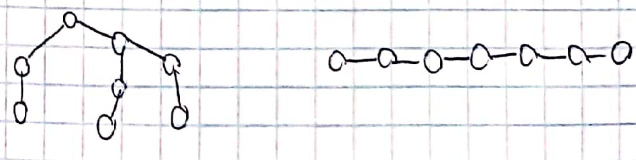
...

Продолжаем, пока очередность u_n не будет равна u_i $1 \le i \le n$

Путь $u_i\, u_{i+1} \dots u_n$ – цикл (ребра разные, вершины разные)

<u>Опр</u>: Дерево – связный граф без циклов.

Пример:



Теорема: В любом дереве ≥ 2 висячих вершины

Доказательство:

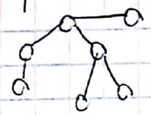
Берем $\forall$ вершину, если она не висячая, идём по ребру, если опять не висячая, есть ребро и т.д.

Циклов нет $\Rightarrow$ Конец.

Чтобы найти вторую, надо начать из первой.

Теорема: Если G - дерево, то $|V| = 1 + |E|$

Пример



7 вершин, 6 ребер.

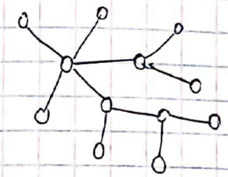
Доказательство по индукции (количество вершин)

Б. $|V| = 1$ $|E| = 0$, сходится $|V| = |E| + 1$

ЛЕКЦИЯ 05 - 10

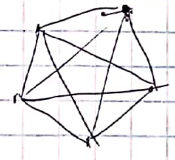
Напоминание

Дерево - связный граф без циклов



$|E| + 1 = |V|$

$\exists$ G - полный граф, $\forall\, u \neq v \in V$ соединены ребром.



Если n вершин ($|V| = n$), то ребер

1) C_n^2 ребер, выбираем пары $= \frac{n(n-1)}{2}$

2) Степени всех вершин $n-1$

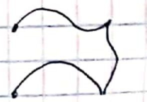
$\sum\limits_{v \in V} deg(v) = 2|E| \Rightarrow n(n-1) = 2|E|$,

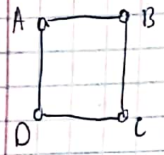
Ответ: $|E| = \frac{n(n-1)}{2}$

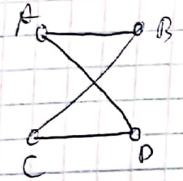
Планарные графы

Опр: G - планарный, если его можно нарисовать на плоскости так, чтобы ребра не пересекались

ребро $[0,1] \to R^2$

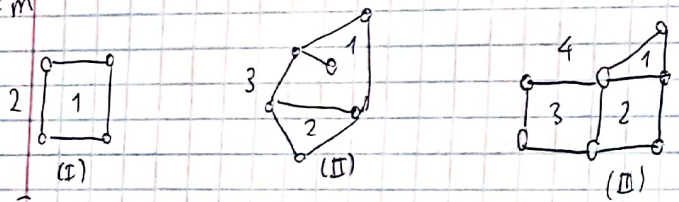






планарный, но неправильно

Формула Эйлера: Если планарный граф $G = (V, E)$ нарисован на плоскости, у него можно посчитать грани, $\exists$ их f, $|V| = n$, $|E| = m$



Берем...

Тогда $n - m + f = 2$

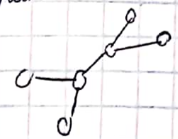
Проверим

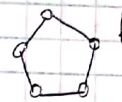
(I): $4-4+2 = 2$ (II): $6-7+3=2$ (III): $7-9+4=2$

Д-во: индукция по количеству ребер

База: G - дерево

1 грань





вокруг грани есть цикл

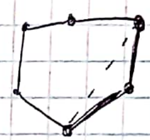
Дерево без циклов.

$$n - m + f = n - (n-1) + 1 = 2$$

Переход: G - не знаем, верно ли (G, G' - связные планарные)

если G' имеет меньше ребер $\Rightarrow$ верно

G - не дерево $\Rightarrow$ есть цикл, берем ребро цикла



вокруг него 2 грани

удалим ребро, получим G' - тоже связен и планарен

n', m', f' - вершины, ребра, грани G'

$n' = n$

$m' = m-1$

$f' = f-1$

По индукции верно.

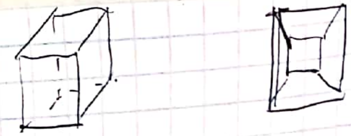
$n' - m' + f' = 2 \Rightarrow n - (m-1) + (f-1) = 2$

$\Rightarrow n - m + f = 2$

Следствия

1) Неважно, как рисовать планарный граф, кол-во граней постоянно

2) Для многогранники также



$8 - 12 + 6 = 2$

3) Если G планарный (не обязательно связный) то

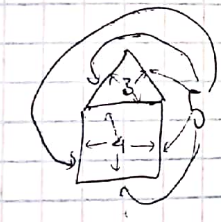
$n - m + f = 1 + |\text{компонент связности } G|$

Д-во. Упр.

4) У каждой грани вокруг ≥ 3 ребра

$3f \leq \sum_{g \in \text{грани}} \text{кол-во ребер вокруг } g \leq 2m$

каждое ребро посчитано 1 или 2 раза



$\Rightarrow 3f \leq 2m$

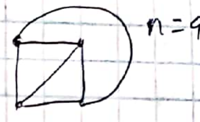
но $n - m + f = 2$

$3n - 3m + 3f = 6 \Rightarrow 3n - 3m + 2m \geq 6$

$\Rightarrow 3n - m \geq 6 \Rightarrow m \leq 3n - 6$

Итого $m \leq 3n - 6$ в связном планарном графе.

Следствие: Полный граф при $n = 5$ - не планарен



Д-во: $n = 5$, $m = \frac{5 \cdot 4}{2} = 10$

$10 \leq 3 \cdot 5 - 6 = 9$??

Замечание: K_5 - полный граф $n = 5$

Утверждение: граф $K_{3,3}$ тоже не планарный

Д-во: $n = 6$, $m = 9$

$9 \leq 3 \cdot 6 - 6$ ✓

нет противоречия

Сколько граней, если планарный

$6 - 9 + f = 2 \Rightarrow f = 5$ граней

В $K_{3,3}$ все циклы четные (ходим лево-право или право-лево)

$\Rightarrow$ У грани ≥ 4 ребра

- невозможно

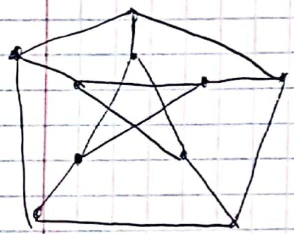
$4f \leq \sum_{g \in \text{грани}} \text{ребер грани } g \leq 2m \Rightarrow m \geq 2f$ $\quad 9 \geq 2 \cdot 5$ ✗

Scanned with CamScanner

Теорема Понтрягина-Куратовского

Граф G планарен $\Leftrightarrow$ если не содержит подграфов, стягивающихся к K_5 и $K_{3,3}$

– не планарен
стягивается к $K_{3,3}$

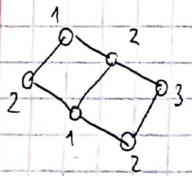


– не планарен, в нём есть K_5
можете поискать $K_{3,3}$, тоже есть

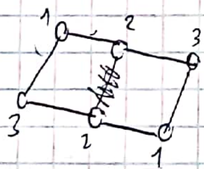
Хроматизм

Опр.: Пусть $G=(V,E)$ - граф

раскраска графа G в k цветов это функция $C: V \to \{1..k\}$ (целое)
причем, если есть ребро (u,v), то $C(u) \neq C(v)$

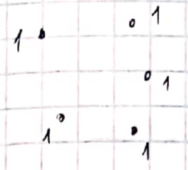


- раскраска в 3 цвета



- не раскраска

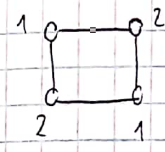
Какие графы можно раскрасить в 1 цвет?

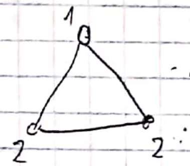


- это графы без ребер

Какие графы можно раскрасить в 2 цветах?

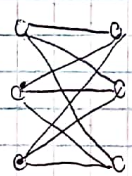
Опр.: Граф G двудолен если его можно раскрасить в 2 цвета



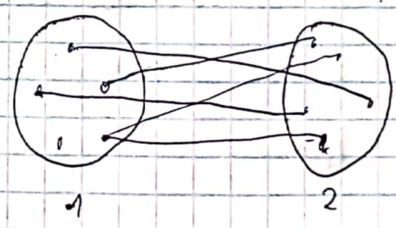


- не двудолен

$K_{3,3}$ - двудольный

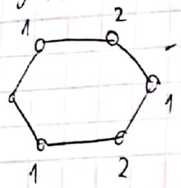


Замечание: Двудольные графы часто рисуют из двух частей (долей)



Th: G - двудолен ⇔ все циклы G имеют четную длину

Д-во: 1) двудолен ⇒ циклы четные



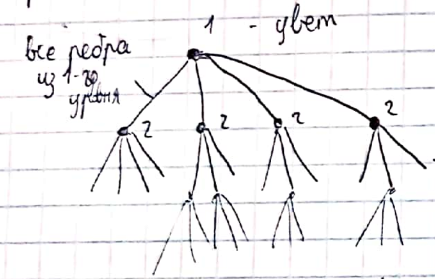
- цикл

поровну цвета 1 и цвета 2

2) циклы четные ⇒ двудолен

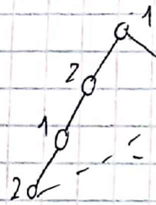
"подвесим граф за вершину"

∀ вершины



ребра назад не рисуем

Назначаем цвета по уровням



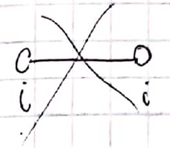
Почему обратное ребро не соединяет одинаковые цвета?

Потому что иначе цикл нечетный

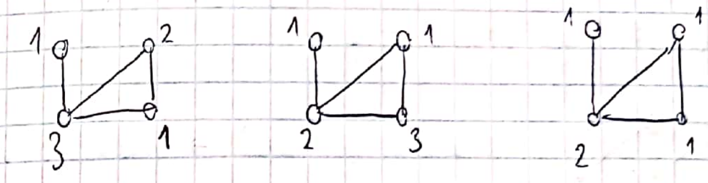
ЛЕКЦИЯ 12 - 10

Напоминание

k - раскраска. k цветов из вершин.



