Бинарное отношение

Опр. M - множество $\neq \varnothing$
(не пустое)

$R \subset M \times M$ - бинарное отношение
(подмножество)

Пояснение

$M \times M$ - мн-во пар из элементов R

Допустим $M = \{a, b, c\}$

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

или $M = \mathbb{N}$ (мн-во нат. чисел)

$$M \times M = \mathbb{N} \times \mathbb{N} = \{(1,1), (1,2), (2,1), (1,3), (2,2), (3,1), (1,4), (2,3) \dots (42,15) \dots\}$$

R - отношение - это подмножество пар

Обозначение

$\underbrace{(x, y) \in R}$ - пара (x, y) принадлежит отношению

иное обозначение xRy

$(x, y) \notin R \longrightarrow \cancel{xRy}$

Примеры

(пары (x, y) такие что $x > y$)

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

$(3,2) \in R \qquad \cancel{3R4} \qquad 3 > 2$

$(3,4) \notin R \qquad 3R2 \qquad \cancel{3 > 4}$

– отношение больше

2) $M = \mathbb{R}$ отношение $\geq$ $\qquad 7 \geq 6 \quad 7 \geq 7$

$\cancel{7 \geq 8}$

3) $M = \mathbb{R}$ отношение $=$ $\qquad 7 = 7 \quad \cancel{7 = 8}$

$(7,7) \in \, =$

$(7,8) \notin \, =$

4) $M = \mathbb{R}$ $\quad \approx \quad$ $(x,y) \in \, \approx$

$x \approx y \iff |x - y| = 1$

5) $M = \mathbb{R}$ $\quad \#$ $\qquad x \# y \iff x^2 > y$

$2 \# 2$, т.к. $2^2 > 2$

$\cancel{1 \# 2}$

6) $M = \mathbb{N}$ или $M = \mathbb{Z}$ $\quad \vdots$ $\qquad x \vdots y \iff \exists k \in \mathbb{Z} : x = ky$

$4 \vdots 2 \qquad 7 \vdots 0$

$\cancel{2 \vdots 4} \qquad 0 \vdots 0$ – для целых

7) $M = \mathbb{Z}$ $\quad \underset{3}{\equiv}$ $\qquad 0 \underset{3}{\equiv} 3 \qquad 1 \underset{3}{\equiv} 4 \qquad \cancel{1 \underset{3}{\equiv} 8}$

$\cancel{0 \underset{3}{\equiv} 2} \qquad 1 \underset{3}{\equiv} 7$

8) $M = \mathbb{N}$

$a \; \text{ц} \; b$, если в числе a "b" цифр

$100 \; \text{ц} \; 3$

$238 \; \text{ц} \; 3$

$\cancel{238 \; \text{ц} \; 8}$

9) $M =$ прямые на $\mathbb{R}^2$ (на плоскости)

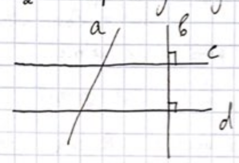
$\parallel$ – $l_1 \parallel l_2$, если l_1 не пересекает l_2 или $l_1 = l_2$

10) $\perp \qquad l_1 \perp l_2$ перпендикулярны

$c \parallel d \qquad b \perp c$

$a \parallel a \qquad \cancel{b \perp a}$

$\cancel{b \parallel a}$



11) $M =$ студенты ЛЭТИ

$x \succ y$ — средний балл за последнюю сессию больше у "x" чем у "y"

12) $M =$ пользователи Одноклассники

$x \to y$, если "y" в друзьях у "x"

Свойства бинарных отношений

Опр. Б. отношение R называют рефлексивным, если $\underline{\forall x \in M \quad x R x}$ $((x, x) \in R)$

Замечание Отношение не рефлексивно $\Longleftrightarrow$ $\exists x \quad x \not R x$ – контрпример

Примеры

$=$ – рефлексивно $\forall x : x = x$

$\geqslant$ – рефлексивно $\forall x : x \geqslant x$

$\approx$ – рефлексивно $\forall x : x \approx x$, т.к. $|x - x| = 0 < 1$

$\vdots$ – рефлексивно $\forall x : x \vdots x$

$>$ – не рефлексивно $2 \not> 2$

Цу – не рефлексивно $3 \cancel{Цу} 3$ (хотя бы один контрпример)

$\to$ – не рефл.

$\perp$ – не рефл.

Опр. Б. отношение R на мн-ве M называется антирефлексивным, если $\forall x \quad x R x$

Замечание R – не анти рефл. $\Longleftrightarrow$ $\exists x : x R x$ (контрпример)

Примеры

$>$ – антирефл. $x \not> x$

$\perp$ – антирефл. $l \not\perp l$

$\to$ – антирефл. {нельзя быть в друзьях у себя

Цу – не антирефл. контрпример: 1 Цу 1

Замечание

1) Цу – не рефл.
   не антирефл.

2) не бывает R, которое и рефл., и антирефл.

(рассмотрим $a \in M$ ↗ $a R a \Rightarrow$ не ар.
↘ $a \not R a \Rightarrow$ не р.

Опр. (св-во б. отн.) Б. отн. R на мн-ве M симметрично, если $\forall x, y \quad x R y \Longleftrightarrow y R x$

Замечание R – не симметрично, $\Longleftrightarrow$

$\exists x, y : x R y$, ~~[illegible]~~ $y \not R x$ контрпример

Примеры

$=$ – симм. $x = y \Longleftrightarrow y = x$

$\approx$ – симм. $x \approx y$
$|x - y| < 1$, $|y - x| < 1$

$\vdots$ – не симм. $4 \vdots 2$ $2 \not\vdots 4$

$\parallel$, $\perp$ – симметр.

Опр. Б. отн. R на мн-ве M антисимм., если $\forall x \neq y$! $xRy \Rightarrow y \not R x$

Замечание R - не антисимм., если $\Leftrightarrow \exists x \neq y$ xRy, yRx - контрпример

Примеры $>$: $x \neq y$, $x > y \Rightarrow y \not> x$ - антисимм.