нельзя ребром один цвет

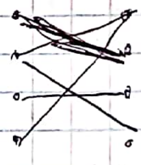


- не раскраска

Граф имеет 1-раскраска ⇔ Граф без рёбер.

Граф имеет 2 раскраски ⇔ двудольный граф (по опр.)

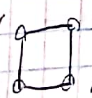
Зам

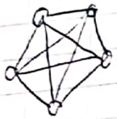


Th. Граф двудолен ⇔ все циклы чётные.

Опр: $G=(V,E)$ - граф

$\chi(G)$ - хроматическое число графа минимальное количество цветов, в которых его можно раскрасить.

Пример: $\chi($  $) = 3$ $\chi($  $) = 2$

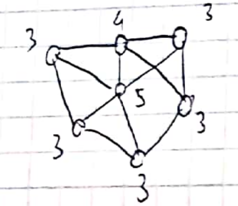
$\chi\Big($  $\Big) = 5$ (все ~~д~~ цвета вершин должны быть разными)

$\chi(K_n) = n$

полный граф n вершин

Зам. Если $k \geq \chi(G)$, то G можно покрасить в k цветов

Утв. $\chi(G) \leq \max \deg v + 1$



(степени вершин) $\to \max \deg v = 5$

$\chi(G) \leq 5+1$

Д-во: индукция по количеству вершин

Б. $n=0$, $n=1$ — верно $\max \deg = 0$, $\chi(G) \leq 1$

П. G, v — вершина $\max \deg = \Delta$

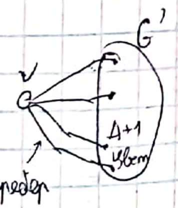
Уберем её, получим $G' = G$ без v

$\mathrm{Max} \deg G' \leq \mathrm{Max} \deg G - \Delta$

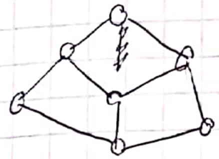
Раскрасим G' в $\Delta + 1$ цвет

цвет v запрещен $\leq \Delta$ цветов

$\Rightarrow$ 1 цвет можно



Утв. G — планарный граф $\Rightarrow \chi(G) \leq 5$



Д-во:

1) в G есть вершина степени ≤ 5 $\quad |V| = n$; $|E| = m$

Если нет, $\Rightarrow \deg v \geq 6 \Rightarrow \sum \deg v \geq 6n$

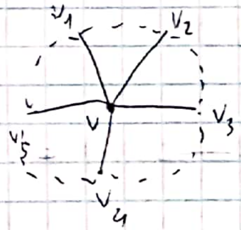
$\Rightarrow 2m \geq 6n \Rightarrow m \geq 3n$, но в планарном G: $m \leq 3n-6$

Раскрасим в 5 цветов по индукции

Б. Графы из 1, 2, 3, 4, 5 вершин — можно раскрасить

П. $\exists$ у неё n вершин (для $n-1$ вершин есть раскраска)

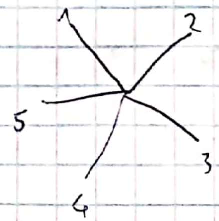
Берем v: $\deg v \leq 5$



Раскрасим G' без v

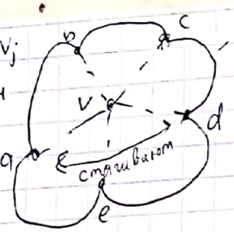
если соседа v_1 используют ≤ 4 цветов $\Rightarrow$ для v есть цвет

Осталось



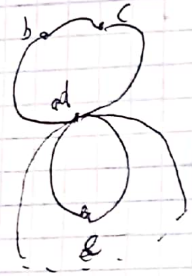
$\exists$ — все 5 цветов

если v_i, v_j все соединены ребром $\Rightarrow$ есть K_5 $\Rightarrow$ G не планарен !



1 грань без v, сделаем $\tilde{G}$

$\tilde{G}$:



$\tilde{G}$ - $n-2$ вершин

$\Rightarrow$ можно раскрасить

вернёмся к G, a и d имеют 1 цвет

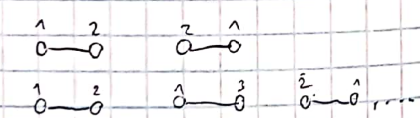
$\Rightarrow$ есть цвет для V

Утв. $\chi(G) \le 4$ (проблема 4ех красок)

Хроматические многочлены

Опр: $\exists\ \chi(G, k)$ - это функция "сколько способов раскрасить G в k цветов"

$$\chi(\text{o—o}, k) = \begin{cases} k=0 & 0 \\ k=1 & 0 \\ k=2 & 2 \\ k=3 & 6 \\ k=4 & 12 \\ \text{иначе} & k(k-1) \end{cases}$$



$\chi(\text{o—o}, k) = k(k-1)$

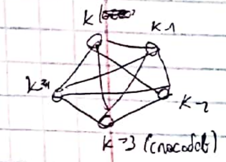
$\chi(\text{o o}, k) = k^2$

Утв: 1) $\chi(\varnothing_n, k) = k^n$

граф из n вершин без ребер

2) $\chi(K_n, k) = \underbrace{k(k-1)(k-2)\dots(k-n+1)}_{n \text{ множителей}}$

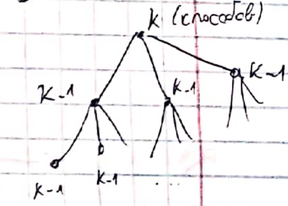
полный



3) $\chi(T_n, k) = k(k-1)^{n}$ подвесим дерево за $\forall$ вершину

дерево

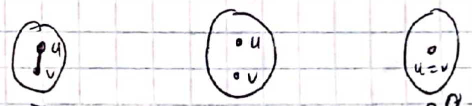
$k-1$ цвет возможен



4) $\bar{G}$ - граф ; u, v вершины с ребром (u,v)

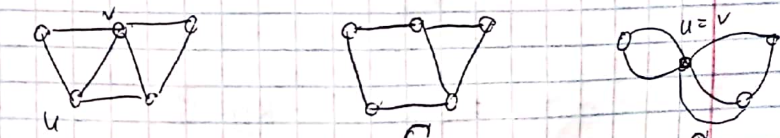
$G^{\bullet} = \bar{G} \setminus (u,v)$ (без ребра)

$G^{0} = \bar{G}$, где u, v стянуть в вершину



$\bar{G}$ $G^{\bullet}$ G^{0}

Пример:



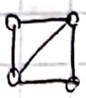
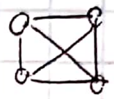
$\bar{G}$ G G^{0}

$$\Psi(G,k) = \Psi(\overline{G},k) + \Psi(G^{\circ},k)$$

$\downarrow$ способы раскрасить G, где u и v - разный цвет

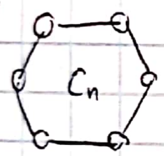
$\downarrow$ способы раскрасить G, где u, v - один цвет

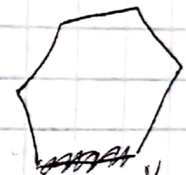
Следствие: $\boxed{\Psi(\overline{G},k) = \Psi(G,k) - \Psi(G_0,k)}$

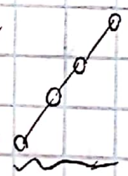
Примеры: $\Psi($  $, k) = \Psi($  $, k) + \Psi($  $, k)$

(полный) (полный)

$$= k(k-1)(k-2)(k-3) + k(k-1)(k-2)$$

5) $\Psi($  $, k) = ?$



$\Psi(C_n, k) = \Psi($  $, k) - \Psi(C_{n-1}, k) =$

$$= k(k-1)^{n-1} - \Psi(C_{n-1},k) =$$

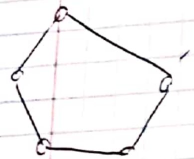
$$= k(k-1)^{n-1} - k(k-1)^{n-2} + \Psi(C_{n-2},k) =$$

$$= \ldots =$$

$$= k(k-1)^{n-1} - k(k-1)^{n-2} + \ldots + (-1)^{n-2} k(k-1) + (-1)^{n-1}\Psi(C_1,k)$$

$$= k(k-1)^{n-1} - k(k-1)^{n-2} + \ldots + (-1)^{n-2} k(k-1) + (-1)^{n-1} k =$$

ЛЕКЦИЯ 19-10



, C_5 цикла из 5 вершин

$\chi(C_n, k)$ - сколько способов раскрасить C_n в k цветов

$\chi(T_n, k) = k(k-1)^{n-1}$
↓
дерево

$\chi(K_n, k) \neq k^n = k(k-1)\dots(k-n+1)$
↓
полный

$\chi(C_n, k) = k(k-1)^{n-1} - k(k-1)^{n-2} + k(k-1)^{n-3} - \dots + (-1)^n k(k-1) + (-1)^{n+1} k \ominus$

$\ominus$ ...

Утв: 1) $\exists\, G$ имеет висячую вершину u: $\chi(G, k) = \chi(G', k) \cdot (k-1)$

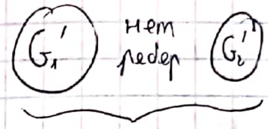
раскрасить u

2)

u соединена с v_1, v_2 и $(v_1, v_2) \in G$

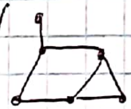
$\chi(G, k) = \chi(G', k) \cdot (k-2)$

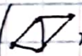
3) $\exists\, G = G_1' \vee G_2'$ Не ребер между G_1' и G_2'



$\chi(G, k) = \chi(G_1', k) \cdot \chi(G_2', k)$

Пример:

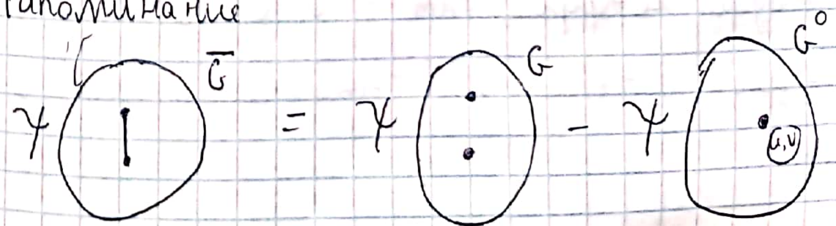
$\chi\Big($  $, k\Big) = (k-1)\,\chi\Big($  $, k\Big) =$

$= (k-1)(k-2)\,\chi\Big($  $, k\Big) =$

$= (k-1)(k-2)^2\,\underbrace{\chi(\Delta, k)}_{K_3} =$

$= (k-1)(k-2)^2\, k(k-1)(k-2)$

Напоминание



Утв: $\Psi(G,k)$ — это многочлен

1) старший коэффициент $= 1$

2) степень $= n$ (количество вершин)