Попробуем прстроить контрпример

$x \neq y$, $x > y$, $y > x$ - невозможно

$\Rightarrow$ нет контрпримера $\Rightarrow$ антисимм.

$\geqslant$ - антисимм.

$x \neq y$, $x \geqslant y$, $y \geqslant x$ - невозможно

$\Rightarrow$ нет контрпримера

$\Rightarrow$ антисимм.

$x \neq y$, $x = y$, $y = x$ - невозможно

$\Rightarrow$ нет контрпримера $\Rightarrow$ антисимметр.

$\underset{3}{\equiv}$ - не антисимм.

$1 \underset{3}{\equiv} 4$ $\quad 4 \underset{3}{\equiv} 1$ $\quad 1 \neq 4$ контрпример

$\vdots$ над $\mathbb{N}$ - антисимметр.

$x \neq y$ $\quad x \vdots y$ $\quad y \vdots x$ - невозможно

$\vdots$ над $\mathbb{Z}$ - не антисимм.

$4 \neq -4$ $\quad 4 \vdots -4$ $\quad -4 \vdots 4$

Лекция 2. 21.09.

Антисимметричность

$\vdots$ на $\mathbb{Z}$ - не антисимм.

$-2 \vdots 2$

$2 \vdots -2$

$2 \neq -2$ - контрпример

$\vdots$ на N - антисимм.

$\left\{ x \neq y \quad \begin{matrix} x \vdots y \\ y \vdots x \end{matrix} \right.$ - не возм. $\}$ контрпример

$x \neq y$ $\quad x \vdots y \Rightarrow y \not\vdots\, x$

~~анти~~симм.

Опр.

R - бин. отн-е асимметрично, если $\forall x, y$ $xRy \Rightarrow y \not R x$ ($x \neq y$ антис.)

контрпример : xRy, yRx

Утв. R - асимметр. $\Leftrightarrow$ R - антисимм. и асимметр.

Пример

$>$ - асимметрично

$\forall x, y$; $x > y \Rightarrow y \not> x$

$\square \to$ нет пар
$\square$ – асим. (пустое – когда R – пустое мн-во)

– "выше" – асимметрично на мн-ве людей

– "начальник" на мн-ве тех, кто работает в университете

x над $y \Rightarrow y$ ~~над~~ x

Транзитивность

R – бин. отношение транзитивно, если

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

контрпример: $xRy,\ yRz,\ x\cancel{R}z$

Пример: $>: x > y,\ y > z \Rightarrow x > z$ – транз.

$\geq$ – транз.

$\vdots$ – транз.

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

$x = ky,\ y = lz \Rightarrow x = (kl)z \Rightarrow x \vdots z$

контрпример:

$\perp$ – не тр.

$x \perp y,\ y \perp z \quad x \cancel{\perp} z$

Ц$_у$ (кол-во цифр)

100 Ц$_у$ $3 \qquad 3$ Ц$_у$ 1

100 ~~Ц$_у$~~ 1 – не транзитивно

Опр. отношение R называется, является отношением эквивалентности, если R – рефл., симм., транз.

Замечание

Учебник Поздняковa (не конст) отн. эквив.

Пример: 1) $=$ на $\mathbb{R}$ (или $\forall$ другое мн.)

$\forall x: x = x$ – рефл.

$\forall x, y: x = y,\ y = x$ – симм.

$\forall x, y, z: x = y,\ y = z \Rightarrow x = z$ – транз.

$=$ – это отношение эквивалентности

2) $\parallel$ параллельность
3) $\equiv_3$
} ОЭ

$\geq$ – не ОЭ, т.к. не симм.

$x \geq y \Rightarrow y \cancel{\geq} x \Rightarrow 2 \geq 1,\ 1 \cancel{\geq} 2$

не ОЭ (не транз.)

$\approx$ – не ОЭ (не транз.)

отношение $\uparrow$ на $\mathbb{N}$

$x \uparrow y$, если у x и y поровну цифр

$2 \uparrow 5$

$12 \uparrow 45$ $\quad 33 \cancel{\uparrow} 100$

$\uparrow$ ОЭ $\quad x \uparrow x$ – рефл.

$x \uparrow y \Rightarrow y \uparrow x$ – симм.

$x \uparrow y,\ y \uparrow z \Rightarrow x \uparrow z$ – транз.

$=,\ \parallel,\ \underset{3}{\equiv},\ \uparrow$ – ОЭ

одинаковое кол-во цифр

один остаток

одно напр.

Опр. R – отношение эквив. на мн-ве M, $x \in M$, класс элемента x

$M_x = \{y \mid xRy\}$

Примеры: $= \quad M_5 = \{5\}$

$\underset{3}{\equiv} \quad M_2 = \{2, 5, 8, 11, \dots\}$ $\quad M_5$

Утверждение

$\forall x, y \in M \quad R$ – ОЭ на M

$M_x = M_y$ или $M_x \cap M_y \ \varnothing$

Доказательство:

$\exists\ M_x \cap M_y \neq \varnothing \Rightarrow$

$\exists z \in M_x;\ z \in M_y \Rightarrow xRz$

yRz

$\underset{\text{симм.}}{\Rightarrow} \quad zRy \underset{\text{транз.}}{\Rightarrow} \boxed{xRy}$

– Теперь проверим, что класс $M_x = M_y$

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

$u \in M_x \Rightarrow xRu$

$xRy \Rightarrow yRx \Rightarrow yRu \Rightarrow u \in M_y$. ч.т.д.

Следствие. R – ОЭ на M, тогда M разбито на несколько классов эквивалентности / классов элементов.

$M = M \quad$ и $\quad M_n$

$M_i \cap M_j \neq \varnothing$

$=$ на $\mathbb{N}$

$N = \{1\} \cup \{2\} \cup \{3\}$

$\underset{3}{\equiv}$ на $N \qquad N = \{0, 3, 6, 9 \dots\}$

$\{1, 4, 7, 10, \dots\}$

$\{2, 5, 8, 11, \dots\}$

Замечание

Если есть $M \neq \emptyset$ разбитое на $M_i \neq \emptyset$

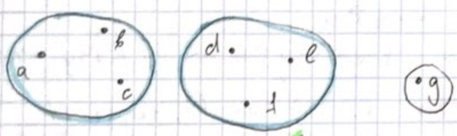
$M = M_1 \cup \ldots \cup M_n$

$M_i \cap M_j = \emptyset$

Тогда можно ввести отношение R

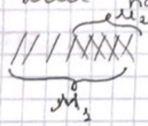
xRy если $\exists M_i : x, y \in M_i$

a b c d e f g



$aRb \quad bRc \quad a\cancel{R}d \quad gRg \quad g\cancel{R}a$

– Для парал. // классы эквивалентности



M_i – направление

Отношение порядка

(выше, лучше, сильнее, быстрее, важнее)

Определение 1.

Предпорядок.

R – бин. отношение

R – Пп, если R-транзитивно

Определение

R-бинарное отношение

$\rceil R$ – транзитивно, антисимметр.

1) рефлексивно – нестрогий порядок

2) антирефлексивно – строгий порядок

– обозначение обычно $>$ строгий

$\geqslant$ нестрогий

Обозначение

тр. $a > b \quad b > c \Rightarrow a > c$

антисимм. $a > b \quad b \not> a$

Примеры: $>$ на $\mathbb{R}$ – строгий порядок

$\geqslant$ на $\mathbb{R}$ – нестрогий порядок

$\vdots$ на $\mathbb{N}$ – нестрогий порядок

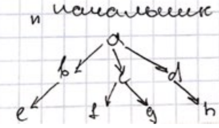
$a \vdots a$ – рефлексивн. (число всегда делится само на себя)

„начальник"

a нач. b

a нач. c – строг.

c нач. f



Определение

$\rceil R$ - строгий или нестрогий порядок

R - линейный, если $\forall x \neq y$

xRy или yRx

R - частичный, иначе

$(\exists x \neq y)$, $(x \not R y)$, $(y \not R x)$

Примеры $>$, $\geqslant$ - лин. порядок

$\vdots$ - частичн. порядок

$2 \not\vdots 3$ $\quad$ $3 \not\vdots 2$

нач. - частичный $\quad$ (в, с)

несравн.

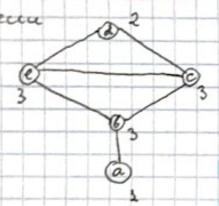
Утверждение

R - порядок (строгий или нестрогий) на M - конечн. $|M| < \infty$

Тогда $\exists x$ - мин., т.е. $\forall y$: $x \not> y$

Напоминание. Графы

степени



1) a b c d

$(a;b)$ путь, $(b;c)$ путь

2) a b

3) a b a

4) a b c d e c d

— различные пути

Опр. Путь в графе $G=(V;E)$ — последовательность

$V_1 e_1 V_2 e_2 \dots V_{n-1} e_{n-1} V_n$

$V_i \in V$ (вершины) $e_i \in E$ (ребра) $e_i = (V_i ; V_{i+1})$

Опр

Замкнутый путь — если $(V_1 = V_n)$

Не замкнутый путь (открытый) $(V_1 \neq V_n)$

Опр. Простой путь, если $e_i \neq e_j$, при $i \neq j$ (нет одинаковых рёбер)

Пример: becde — путь прост, но не замкнут

Опр.

Пути		Все ребра разные	Все вершины разные
замкн.	замкн. путь	пр. замкн. путь	цикл
открыт.	откр. путь	пр. откр. путь	цепь

Путь (повторяется ребро)

Простой путь (не повтор. ребра)

Цепь

Цикл

Th Если $\exists$ путь между вершинами u, v => есть цепь от u до v

Д-во Пусть путь $\underline{u}\, e_1 v_1 e_2 u_2 \dots e_n \underline{v_n}$ — рассмотрим все пути из этих рёбер и выберем min. Это будет цепь

иначе:

$$u \dots v_i \dots v_i \dots v$$

$$\exists u_i = v_j$$

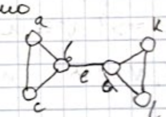
укоротим $u \dots v_i = v_j \dots v$?? противоречие

Th Если есть простой замкн. путь через ребро e => есть цикл через e

Д-во Аналогично

Замечание

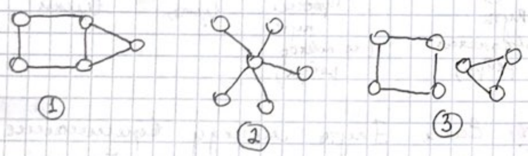
— цикла через e нет



dbacbd — не простой путь (e повторяется)

Связность графа

Опр. $G \overset{(V,E)}{-}$ связен, если $\forall u, v \in V$ $\exists$ цепь (путь) из v в V.

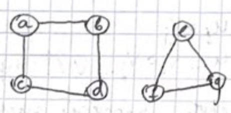


1, 2 – связные
3 – не связные

Введем отн. $\equiv$ (эквивал.) на вершинах графа:

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

Пример

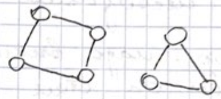


$a \equiv c$
$a \equiv d$
$a \cancel{\equiv} e$ – не верно

Проверим, что $\equiv$ – это отношение эквив.