3) знаки чередуются

4) младший коэффициент $= 0$

5) коэффициент при $k^{n-1} = \pm m$ (количество ребер)

Док: Индукция по количеству вершин, при равном кол-ве вершин : количество ребер

База: пустой граф из n вершин $\Psi(\vdots\vdots, k) = k^n$

Переход: $\overline{G}$

с ребром $\Psi(|, k) = \underbrace{\Psi(:, k)}_{\text{работает}} - \underbrace{\Psi(\cdot, k)}_{\text{работает}}$

1) Старший коэффициент $(1 k^n) - (k^{n-1}) \ldots$

2) степень $= n$

3) $(k^n - k^{n-1} + k^{n-2}) - (k^{n-1} - k^{n-2} + k^{n-3} + \ldots)$

4) младший ~~коэф~~ коэффициент $= 0 - 0 = 0$

5) $-$ ребер $G \cdot k^{n-1} - k^{n-1} = -$(количество ребер $G + 1) k^{n-1}$

На практике

$\Psi(\hat{\square}, k) = (k-1)\Psi(\Delta, k) = (k-1)k(k-1)(k-2) =$

$= k(k-1)^2(k-2) = k^4 - 4k^3 + 5k^2 - 2k$

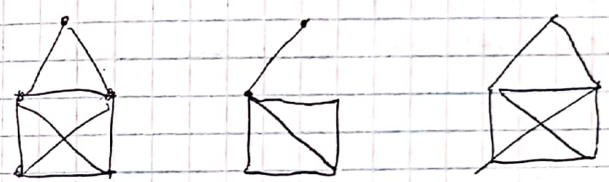
~~Раскроем скобки~~

Утв: $\Psi(G)$ — хром число (мин. число цветов для раскраски)

$\Psi(G,k)$ $k = 0, 1, 2, \ldots \Psi(G) - 1$ — корни многочленов

$\Psi(G)$ — не корень

Эйлеровы графы



Нарисовать не провед дважды по одному ребру

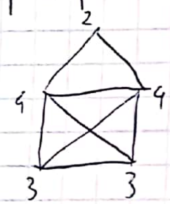
Опр:

1) Эйлеров путь — ~~простой~~ путь, содержащий все ребра, не проходим дважды по 1 ребру

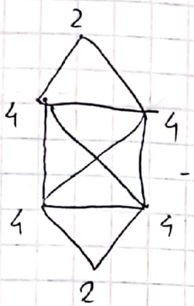
2) Эйлеров цикл — цикл, содержащий все ребра, не проходим дважды по 1 ребру

Утв: $G = (V, E)$, содержит эйлеров цикл $\Leftrightarrow$ G связен, и $\deg v$ – чёт $\forall\, v \in V$

Пример



– нет Эйл. цикла



– есть эйл. цикл

Д-во: $\Rightarrow$ Граф связен

кол-во входов = кол-во выходов

$\to$ deg чётно

$\Leftarrow$ Начнём строить цикл

... из $\forall$ вершин, выбираем ребро, которая ещё не использовалось

в каждой вершине по пути использовано чёт. ребер (k входов, k выходов)

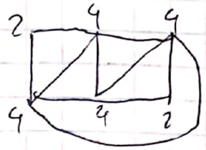
+1 ребро, через которое вошли

$\Rightarrow$ использовали нечет ребер

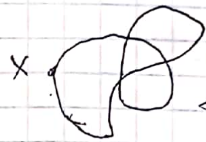
$\Rightarrow$ есть ещё одно, по нему можно уйти

Кроме начальной из неё вышли на 1 раз больше

$\Rightarrow$ мы закончим ходить в начальной вершине.



обошли не всё, выкинем просмотренные ребра

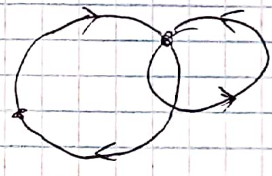


$\leftarrow$ построенная часть в остатке все степени чётные

Т.к. G связен, из начальной все вершины X можно попасть в $\forall$ вершину и ребро

Повторим процесс из $\forall \in 1$ циклу, из которой везёт ~~мо~~ ... ребро

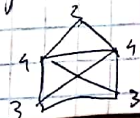
Объединим 2 цикла



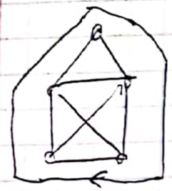
Продолжать, пока все ребра не объединяется в 1 цикл.

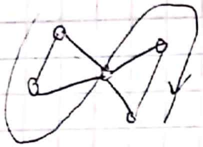
Th. G содержит эйлеров путь $\Leftrightarrow$

1. G связен
2. [степени всех вершин четны
   степени всех вершин кроме двух – четны
   (в этом случае нечетные вершины – это начало и конец)

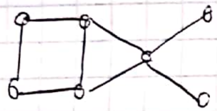


Опр: Гамильтоновы пути / циклы - простые цепи / циклы по всем вершинам





Г. путь



- нет Г. пути

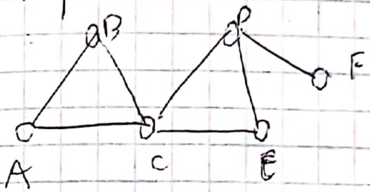
ЛЕКЦИЯ 26-10

В прошлый раз: Гамильтоновы (по всем верш.) / Эйлеровы (по всем ребрам) пути / циклы

Длины путей в граф

Опр. Длина пути в графе - кол-во рёбер в пути

Пример



ABCDF - путь от A до F

длина 4 (4 ребра)

ACEDF - длина 4

ACDF - длина 3

ABCEDF - длина 5

Опр: Расстояние между вершинами - мин. длина путь между вершинами или $+\infty$, если путь нет

Обозначение: $d(X,Y)$ - расстояние от X до Y

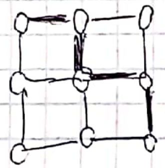
Пример: $d(A,F) = 3$

Опр: Диаметр графа - макс. расстояние между вершинами графа.

Пример:

-) В примере выше, диаметр = 3 (достигается на AF)

Scanned with CamScanner

г)  - диаметр = 4

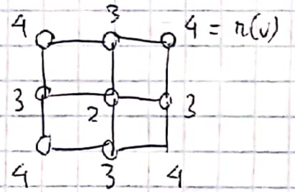
Все другие расстояния ≤ 4

Опр: Для каждой вершины графа $G = (V, E)$ можно посчитать ~~max~~ расстояние до других вершин

$$r(v) = \max\{d(v,s) \mid s \in V\}$$

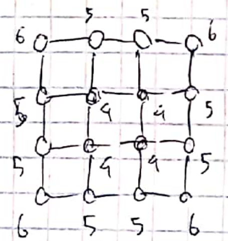
Радиус: $r(G) = \min\{r(v) \mid v \in V\}$

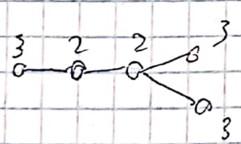
те вершины, на которых достигается min - это центр.



$r(G) = 2$ - радиус графа.

Центров может быть много



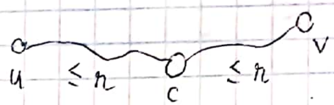


4 центр
$r(G) = 4$

2 центр
$r(G) = 2$

Утв: $G = (V, E)$: $d(G) \leq 2r(G)$

Д-во: $\exists$ c - центр графа, $u, v \in V$



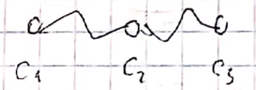
$d(c,u) \leq r$

$d(c,v) \leq r$

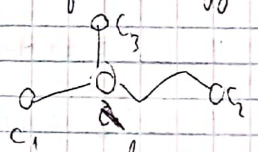
$\Rightarrow d(u,v) \leq 2r \Rightarrow d(G) = \max\limits_{u,v} d(u,v) \leq 2r$

Утв. В дереве ≤ 2 центра

Д-во: $\exists$ их 3



Построим пути между c_1 и c_2, потом c_2 и c_3 (в дереве равно 1 путь между вершинами)



вершина c_0

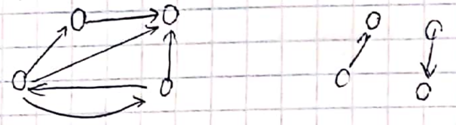
$r(c_0) < r(c_1) = r(c_2) = r(c_3) = r(G) = r$.

Замечание: Будем дальше иногда использовать ориентированные графы $G = (V, E)$

(ребра в ориентированном графе иногда называют . . .)

$E \subset \{(u,v)$ - упорядоченная пара$\}$

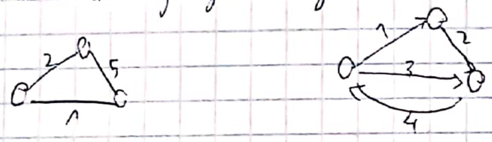
Пример



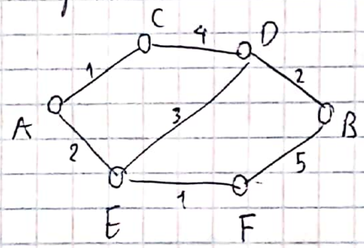
Замечание: У рёбер будут веса

$G = (V, E)$ вес - это ~~ER~~ $f : E \to R$

т.е. у число на каждому числу



Расстояние на графе с весами считается как $\min \sum$ весов по всем путям.



$d(A,B) = ?$

$d(ACDB) = 1+4+2 = 7$

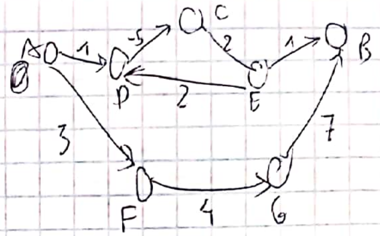
$d(ACDEFB) = 1+4+3+1+5 = 14$

$d(AEFB) = 2+1+5 = 8$

$d(AEDB) = 2+3+2 = 7$

$\Rightarrow d(A,B) = 7$

Замечание: Расстояние во взвешенном графе не всегда существует



$d(F,G) = 4$

$d(G,F) = +\infty$

$d(A,B) = ?$

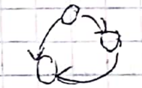
$d(ADCEB) = 1-5+2+1 = -1$

$d(ADCEDCEB) = 1-5+2+2-5+2+1 = -2$

и т.д. $\min = -\infty$

Утв: В графе есть все расстояния $\Leftrightarrow$ в графе нет цикла отрицательной длины

~~Д-во: Если есть~~ цикл < 0



$\Rightarrow \forall$ две вершины этого цикла не имеют расстояния (или $-\infty$)

Если нет расстояния, т.е. для u, v есть пути угодно маленькие, $\exists$ есть путь длинее $n = |V|$ рёбер $\Rightarrow$ повтор вершин в пути ~~Как хранятся~~

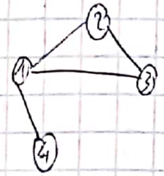
- это и будет отрица. цикл

Как хранятся графы в компьютере (представление графа в компьютере).

1. Матрица смежности: таблица вершин × вершины.

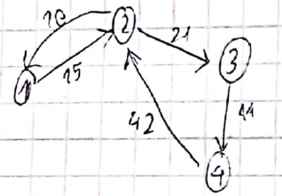
$$a(i,j) = \begin{cases} 0, & \text{если нет ребра} \\ 1, & \text{если есть ребро} \end{cases}$$

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 & 0 \\ 3 & 1 & 1 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 \end{array}$$



- симметрична для неориентированного графа

Для графов с весами $a(i,j)$ = вес ребра i, j или $+\infty$, если нет



$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & +\infty & 15 & +\infty & +\infty \\ 2 & 10 & +\infty & 21 & +\infty \\ 3 & +\infty & +\infty & +\infty & 14 \\ 4 & +\infty & 42 & +\infty & +\infty \end{array}$$

Объём памяти: $n^2 = |V|^2$

2. Списки смежности

- Для каждой вершины вершины хранит список соседей

Пример выше:

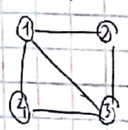
1: 2 (15)

2: 1 (10), 3 (21)

3: 4 (14)

4: 2 (42)

Пример



1: 2, 3, 4

2: 3, 1

3: 1, 2, 4

4: 1, 3

Память $\approx |E|$: кол-во рёбер.

3. Неявные способы

Умеем вычислять всех соседей $\forall$ вершины

Пример: Задача обход конём шахматной доски

Граф: вершины = клетке (64 шт)

рёбра - вершины через ход коня

Можно $\forall$ клетки (вершины) посчитать, куда можно попасть.

Задача обхода конём = Гамильтонов цикл в этом графе.

Пример: Задача: дано две вершины u, v. Найти $d(u,v)$ и путь, на котором достигаться это расстояние.

Замечание: Оказывается, что найти путь от u до v это ~~тоже самое, что~~ тоже самое, что искать путь от u до всех вершин.

Алгоритм Форда-Беллмана:

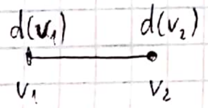
Дано $G = (V, E)$, $u \in V$, найти расстояние $d(u,v)$, для $\forall\ v \in V$.

Будем писать $d(v) = d(u,v)$, т.к. u не меняется.

Будем хранить в массиве d текущие найденные расстояния

В начале $d(u) = 0$, $d(v) = +\infty$ если $v \neq u$

Релаксация ребра $e = (v_1\ v_2)$

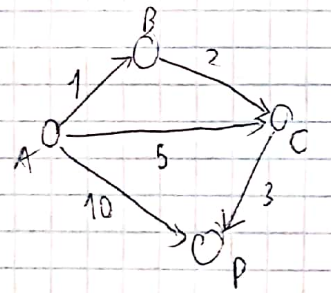


Если $d(v_1) + f(v, v_2) < d(v_2) \Rightarrow d(v_2) := d(v_1) + f(v_1\ v_2)$

Алгоритм: Повторить $n-1$ раз. Перебрать все ребра e и каждое релаксировать.

(в неориент. графе, т.е две релакс. на ребро)

Пример:



$n = 4$ (4 вершины)

Ребра

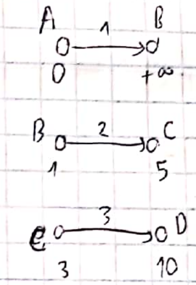
A : B(1), C(5), D(10)

B : C(2)

C : D(3)

D :

		A	B	C	D
d:		0	$+\infty$	$+\infty$	$+\infty$
Шаг 1	AB	0	1	$+\infty$	$+\infty$
	AC	0	1	5	$+\infty$
	AD	0	1	5	10
	AB	0	1	3	10
	CD	0	1	3	6
Шаг 2	AB	—	//	—	
	AC	—	//	—	
	AD	—	//	—	
	AB	—	//	—	
	CD	—	//	—	
Шаг 3	AB	—	//	—	
	AC	—	//	—	
	AD	—	//	—	
	AB	—	//	—	
	CB	—	//	—	



$0 + 1 < +\infty$

$1 + 2 < 5$

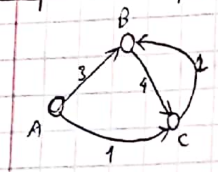
$3 + 3 < 10$ 6

Ответ: $d(A) = 0$; $d(B) = 1$; $d(C) = 3$; $d(C) = 6$

Время работы $\approx |V| \cdot |E| \leq |V|^3$

ЛЕКЦИЯ 09-11

Алгоритм Форда - Беллмана



A : 3B , 1C

B : 4C

C : 2B

Пути из A

Сначала d:

	A	B	C	
d	0	$+\infty$	$+\infty$	Релаксируем
	0	3	$+\infty$	$(\underset{0}{A} \xrightarrow{3} \underset{+\infty}{B} ;\ 0+3<\infty)$
	0	3	1	$(\underset{0}{A} \xrightarrow{1} \underset{+\infty}{C} ;\ 0+1<+\infty)$
				$(\underset{3}{B} \xrightarrow{4} \underset{1}{C} ;\ 3+4>1)$
	0	2	1	$(\underset{1}{C} \xrightarrow{1} \underset{3}{B} ;\ 1+1<3)$

$n=3 \Rightarrow n-1=2$ раза цикл релаксаций

AB, AC, BC, CB

Нет ~~AC~~

Ответ : A B C

0 2 1

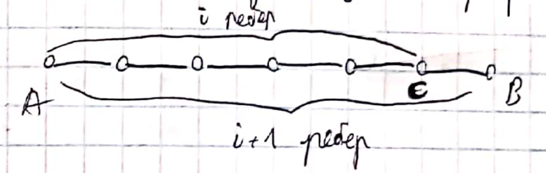
Корректность алгоритма

Th : В конце массив d содержит расстояния от A

Д-во : После i-го цикла релаксации всех ребер, d хранит числа $d(v) \leq \min$ длин путей, в которых $\leq i$ ребер.

... База индукции $i=0$; min (пути из 0 ребер) - только A - A $d(A)=0$, $d(u)=+\infty$)

$\exists$ есть опт. путь из $i+1$ ребра

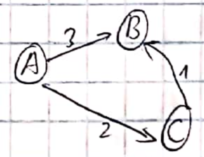


По предположению $d(C) = dist(A,C)$

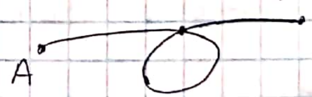
Длин пути ~~A-C-B~~ $\cancel{A}C-B = dist(C) + вес(CB)$ (ребро CB)

$= d(C)$

Проверка $d(C) + вес(CB) < d(B)$ – верно, т.к. путь A-C-B ~~опт~~ опти $\Rightarrow d(B) > d(C) + вес(CB)$



Почему n-1 этап? Оптимальный путь не содержит циклы. => Оптимальный путь $\leq n-1$ ребро



Замечание: Мы вычисляли только расстояния, но путь неизвестен.

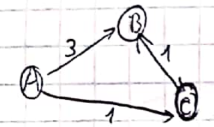
Как восстанавливать пути?

Будем сохранять информацию об успешных ~~расстояниях~~ релаксациях.

Prev - массив вершин

Если релаксация $u \longrightarrow v$ успешна, то Prev[v] = u {

(опт. путь в v лежит через u)



d	A	B	C
	0	+∞	+∞
AB -	0	3/A	+∞
AC -	0	3/A	1/A
CB -	0	2/C	1/A

(/A - через A)

Восстановить путь в B: $A \longrightarrow C \longrightarrow B$, где $A = prev(C)$, $C = prev(B)$

~~В общем случае~~

В общем случае путь $A \to v$ - это $A \to \ldots \to prev(prev(v)) \to prev(v) \to v$

Алгоритм Дейкстры

В отличие от алгоритма Форда - Беллмана требует, чтобы вес $w(e) \geqslant 0$

Алгоритм:

Дан граф $G = (V, E)$, $A \in V$. Найти расстояния до всех вершин $d(u) = dist(A, u)$

$P = \varnothing$

$d(A) = 0$, $d(u \neq A) = +\infty$ ← Обработанные вершины

~~for $\forall v \in V$ (по всем вершинам)~~

Повторяй n раз $(n = |V|)$

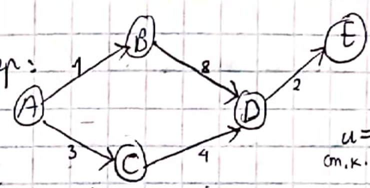
Выбирать $u \in V \setminus P$, где $d(u) \to \min$ (из н обработ. min d)

for $e \in$ ребра из u, $e = (u, v)$

Релаксируем ребро e.

$P = P \cup \{u\}$.

Пример:

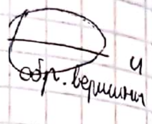


	A	B	C	D	E	
d	$\boxed{0}$	∞	∞	∞	∞	P=∅
u = A (т.к. 0 - min) $A \xrightarrow{1} B$ (0, ∞) $A \xrightarrow{3} B$ (0, ∞)		$\boxed{1}$	3	∞	∞	P={A}
u = B $B \xrightarrow{8} D$ (1, ∞)			$\boxed{3}$	9	∞	P={A,B}
u = C $C \xrightarrow{4} D$ (3, 9)				$\boxed{7}$	∞	P={A,B,C}
u = D $D \xrightarrow{2} E$ (7, ∞)					$\boxed{9}$	P={A,B,C,D}

Эффективность: $|V| \cdot |E| \cdot \log|V|$ ← выбор min

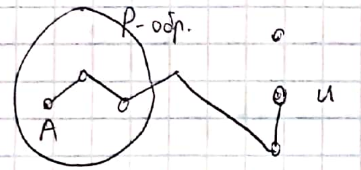
Карректность

Идея: На каждом шаге $d(u) = \min$ путей A



База шаг = 0 $\quad d(A) = 0 \quad d(u) = +\infty$

Переход



Выбрали $u = \min$ вершины из $V \setminus \{P\}$

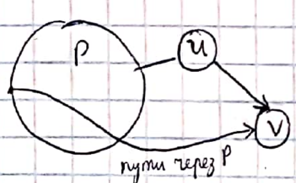
$\exists$ есть опт. путь в u: $\underbrace{A - \ldots - \bar{u}}_{\text{обр.}} \underbrace{- \ldots - u}_{X}$

$$dist(\bar{u}) = dist(u) - x$$

По предположению $dist(\bar{u}) = d(\bar{u})$

$$dist(\bar{u}) < dist(u) \leq d(u)$$

$$\Rightarrow d(u) > d(\bar{u}) \Rightarrow d(u) \text{ был } \min \text{ ??}$$



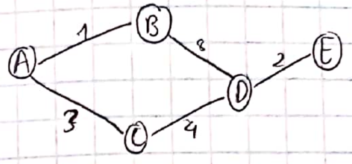
Опт. путь в v идёт через u

$$dist(A,u) + \omega(u,v) = dist(A,v)$$

$\Rightarrow$ Релаксация $u \to v$ успешна и $d(v)$ получит (опт.) расстояние.