1) рефл. $u \equiv u$ – верно путь u
2) если $u \equiv V \Rightarrow ? \; V \equiv u$ путь $u e_1 V_1 \dots V$
$V \dots V_1 e_1 u$
3) транз. $u \equiv V$, $V \equiv w$ путь $\underbrace{u e_1 \dots V \; \dots \; w}_{\text{путь } u - w}$

Опр. классы эквивалентности $\equiv$ это "компоненты связности"



– квадрат и треугольник по отдельности являются компонентами связности
– две компоненты связности

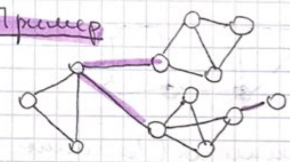
Опр. $\underset{(V_1, E_1)}{G_1}$ – подграф $\underset{(V, E)}{G}$ если $V_1 \subset V$
$E_1 \subset E$

Пример

Замечание G – свой подграф
$\emptyset, \emptyset$ – подграф чего угодно

Опр. $G = (V, E)$ – Ребро e называется мостом если количество компонент связности G < кол-ва комп. связн. $(V, E \setminus \{e\})$

Пример



— мосты

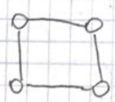
– односвязный граф

Опр. степень связности графа G — это min кол-во рёбер, которые надо выкинуть, чтоб G стал несвязным

Опр. двухсвязный граф — надо выкинуть хотя бы 2 ребра, чтобы он стал несвязным

Замечание двусвязный $\Longleftrightarrow$ нет мостов

Пример

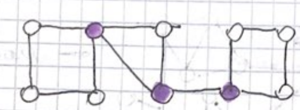


— двусвязный граф

Опр. вершина $v \in V$ называется точкой сочленения, если кол-во компонент связности

$G <$ кол-во комп. св-ти $G = (V \setminus \{u\}, E \setminus \{(v,u) \mid (v,u) \in E\})$

Пример



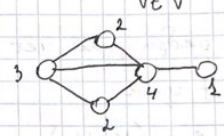
● — точки сочленения (вершины моста)

Считаем рёбра, вершины

Th В графе $G = (V, E)$

если $\deg(u)$ — степень вершины u

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



рёбер:

$$6 = \frac{1}{2}(3+2+2+4+1)$$

— верно

Д-во $\deg(v)$ = кол-во рёбер, выходящих из вершины

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

Следствие 1) сумма степеней вершин всегда чётна

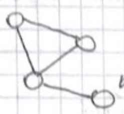
2) вершин нечётной степени — чётно

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

Решение (нет, т.к. это граф из 15 (неч.) вершин степ 3 (неч.))

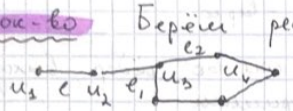
Опр. Висячая вершина – это вершина степени 1

Пример



Th Если в графе есть ребра, но нет висячих вершин, то $\exists$ цикл

Док-во Берём ребро $e = (u_1 u_2)$



u_2 – не висячая, $\Rightarrow$ из неё есть еще ребра $(u_2 u_3)$

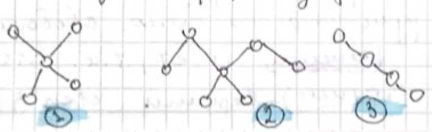
u_3 – ——//—— ——//—— $e_2 (u_3 u_4)$

продолжаем, пока очередной u_n не будет равен u_i $1 \le i \le n$

Путь $u_i, u_{i+1}, \dots u_n$ – цикл $\left(\begin{matrix}\text{ребра разные} \\ \text{верш. разные}\end{matrix}\right)$

Опр. Дерево – связный граф без циклов

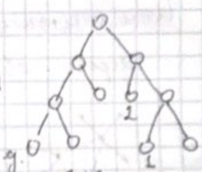
Примеры



Th В $\forall$ дереве ≥ 2 висячих вершин

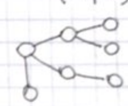
Док-во Берем $\forall$ вершину, если она не висячая, идём по ребру, если опять не висячая, есть ещё ребро и т.д. циклов нет $\Rightarrow$ будет конец это в/верш

Чтобы найти вторую, надо начать из первой



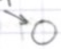
Th Если G – дерево (V, E), то $|V| = 1 + |E|$

Пример



7 вершин
6 рёбер

Д-во по индукции (кол-во вершин)

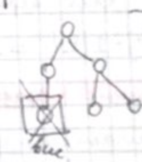
Б. $|V| = 1$ граф  $|E| = 0$ $|V| = \underset{0}{|E|} + 1$

П. $\exists$ $|V| = n+1$

Найдем висячую вершину $\Big\}$ кол-во рёбер

удалим её вершину

$G' = (V \setminus \{v\}, E \setminus \{e\})$ – тоже дерево, т.к. связен, нет циклов



$\Rightarrow |V'| = 1 + |E'|$

$\Rightarrow |V| = 1 + |E|$

Лекция 5. 05.10.21

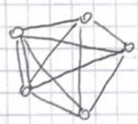
Напоминание.

дерево – связный граф без циклов

$|E| + 1 = |V|$

$|E|$ – кол-во рёбер, $|V|$ – кол-во вершин

] G – полный граф, $\forall u \neq v \subset V$ соединены ребром



– если n вершин ($|V| = n$), то рёбер

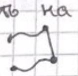
1) C_n^2 рёбер, выбираем пары $= \frac{n(n-1)}{2}$

2) степени всех вершин $n-1$

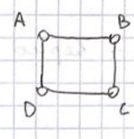
$\sum \deg(v) = 2|E| \Rightarrow n(n-1) = 2|E|$

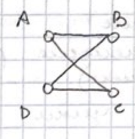
Планарные графы

Опр. G – планарный граф – если можно нарисовать на плоскости так, чтобы рёбра не пересекались



ребро $[0,1] \to \mathbb{R}^2$



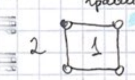


– планарный, но неправильно нарисован.

Ф-ла Эйлера

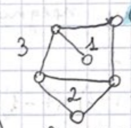
Если планарный граф $G = (V, E)$ нарисован на плоскости, у него можно посчитать грани,
] их f (грани), $|V| = n$, $|E| = m$

1



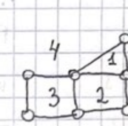
2 – грани (внутр и внешн.)