Для восстановления пути нужен аналогичный Prev

Успешная релаксация $u \to v \quad Prev[v] = u$



A	B	C	D	E
0	∞	∞	∞	∞
0	1/A	3/A	∞	∞
		3/A	9/B	∞
			7/C	∞
				9/D

Путь $A \to C \to D \to E$

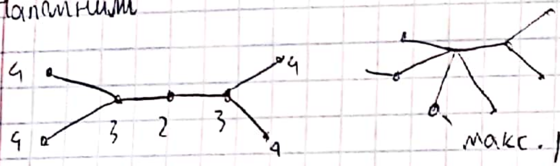
prev(C) prev(D) prev(E)

ЛЕКЦИЯ 16-11

Док.

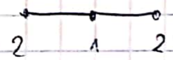
Утв. В дереве: $1 \le$ центров ≤ 2

Напомним

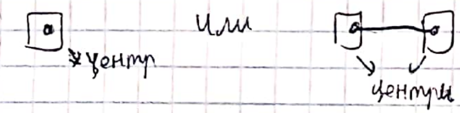


макс. расстояние

Д-во: Если убрать у дерево все висячие вершины, max расстояния уменьшается на 1.



Если повторить убирания висячих ...



Алгоритм Флойда

Дан граф $G = (V, E)$

Вернуть $d(u,v)$ "таблица" расстояний

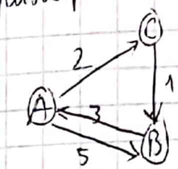
$u \backslash v$	v
u	$d(u,v)$

Алгоритм

$d_0 = d_0(u,v) = 0$

$$d_0(u,v) = \begin{cases} \infty, & \text{если нет ребра } u-v \\ \omega(u,v), & \text{если есть ребро } u-v \end{cases}$$

Пример



d_0:

	A	B	C
A	0	5	2
B	3	0	∞
C	∞	1	0

```
for k ∈ V
    for u ∈ V
        for v ∈ V
            if d(u,v) > d(u,k) + d(k,v)
                d(u,v) = d(u,k) + d(k,v)
```

Пример (выше)

$k = A$

	A	B	C
A	0	5	2
B	3	0	5
C	∞	1	0

$k = B$

	A	B	C
A	0	5	2
B	3	0	5
C	4	1	0

$k = C$

	A	B	C
A	0	3	2
B	3	0	5
C	4	1	0

Корректность

Утв. После шага k в $d(u,v) = \min d(\text{пути})$

$u - \bigcirc - v$ (пути от 1 до k) по индукции по k

Пример (выше)

$k = A$

0	5	2
3	0	5
∞	1	0

$k = B$

0	5	2
3	0	5
4	1	0

$k = C$

0	3	2
3	0	5
4	1	0

База $k=0$

$d(u,v) = \min(u \underset{\text{нет}}{-\bigcirc-} v)$ действительно, ~~в~~начале d содержит длина ребер. по индукции по индукции по k

Переход $k \to k+1$ $\quad u \underset{\text{вершины от 1 до } k+1}{-\bigcirc-} v$

$\exists$ есть опт. путь из $u \underset{1 \quad k+1}{-\bigcirc-} v$

1) $\to$ в нет $k+1$ $\Rightarrow$ его длина $= d(u,v)$

2) $\to$ есть $k+1$: $u - (1 \to k) - (k+1) - (1 \to k) - v$

его длина $d(u,k+1) + d(k+1,v)$

это ровно проверка из цикла ...

$\Rightarrow$ В конце $d(u,v) = \min(u \cdots v) = dist(u,v)$

Замечание

1. Чтобы восстановливать путь, можно ввести through

if $d(u,v) > d(u,k+1) + d(k,v)$

$\Rightarrow d(u,v) =$ "

$throug(u,V) = k$

Для восставлавления пути $A \text{---} \underset{=th(A,B)}{D} \text{---} B$

2. $A \text{--} \underset{=th(A,I)}{J} \text{--} I \text{--} \underset{=th(I,B)}{K} \text{--} B$

3) и т.д. если $th(X,Y)$ нет записи $\Rightarrow$ ребро $X \to Y$ это опт. путь

Замечание: Алг. Флойда ищет: транзитивное замыкание бинарных отношений

$\exists$ R - бин. отношение на M

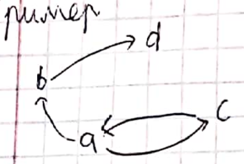
$\exists \bar{R}$ - транзитивное замыкание R, если

1) $\bar{R} \supset R$

2) $\bar{R}$ - транзитивно

3) $\forall \bar{\bar{R}} \quad \bar{R} \supset \bar{\bar{R}} \supset R$

Пример

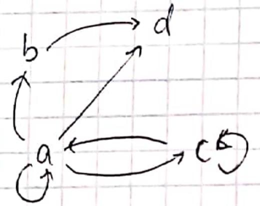


- не транзитивно

aRb, bRd но $\overline{aRd}$

aRc, cRa но $\overline{aRa}$

Сделаем транзитивным



Th: $\exists R$ - бин. отн. на M

$\exists G = (M, R)$ - граф отношения

Тогда $\bar{R}$ - это $x \bar{R} y \Leftrightarrow$ есть путь $x - y$ в G

Применение Флойда к графу $G = (M, R)$

d_0

	y
x	1

$d_0(x,y) = 1$, если xRy

$d_0(x,y) = \infty$, если ~~xRy~~ $x \not R y$

После конца алгоритма

	y
x	∞

замыкание - это $x\bar{R}y$, если $d(x,y) < \infty$

На практике

Алг. Тр. Замыкания

$\bar{R} := R$

for $k \in M$

  for $x \in M$

    for $y \in M$

      if xRk & kRy

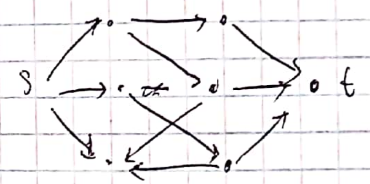
        $\bar{R} \leftarrow (x,y)$ т.е. сделать $x\bar{R}y$

Потоки в сетях.

Опр. Сеть - это $G = (V, E)$ ориентированный

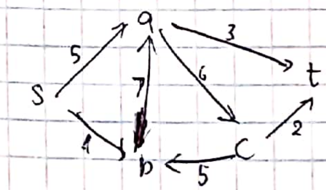
$s \in V$ $\nexists e = (u, s)$ ничего не входит

$t \in V$ $\nexists e = (t, u)$ ничего не выходит



$c: E \to \mathbb{N}$ - пропуск способность ребер целые > 0

Пример:

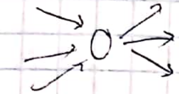


Опр. Поток f в сети G - это $f: E \to R$

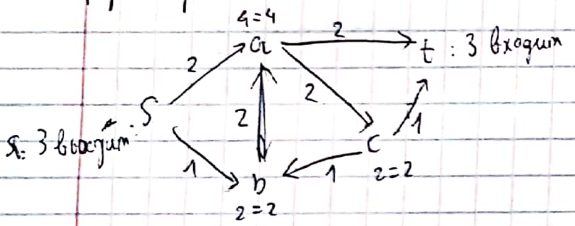
1) $0 \le f(e) \le c(e)$

2) $\forall u \ne s,t$

$$\sum_{e=(v,u)\in E} f(e) + \sum_{e=(u,v)\in E} f(e)$$



Пример

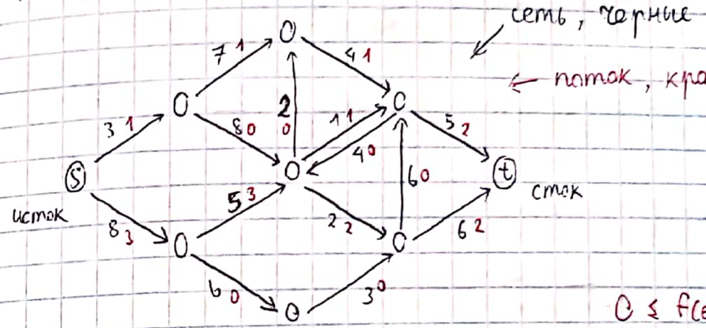


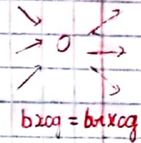
(выход = вход)

ЛЕКЦИЯ 23-11

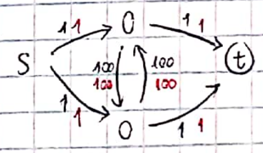
Потоки в сетях

Пример





$0 \le f(e) \le c(e)$

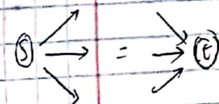


- поток корректный

Величина потока в примере 1 : 4 , в примере 2 : 2

Th: Дана сеть (G, c), поток f на G.

Тогда $\sum\limits_{u:e=(s,u)} f(e) = \sum\limits_{u:e=(u,t)} f(e)$



Рассмотрим $\sum\limits_{e\in E} f(e) = 1) \sum\limits_{v\in V} \sum\limits_{e:e=(u,v)} f(e) \ominus$

$\ominus \underbrace{\sum\limits_{e:e=(u,s)} f(e)}_{0} + \sum\limits_{e:e=(u,t)} f(e) + \sum\limits_{v\in V\setminus\{s;t\}} \sum\limits_{e:e=(u,v)} f(e) =$ вытекает $+$

$+ \sum\limits_{v\in V\setminus\{s,t\}} \sum\limits_{e:e=(u,v)} f(e) =$

$= \text{вытекает} + \sum_{v \in V} \sum_{e: e=(v,u)} f(e) - \sum_{e: e=(s,u)} f(e) - \sum_{e: e=(t,u)} f(e) =$

$= \text{вытекает} + \sum_{v \in V} \sum_{e: e=(v,u)} f(e) - \text{втекает} - 0 =$

$= \text{вытекает} - \text{втекает} + \sum_{e \in E} f(e)$

$\Rightarrow \sum_{e \in E} f(e) = \text{вытекает} - \text{втекает} + \sum_{e \in E} f(e)$

$\Rightarrow \text{вытекает} = \text{втекает}$

$$\sum_{e: e=(u,t)} f(e) = \sum_{e: e=(s,u)} f(e)$$

Опр. Эта величина называется величиной потока (размер) $w(f)$

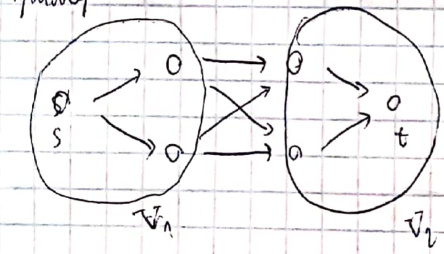
Определение.

Разрез в сеть $(G, c) = (V, E)$.

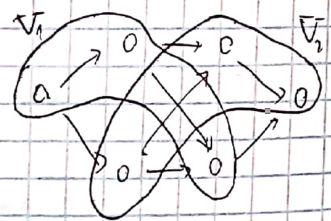
Разрез $C = (V_1, V_2)$ $s \in V_1$; $t \in V_2$ $V_1 \cup V_2 = V$

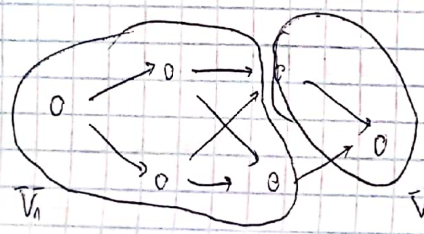
$V_1 \cap V_2 = \varnothing$

Пример



или



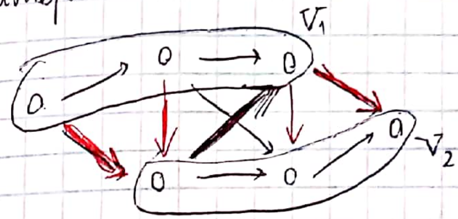


Определение: E_c – рёбра разреза – это все рёбра, которые идут из V_1 в V_2 или наоборот.

E_c^+ – прямые рёбра разреза (из V_1 в V_2)

E_c^- – обратные рёбра разреза (V_2 в V_1)

Пример.



обратное E_c^-

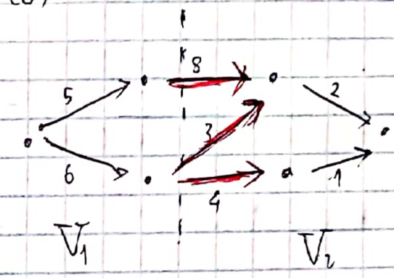
прямое E_c^+

$E_c = E_c^- \cup E_c^+$

Определение: Величина разреза $= \sum_{e \in E_c^+} c(e)$

Обозначение $\mathcal{C}(G)$

Например



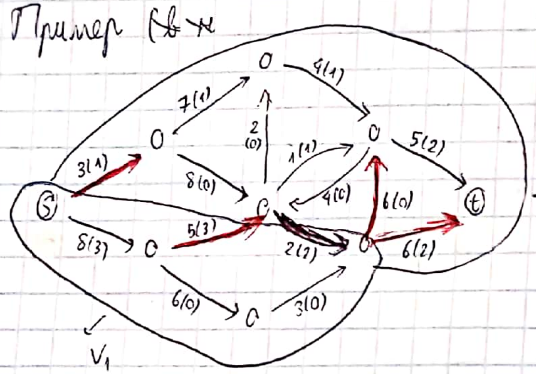
$C = (V_1, V_2)$

$c(C) = 8 + 3 + 4 = 15$

Утв: Пусть есть сеть $(G, c) = (V, E)$, поток f, разрез $C = (V_1, V_2)$

Тогда $w(f) = \sum_{e \in E_c^+} f(e) - \sum_{e \in E_c^-} f(e)$

Пример



$w(f) = 1+3 = 2+2 = 4$

$\sum_{e\in E_c^+} f(e) = 1+3+0+2 = 6$

$\sum_{e\in E_c^-} f(e) = 2$

$4 = 6-2$

Посчитаем сумму $\sum_{v\in V_1}\left(\sum_{e:\, e=(u,v)} f(e) - \sum_{e:\, e=(v,u)} f(e)\right) \ominus$

$\ominus$ 1) для $\forall v \in V\setminus\{s\}$ внутреняя $\sum - \sum = 0$

для $v=s$ получается $w(f) = \sum_{e:\, e=(s,u)} f(e)$

2) $\sum_{\substack{e=(u,v)\\ u\in V_1,\, v\in V_1}}(f(e)-f(e)) + \underbrace{\sum_{e\in E_c^+} f(e) + \sum_{e\in E_c^-}(-f(e))}_{\text{см. условие}} =$

$= 0 +$ величина из условия

Обозначение $w(c,f)$ - величина / размер потока через разрез.

$w(c,f) = \sum_{e\in E_c^+} f(e) - \sum_{e\in E_c^-} f(e)$

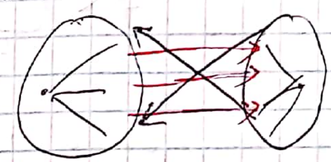
Замечание $\forall C : w(f) = w(c,f)$ - по Теореме.

Замечание: Будем решать задачу о максимальном потоке в сети, т.е. найти $f: w(f)\to max$

Утв. Дана G, c - сеть, C - разрез

Тогда $w(f) \le c(C)$

Д-во



$w(f) = w(C,f) = \sum_{e\in E_c^+} f(e) - \sum_{e\in E_c^-} f(e)$

$\le \sum_{e\in E_c^+} f(e) \le \sum_{e\in E_c^+} c(e) = c(C)$

$\Rightarrow w(f) \le c(C)$

Утв. В сети G $w(f_{max}) \le c(C_{min})$, где $W(f_{max}) = \max_{f\text{-поток}} w(f)$

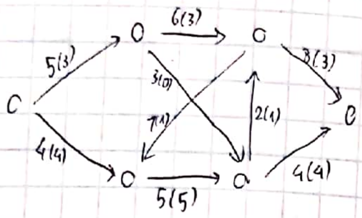
$c(C_{min}) = \min c(C)$, C - разрез.

Th Форда - Фалкерсона: $w(f_{max}) = c(C_{min})$ в сети $(G,c) = (V,E)$, $c(e)\in N$. (для простого считаем, что пропускные способности целые)

Scanned with CamScanner

Опр. Дополнительный граф для потока

$\bar{G}$ имеет $\bar{V} = V$



$\bar{E}$:

если $f(e) < c(e)$, $e = (u, v)$

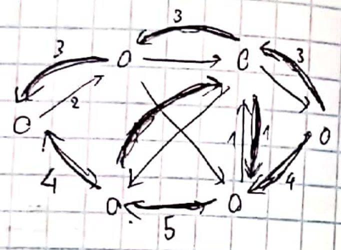
т.е. $e' = (u', v')$

$g(e') = c(e) - f(e)$

если $0 < f(e)$ $e = (u, v)$

т.е. $e'' = (v', u')$

$g(e'') = f(e)$



Д-во: Начнём с первого потока и будем его постоянно увеличивать.

Построим дополнительный граф $\bar{G}$ и найдем в нём путь из s в t $(s) \to 0 \to \ldots \to 0 \to (t)$

Найдем min $g(e)$ на этом пути. Пусть] это x.

Вычтем в доп. графе $-x$ на каждом ребре:

$0 \to 0 \to 0 \to 0$

1) $c(e) - f(e)$ $(f(e) := f(e) + x)$

$\to c(e) - f(e) - x$

$0 \to 0 \xrightarrow[f(e)]{} 0 \to 0$

2) $f(x) - x$ $f(e) := f(e) - x$

Поймём, что 1) новый поток f' остался потоком.

2) величина потока увеличилась на x.

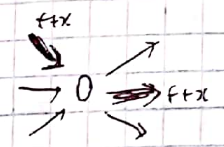
Проверяем, что это поток $0 \le f'(e) \le c(e)$

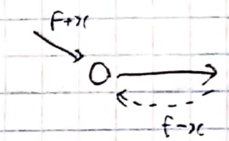
уменьшаем по обратному

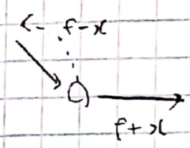
увеличиваем по прямому

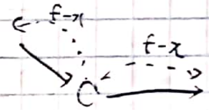
$c(e) - (f(e) + x) \ge 0$

В вершинах верно $\sum вход = \sum исх$

1) 

2) 

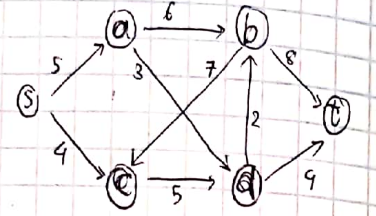
3) 

4) 

Итого f' - поток.