2



3 грани

3



Тогда: $n - m + f = 2$

Проверим:

1. $4 - 4 + 2 = 2$
2. $6 - 7 + 3 = 2$
3. $7 - 9 + 4 = 2$

Док-во:

Индукция по кол-ву рёбер

<u>База</u> G – дерево

• у него всегда одна грань

– дерево без циклов | – вокруг грани всегда есть цикл

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

Переход – G – не знаем,
– если G' имеет меньше рёбер = верно
(G, G') – пл. графы связные)
G – не дерево => есть цикл

– вокруг него 2 грани
– удалим ребро, получим G' – тоже связен и планарен

n' m' f' – вершины, рёбра, грани G

$n' = n$

$m' = m - 1$

$f' = f - 1$

По индукц. предпол.

$n' - m' + f' = 2$

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

$\Rightarrow n - m + f = 2$

Следствия

1. неважно, как рисовать планарный граф, кол-во граней постоянно

2. про многогранник так же

$8 - 12 + 6 = 2$

3. если граф планарный (не обяз. связен), то $n - m + f = 1 + $ |компон. связн. G)

Док-во: упр.

4.] У каждой грани вокруг ≥ 3 ребра

$\sum\limits_{y \in \text{грань}}$ кол-во ребер вокруг $y \geq 3f \leq 2m$ – вывод

каждое ребро посчитано 2 раза

но $n - m + f = 2$ $\quad * 3$

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

$\Rightarrow 3n - m \geq 6 \Rightarrow \boxed{m \leq 3n - 6}$

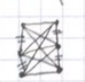
Следствие: Полный граф при $n = 5$ – не планар.

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

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

Замечание K_5 – полный граф $n = 5$

Утв. Граф $K_{3,3}$ – тоже не планарный



Д-во: $n = 6$ $\quad m = 9$

$9 \leq 3 \cdot 6 - 6$ ✓

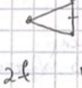
нет противоречий

– Сколько граней, если планарный

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

– в $K_{3,3}$ все циклы чётные (ходим лево-право, или право-лево)

$\Rightarrow$ у грани ≥ 4 ребра



невозможно

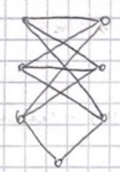
$4f \leq \sum\limits_{y \in \text{грани}}$ ребра грани $y \leq 2m$

$\Rightarrow m \geq 2f$ $\quad$ но $9 \geq 2 \cdot 5$ – не верно

Th. Теорема Понтрягина - Куратовского

— ~~[illegible]~~ граф G - планарен $\Leftrightarrow$ если не содержит подграфов, стягивающихся к K_5 и $K_{3,3}$

Пример.



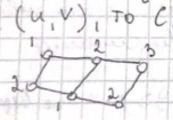
— стягивается к $K_{3,3}$ => не планарен

— не планарен, в нем есть K_5

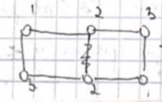
Хроматизм

Опр. $\rceil$ $G = (V, E)$ граф

раскраска графа G в k цветов это ф-ия $C: V \to \{1..k\}$ (целые), причем если есть рёбра (u, v), то $C(u) \neq C(v)$

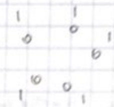


— раскр. в 3 цвета



— не раскр.

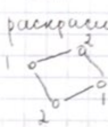
— Какие графы можно раскрасить в один цвет?



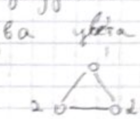
— это граф без рёбер

— Какие графы можно раскрасить в 2 цвета?

Опр. Граф G двудолен если его можно раскрасить в два цвета

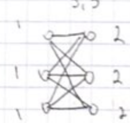


— двудольный



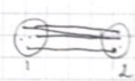
— недвудольный

$K_{3,3}$ — двудольный



Замечание

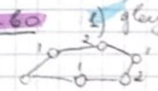
— Двудольные графы часто рисуют из двух частей (доль)



— пример

Th. G — двудолен только тогда, когда все его циклы имеют чётную длину

Док-во 1) двудолен => циклы чётные



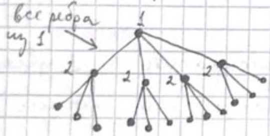
— циклы

2) циклы чётные => ? двудолен

"подвесим граф за вершину"

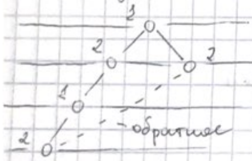
∀ вершина

• 1 - цвет



-рисуем рёбра, которые не идут назад

-назначаем цвета по уровням

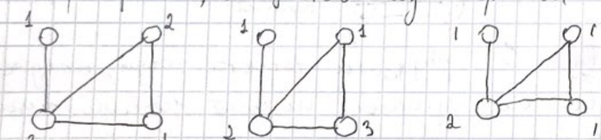


- почему обратное ребро не соединяет одинак. цвета?

-потому что иначе циклы нечетный

Лекция 6. Напоминание

k - раскраска, k - цветов из вершин



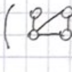
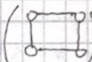
разные графы

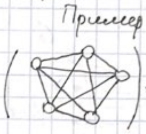
- не раскраска

Опр. $G=(V,E)$ - граф

$\chi(G)$ - хроматическое число графа

- Мин. кол-во цветов, в кот. его можно покрасить

Пример: $\chi($  $)=3$ $\chi($  $)=2$

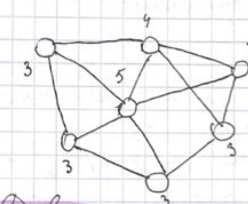
$\chi($  $)=5$ $\chi(K_n)=n$

полный граф, n вершин

Замечание: Если $k \geq \chi(G)$, то G можно покрасить в k цветов

Утв. $\chi(G) \leq \max \deg V + 1$

макс. степень вершины



- это степени вершин

$\max \deg = 5$

$\chi(G) \leq 6$

Д-во. Инд. по кол-ву вершин

База $n=0$, $n=1$ - верно $\max \deg = 0$ $\chi(G) \geq 1$

Переход G, V - вершина $\underbrace{\max \deg}_{= \Delta}$. Уберем её,

получим $G' = G \setminus V$

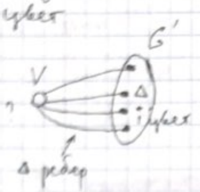
$\max \deg G' \leq \max \deg G = \Delta$

раскрасим G' в $\Delta + 1$ цвет

цвета V: запрещены $\leq \Delta$

цветов $\Rightarrow \geq 1$ цвет можно

и т.д.