ЛЕКЦИЯ 30-11

Пример: Строить пути



Сначала поток = 0

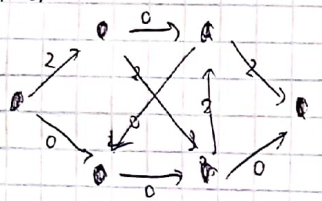
Дополнительный граф

Путь идёт из s в t

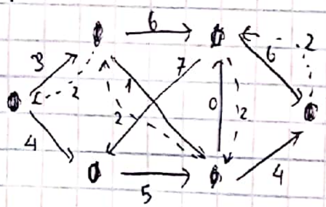
1) $s \underset{5}{-} a \underset{3}{-} d \underset{2}{-} b \underset{8}{-} t$ ← min = 2

добавляем к потоку +2 на каждое из этих рёбер

поток

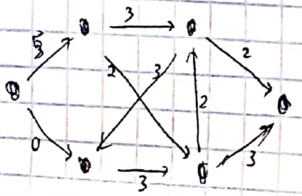


c - f

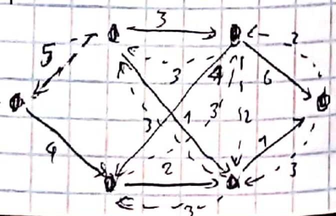


2) путь: $s \underset{3}{-} a \underset{6}{-} b \underset{7}{-} c \underset{5}{-} d \underset{4}{-} t$ min = 3

поток

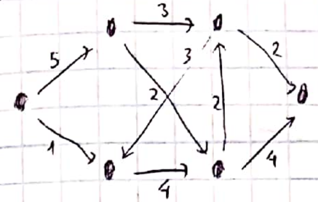


c - f

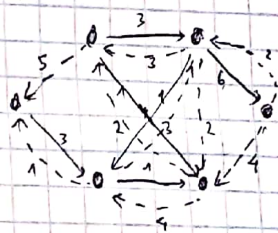


3) $s \underset{4}{-} c \underset{2}{-} d \underset{1}{-} t$ min = 1

поток

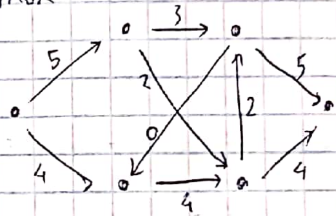


c - f

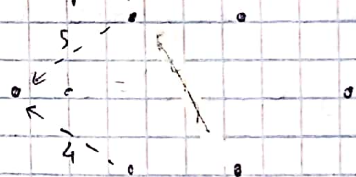


4) $s \underset{3}{-} c \underset{3}{-} b \underset{6}{-} t$ min = 3

поток



Доп. граф



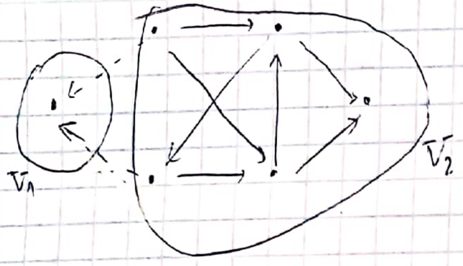
Пути s - t нет

Продолжаем д-во Th Форда - Фалкерсона

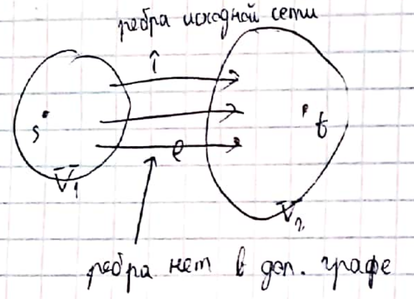
Если пути нет, то поток оптимальный!

∃ V_1 - вершины, достижимые из s по рёбрам доп. графа

$V_2 = V \setminus V_1$



$t \in V_2$ т.к. нет пути $s \to t$. Получим разрез C.



Ребра E_C^+

В доп. графе веса $=0 \Rightarrow c(e) - f(e) = 0$

$$c(C) = \sum_{e \in E_C^+} c(e) = \sum_{e \in E_C^+} f(e) = \sum_{e \in E^+} f(e) - \overbrace{\sum_{e \in E^-} f(e)}^{0} = c(f)$$

Течёт ли что-то назад? Нет, иначе V_1 неверен

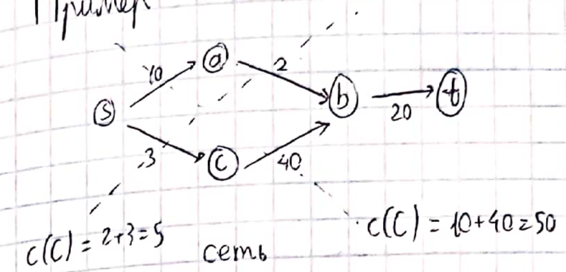
Итого мы нашли разрез C: $c(C) = c(f)$

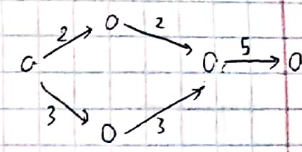
В прошлый раз мы доказали $\forall C$ - разрез и $\forall f$ - поток $c(C) \geq c(f)$

Получается C: min разрез; f - max поток

~~Пример~~

Пример





min разрез $= 5$

max поток $= 5$

Замечание: Метод ФФ строит min разрез и max поток

Утв. Если каждый раз искать путь с min количеством ребер, то время поиска max потока $\sim V^2E$

Без доказательства.

Утв. Для плоской сети (без пересен. ребер) эффективно искать верхние пути

Задача о паросочетаниях

Дан двухдольный граф $G = (U \vee V, E)$

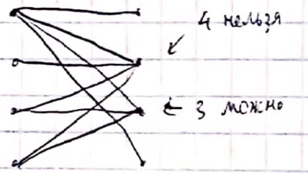
Опр. Паросочетание в G - это $P \subset E$, где ребра из P не имеют общих вершин. Пример

Опр. Максимальное паросочетание, это $P \subset E$

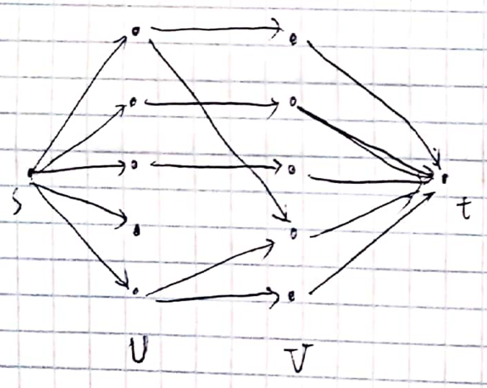
$|P| \to \max$ из возможных

Пример

$P = \{cA, bC, aB\}$



Сводим к задаче о потоке



Ребра из $s \to U$, $V \to t$

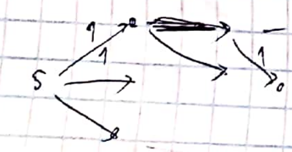
Ребра $U \to V$ слева на право

$C(e) = 1$

Утв. Каждому потоку (из $f = 0, 1$) соотв. паросоч.

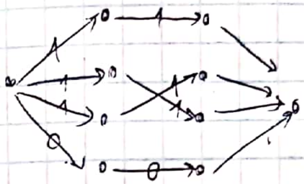
Д-во. Ребра с $f(e) = 1$ - это ребра паросочетания ~~это~~

$\Rightarrow$ поток



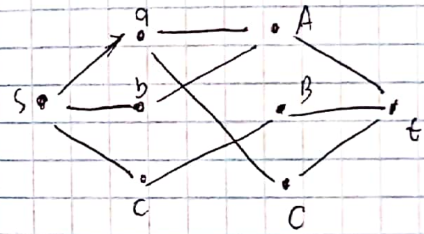
— только одно из ребер = 1

$\Leftarrow$ Паросочетание соответствует поток, где $f(e) = 1$ для ребер паросочетания

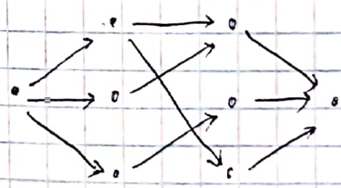


Следствие: Размер max паросочетания = размер max потока

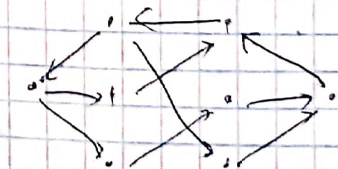
Строим паросочетание методом ФФ



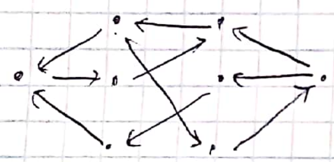
Строим доп. граф но без чисел (всегда 1)



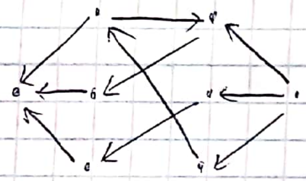
путь s a A t ~~путь~~



путь s - C - B - t



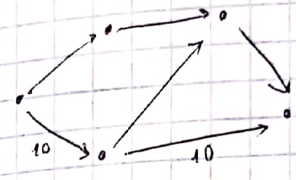
путь s - b - A - a - C - t



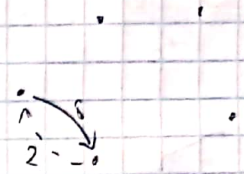
Ответ aC, bA, cB

ЛЕКЦИЯ 07-12

Потоки в сетях

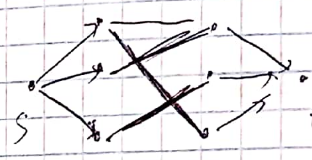


сеть



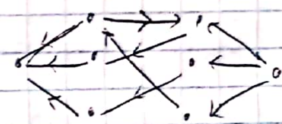
граф

Паросочетания

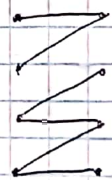


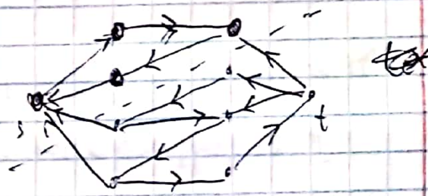
U V

доп. граф



Пример ⊛

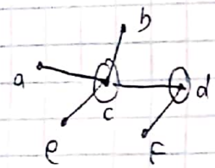




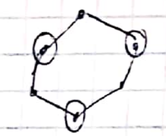
Задача о максимальном контролирующем множестве

Опр. $G = (V, E)$ $C \in V$ - контролирующее множество, если $\forall e \in E$ $\underset{=(u,v)}{}$: $u \in C$ или $v \in C$

Примеры



$C' = \{c, d\}$

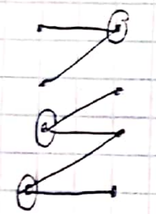


Замечание: $C = V$ - контр. мн-во.

Задача найти КМ min размера.

Будем решать для двудольного графа

В примере ⊗

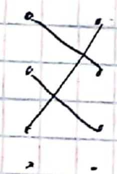


Утв. В двудольном графе $G = (U \cup V, E)$ ∃ C - к. мно-во.

∃ P - паросочетание

Тогда $|C| \geq |P|$

Д-во: P

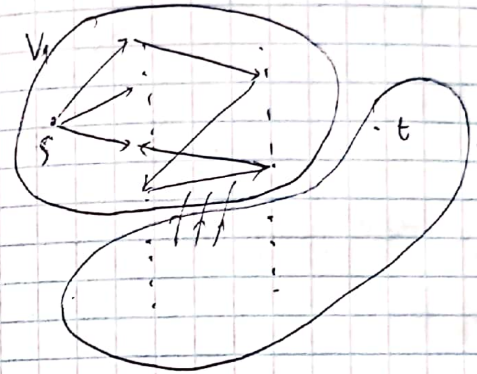


У каждого ребра $e \in P$ есть вершина u или $v \in C$ $\overset{\shortparallel}{(u,v)}$

Утв. ∃ $G = (U \cup V, E)$ - двуд. граф.

Размер макс. паросочетания = размер min контр. мно-ва.