Утв. G - планарный граф

- граф с непересекающимися ребрами

$\Rightarrow \chi(G) \leq 5$

Пример: карта со странами

Д-во

1) в G есть вершина степени ≤ 5

Если нет, $\Rightarrow \deg V \geq 6 \Rightarrow$

$\sum \deg V \geq 6 \cdot n$, где $|V| = n$, $|E| = m$

$\Rightarrow 2 \cdot m \geq 6n \Rightarrow m \geq 3n$, но в планарном

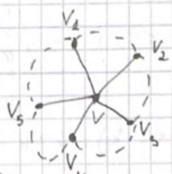
G: $m \leq 3n - 6$?? - противоречие

Утв. Раскрасим в 5 цветов по индукции

Б. графы из 1, 2, 3, 4, 5 вершин - можно раскрасить

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

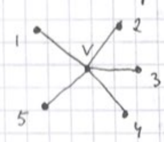
Берём V: $\deg V \leq 5$



- раскрасим G' без V

- если соседи V_i используют ≤ 4 цветов

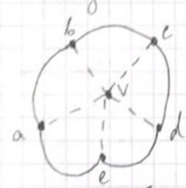
- осталось



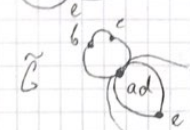
- все 5 цветов

- не подходят

Тогда:



1 грань без V. сделаем $\tilde{G}$



$\tilde{G}$ - $n-2$ вершины

$\Rightarrow$ можно раскрасить

вернёмся к G, a и d имеют один цвет $\Rightarrow$ есть цвет для V - ч.т.д.

Утв $\chi(G) \le 4$ (проблема 4ёх красок)

Хроматические многочлены

$\exists$ $\chi(G,k)$ – это функция „сколько способов раскрасить G в k цветов"

$$\chi(\circ\!-\!\circ,\ k) = \begin{cases} k=0 & 0 \text{ – способов} \\ k=1 & 0 \text{ – способов} \\ k=2 & 2 \text{ – способа} \\ k=3 & 6 \text{ способов} \\ k=4 & 12 \text{ – вариантов} \end{cases}$$

$\chi(\circ\!-\!\circ,\ k) = k(k-1)$

$\chi(\circ\ \ \circ,\ k) = k^2$

Утв $\chi(\emptyset_n, k) = k^n$

граф из n вершин без ребер

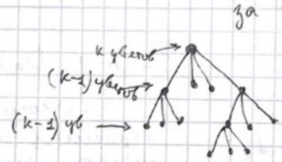
Утв $\chi(K_n, k) = k(k-1)(k-2)\dots(k-n+1) = k^{\underline{n}}$

убыв. степ.

Утв $\chi(T_n, k) = k(k-1)^{n-1}$

дерево

подвесим дерево за $\forall$ вершину



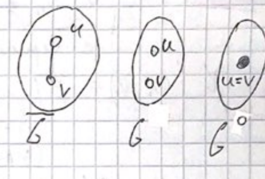
Утв $\bar{G}$ – граф

u, v – вершины с ребром (u, v)

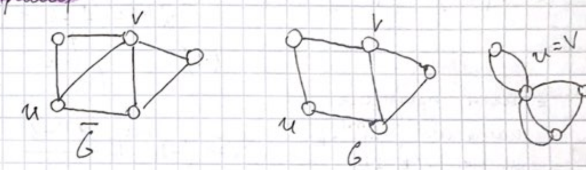
$G = \bar{G} \setminus (u,v)$

без ребра

$G^\circ = \bar{G}$ где u, v стянуты в вершину



Пример



$$\chi(G,k) = \chi(\bar{G},k) + \chi(G^\circ,k)$$

способы раскр. G, где u и v – разн. цвет

способы раскр. G где u, v – один цвет

Следствие

$$\chi(\bar{G},k) = \chi(G,k) - \chi(G^\circ,k)$$

Пример $\chi(\ \ ,k) = \chi(\ \ ,k) + \chi(\ \ ,k) =$

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

Утв $\chi(C_n, k) = ?$

n - вершин

$$\chi(C_n, k) = \chi(\text{дерево}, k) - \chi(C_{n-1}, k) =$$

n - вершин (дерево)

$$= k(k-1)^{n-1} - \chi(C_{n-1}, k) = \underbrace{k(k-1)^{n-1} - k(k-1)^{n-2} \ldots + C_{n-2}}_{} =$$

$$= k(k-1)^{n-1} - k(k-1)^{n-2} + k(k-1)^{k} \ldots \mp k(k-1) \pm k =$$ геом. прогр.

$(-1)^n$ $\quad$ $(-1)^{n+1}$

= геом прогрессия

начало: $(-1)^{n+1} \cdot k$

множ: $-(k-1)$

членов: n шт

$$S = (-1)^{n+1} k \frac{q^n - 1}{q - 1} =$$

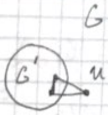
$$= (-1)^{n+1} k \frac{(1-k)^n - 1}{-k} = (k-1)^n - (-1)^n$$

Утв. 1)] G имеет висячую вершину u

$$\chi(G, k) = \chi(G', k) \cdot (k-1)$$

раскрасить u

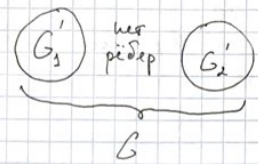
2)



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

раскрасить u

3) $G = G_1' \cup G_2'$ нет рёбер между G_1' и G_2'



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

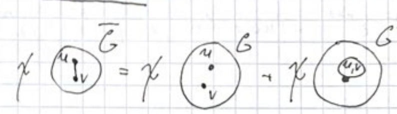
Пример:

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

$$= (k-1)(k-2)\,\chi(\ldots, k) = (k-1)(k-2)^2\,\chi(\Delta) = (k-1)(k-2)^2 \cdot k(k-1)(k-2)$$

k^3

Напоминание



$$\chi(\overline{G}) = \chi(G) + \chi(G^0)$$

Утв. $\chi(G, k)$ - это многочлен

1. стар. коэф. = 1
2. степень = n (кол-во вершин)
3. знаки чередуются
4. млад. коэф. = 0
5. коэф. при $k^{n-1} = \pm m$ - кол-во рёбер

Д-во Индукция

по кол-ву вершин, при равном кол-ве вершин: кол-во ребер.

База пустой граф из n вершин

$$\chi(\because, k) = k^n = \pm 0\, k^{n-1}$$

Переход

с ребром $$\chi(G, k) = \underbrace{\chi(\overline{G}, k)}_{\substack{\text{работает ИП} \\ \text{(мало рёбер)}}} - \underbrace{\chi(\cdot, k)}_{\substack{\text{работает ИП} \\ \text{(мало вершин)}}}$$

1) ст. коэф. $(1k^n -) - (k^{n-1} - ..)$

2) степ $= n$ (кол-во вершин)

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

4) мл. коэф. $= 0 - 0 = 0$

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

$= -$ ребер $G' \cdot k^{n-1}$

На практике

$$\chi(\triangle\!\!<, k) = (k-1)\chi(\triangle, k) = (k-1)k(k-1)(k-2) =$$

раскрою скобки $= k^4 - 4k^3 + 5k^2 - 2k$

Утв. $\chi(G)$ - хроматическое число (миним. число цветов k для раскраски)

$\chi(G, k)$ $\quad \chi(G) - 1$

$k = 0, 1, 2, \ldots$ - корни многочлена

$\chi(G)$ - не корень

Эйлеровы графы

Опр. Эйлеров путь - простой путь, содержащий все рёбра

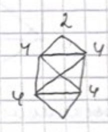
Эйлеров цикл - цикл, содержащий все рёбра

не проходя дважды по ребру

Утв. G содержит Эйлеров цикл $\Leftrightarrow$ G связен, и $\deg V$ - чёт. $\forall v \in V$

Пример.

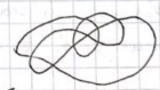
- нет Эйл. цикла



- есть Эйл. цикл

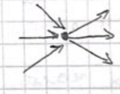
Д-во

$\Rightarrow$



Эйл. цикл

Граф связен

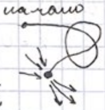


кол-во входов = кол-во выходов $\Rightarrow \deg$ чётна

<= (в обратную сторону)

- начнём строить цикл
- идём из ∀ вершины, выбираем ребро, которое ещё не использовали

начало



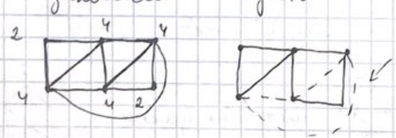
- в каждой вершине по пути использовали чёт ребер (k входов, k выходов)
- +1 ребро, через которое вошли (2k+1)

=> использовали нечёт рёбер

=> есть ещё одно, по нему можно уйти

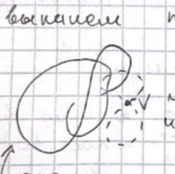
- кроме начальной. из неё вышли на 1 раз больше.

=> то закончим ходить в нач. вершине



не все обошли

- вычитаем просмотренные рёбра

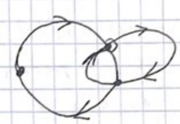


т.к. G связен из нач. вр. x можно попасть в ∀ вершину и ребро

построенная часть, в остатке все степени чётные

Повторим процесс из v ∈ 1 циклу, из которой ведёт первое ребро

- объединим 2 цикла



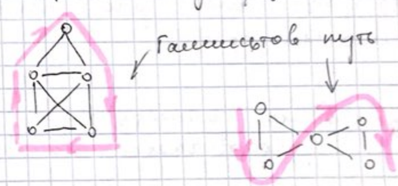
- продолжать пока все рёбра не объединятся в цикл. ч.т.д.

Th G содержит эйлеров путь <=>

1) G связен
2) [степени всех вершин чётные
   [степени всех вершин кроме двух чётные

Опр: Гамильтонов цикл/путь
- простые пути / циклы по всем вершинам

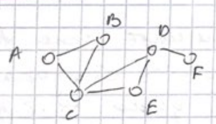
Гамильтонов путь



Длины путей в граф

Длина пути в графе - кол-во ребер 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 пути между вершинами или $+\infty$, если пути нет.

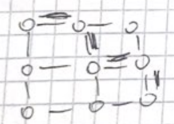
Обозначение: $d(x, y)$ - расстояние от x до y

Пример $d(A, F) = 3$

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

Пример

В примере выше диаметр = 3 (достигается на AF)



- диаметр 4

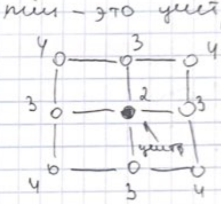
Опр. Для каждой вершины графа $G = (V, E)$ можно посчитать max расстояние до других вершин

$$r(V) := \max \{ d(V, S) \mid S \in V \}$$

Радиус:

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

Те вершины, на которых достигается min - это центр



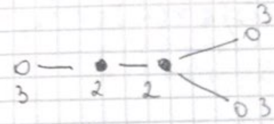
$r(G) = 2$ - радиус графа

Центров может быть много

- 4 центра

$r(G) = 4$

или

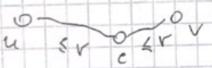


$r(G) = 2$

Утв. в $G=(V,E)$ $d(G) \le 2r(G)$

Д-во: $\exists$ - c центр графа

$u, v \in V$



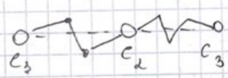
$d(c,u) \le r$
$d(c,v) \le r$

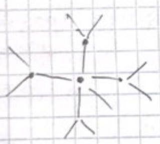
$\Rightarrow d(u,v) \le 2r \Rightarrow d(G) = \max d(u,v) \le 2r$

ч.т.д.