Д-во: Построим макс. паросочетание по Форду-Фалкерсону и ∠ разрез



$|U| = x$ $\quad |U \cap V_1| = a$

$|V| = y$ $\quad |V \cap V_1| = b$

$c[V_1, V_2] = \sum 1$

разрез $\quad$ e - ребро u,v $u \in V_1$ $v \in V_2$

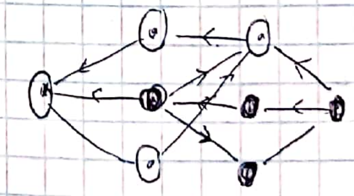
Итого $m = c[V_1, V_2] = x - a + b + \overset{\leq m}{n} \leq x - a + b + m$

$\Rightarrow x - a + b \geq 0$

Возьмём в качестве контр. мно-ва

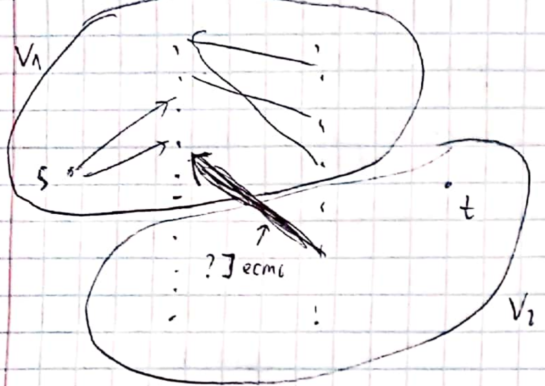
$C = \underset{(U \cap V_2)}{\underset{\shortparallel}{(U \setminus V_1)}} \cap (V \setminus V_1)$ $\quad$ есть ли ребро из $V \cap V_2$ в $U \cup V_1$

Пример



V_1 ○

V_2 ●



контр. мно-во - это

$U \cap V_2$ $\quad$ $V \cap V_1$

∃ есть ребро $e = (u, v)$

1) → значит $v \in V_1$??

2) $u \to v$ как попасть в u только из V

← - Невозможно этом

Значит в min разрезе нет ребер между $U \cap V_2$ и $V \cap V_1$

Вывод 1: C - контр. мно-во
$= (U \cap V_2) \cup (V \cap V_1)$

$s \to u \nrightarrow \to t$

Вывод 2: $C[V_1, V_2] = \sum\limits_{\substack{e \in u,v \\ u \in V_1 \\ v \in V_2}} 1 = (x - a) + 0 + b = |C|$

// разрез

$|P|$

$\Rightarrow |P| = |C|$

<u>Поиск в глубину, ширину</u>

1) структура данных для хранения вершин

D - стек или очередь

$v \to D$ положить v в D

$v = \nless D$ посмотреть

$D \to v$

Стек: первый вошёл, последний вышел.

Очередь: первый вошёл, первый вышел.

Пример:	Стек	Очередь
$a \to D$	a	a
$b \to D$	ba	ba
$c \to D$	cba	cba
$\nless D$ →	c	a
$D \to$	ba	cb
$\nless D$ →	b	b

Поиск в ширину (D очередь) или глубину (D стек)

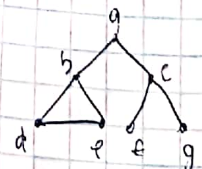
$D \leftarrow v_0$ начальная вершина $\quad$ Были$(v_0) = 1$

пока D не пуст $\quad u = \nless D$

если его ребро Было(v) тогда $v \to D$ $\quad$ Были$(v) = 1$

иначе достать $D \to u$

Пример



В глубину

a	ca
ba	fca
dba	ca
edba	gca
dba	ca
ba	a
a	

В ширину

a	fedc
ba	gfedc
cba	gfed
cb	gfe
dcb	gf
edcb	g
edc	=

ЛЕКЦИЯ 21-12

Поиск в глубину / ширину

D-стек   D-очередь

Алгоритм поиска

Дана начальная вершина u

$D \leftarrow u$ Used $\leftarrow \varnothing$ (обработанные, то, что было с D)

пока $D \neq \varnothing$

$u \leftarrow$ peek D (смотрим)

Если есть ребро v - w, где $w \notin$ used : ~~D←w~~

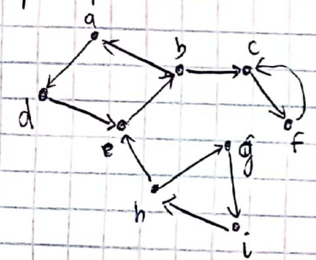
$D \leftarrow w$; Used = used $\cup \{w\}$

иначе ~~D~~ $\leftarrow$ D (убираем вершину из D)

Введем: h(u) - номер, какой попала в D

b(u) - обратный номер, какой ушла из D

Пример

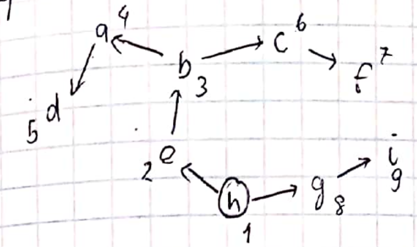


В глубину из h

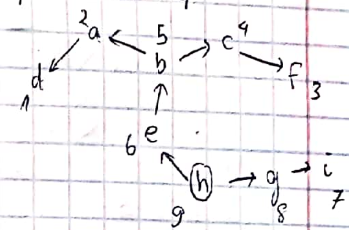
Стек

h	h
he	hg
heb	~~hgi~~
heba	∅
heb~~ad~~	
heb	
hebc	
heb~~cf~~	

Дерево поиска



обратный номер

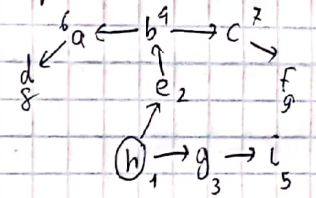


В ширину из h

Очередь

h	gb	acd
he	gbi	cd
heg	bi	cdf
eg	bia	∅
~~eg~~b	~~b~~iac	
	ac	

Дерево поиска

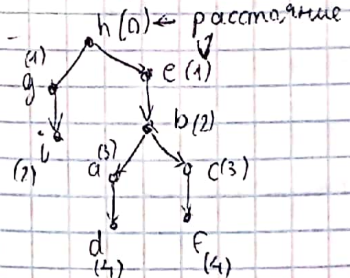


Замечание: При поиске в ширину h(u) = b(u)

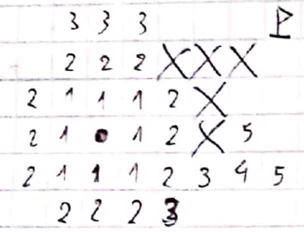
Утв. Поиск в глубину перебирает вершины в том же вершине

Алгоритм Дейкстры (веса ребер = 1)

Действительно, добавление вершины в D - это релаксация ребер v - w. Удаление из D - убирание вершины с min расстоянием



Пример задачи



Полный поиск в глубину

Пока есть непосещённая вершина u

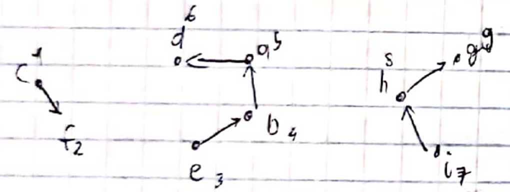
Поиск в глубину (u)

Пример:

dfs (c)

dfs (e)

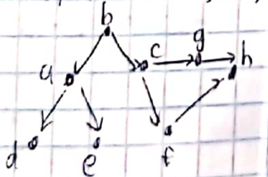
dfs (i)



Утв. Пусть G - ор. граф без циклов

Пусть есть путь $u \to v$ (нет пути $v \to u$, т.к. нет циклов)

Тогда после полного dfs $b(u) > b(v)$



Д-во: Делаем dfs. Куда попали раньше?

1) сначала попали в u

$u \to \cdot \to \cdot \to \cdot \to v$

В стеке будет: $u \dots v$

$\Rightarrow$ Сначала из стека уйдёт v, потом u.

2) Сначала v $\Rightarrow$ Закончим просмотр, не попав в u

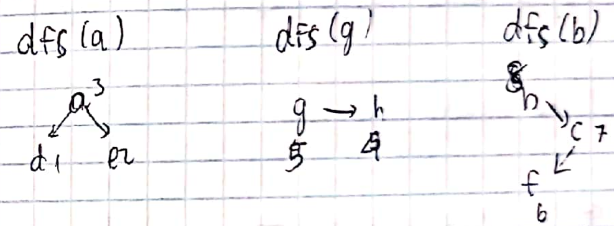
$u \to \cdot \to \cdot \to v$

$\Rightarrow$ Номер $b(v)$ присвоится раньше чем $b(u)$

Следствие: Алгоритм топ. сортировки

Делаем полный dfs и мин. порядок задаём как $b(u)$

Пример:

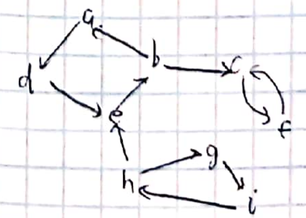


Ответ: d e a h g f c b

Компоненты сильной связности

Напоминание: $G = (V, E)$ - ор. граф. Введём отношение $\leftrightarrow$ на V.

$u \leftrightarrow v :=$ если есть путь $u \to v$ и $v \to u$



$c \leftrightarrow f$ $\quad$ $i \leftrightarrow h$

$a \leftrightarrow e$ $\quad$ $b \leftrightarrow c$

$\{abde\}$ $\{cf\}$ $\{ghi\}$

3 комп. сильной связности

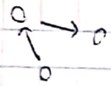
Напомним $\leftrightarrow$ отн. экви-ти.

Классы эквивалентности называются компонентами сильной связности.

Опр. $\exists$ $G=(V,E)$ – ор. граф

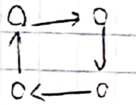
$G^c = (V^c, E^c)$ – граф , если

$V^c = V/\leftrightarrow$ (классы экви-ти)

В примере 

E^c u^c в v^c есть ребро, если $\exists\, e=(u,v)$, где $u \in U^c$, $v \in V_0$

Замечание: G^c не имеет циклов

 $\Leftarrow \Rightarrow$ У вершины $\leftrightarrow$

Утв. $\exists$ $G = (\bar{V}, E)$ – ор. граф

G^c – граф G

Делаем полный dfs в G

Тогда если в G^c есть путь из u^c в v^c, то ~~b(u) > b(v)~~ $\max b(u) > \max b(v)$,

для любых $u \in U_0$, $v \in V_0$

Д-во: Аналогично прошлому утв.

Следствие: Поиск компонента сильной связности

1. Полный dfs в G

2. Находим u: $\max b(u)$. Делаем dfs по обратным ребрам G.