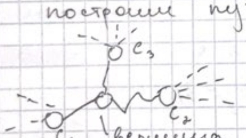
Утв. В дереве ≤ 2 центров

$\exists$ их 3.





построим пути между c_1 c_2 потом c_2 c_3 (в дереве ровно 1 путь между верш.)



вершина развилки c_0:

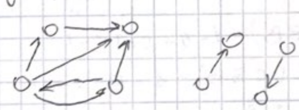
$r(c_0) < r(c_1) = r(c_2) = r(c_3) = r(G) = r$

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

(ребра в ор. графе иногда называют дугами)

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

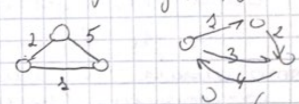
Пример



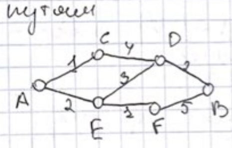
Замечание У ребер будут веса

$G=(V,E)$ вес - это $f: E \to R$

т.е. $\exists$ число каждому ребру



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



$d(A,B)$

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

$d(ACEFB) = 1+4+3+1+5=1$

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

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

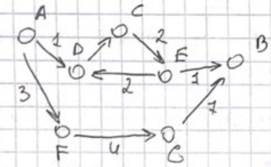
min=7

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

Замечание расстояние во взвешенном графе не всегда существует

$d(F,G) = 4$

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



$d(A,B) = ?$

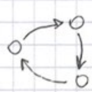
$d(ADCEB) = 1 - 5 + 2 + 1 = -1$

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

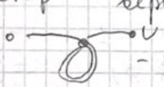
и т.д. $\min = -\infty$

Утв. В графе есть все расстояния $\Leftrightarrow$ в графе нет цикла отр. длины

Д-во Если есть цикл < 0 $\Rightarrow$ $\forall$ две вершины этого цикла не имеют расстояния (мин $-\infty$)



Если нет расстояния, т.е. для u, v есть пути сколь угодно маленькие, $\exists$ есть путь длиннее $n = |V|$ ребер $\Rightarrow$ повтор вершин в пути



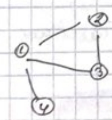
– это и будет отр. цикл

Как хранятся графы в компьютере (представление графа в компьютере)

1. Матрица смежности: таблица вершины × вершины

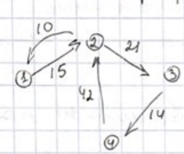
$$a(i,j) = \begin{cases} 0, \text{ если нет рёбер} \\ 1, \text{ если есть рёбра} \end{cases}$$

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



– симметрична для неориент. графа

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



	1	2	3	4
1	$+\infty$	15	$+\infty$	$+\infty$
2	10	$+\infty$	21	$+\infty$
3	$+\infty$	$+\infty$	$+\infty$	14
4	$+\infty$	42	$+\infty$	$+\infty$

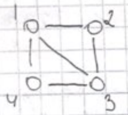
Объем памяти: $n^2 = |V|^2$

(2) Списки смежности

– для каждой вершины храним список соседей

Пример 1: 2(15) 2: 1(10), 3(21) 3: 4(14) 4: 2(42)

Пример



1 : 2 3 4

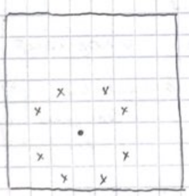
2 : 1 3

3 : 2 4

4 : 1 3

Память $\approx |E|$ кол-во ребер

Пример Задача: обход конём шахм. доски



граф: вершины - клетки 8×8

рёбра - вершины через ход коня

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

- Задача обхода конём = Гамильтов цикл в этом графе

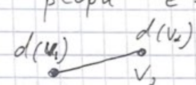
Задача Дано две вершины u, v, найти $d(u, v)$ и путь, на котором достигается это расстояние

Замечание Оказывается что найти путь от u до v это тоже самое, что найти путь от u до всех верш.

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 текущее найденное расстояние. В начале

$d(u) = 0 \quad d(v) = +\infty$ если $v \neq u$

Релаксация ребра $e = (v_1, v_2)$



если $d(v_1) + f(v_1 v_2) < d(v_2) \Rightarrow$

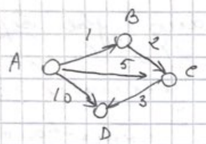
$$d(v_2) := d(v_1) + f(v_1 v_2)$$

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

(в неор. графе o—o = o⇄o, т.е. две релакс. на ребро)

Пример

$n = 4$ (4 вершины)



	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: —//—

CD: —//—

ребра

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

B : C(2)

C : D(3)

D :

релаксац.:

$\underset{0}{A} \xrightarrow{1} \underset{+\infty}{B}$

$0 + 1 < +\infty$

$\underset{1}{B} \xrightarrow{2} \underset{5}{C}$ $\quad 1+2 < 5$

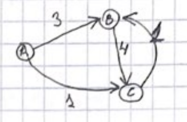
$\underset{3}{C} \xrightarrow{3} \underset{10}{D}$ $\quad 3+3 < 10$

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

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

Лекция 09.11.21

Алгоритм Форда-Беллмана



A : 3B, 1C

B : 4C

C : 1B

пути из A.

сначала

	A	B	C
d:	0	$+\infty$	$+\infty$
	0	3	$+\infty$
	0	3	1
	0	2	1

1. релаксируем

$\underset{0}{A} \xrightarrow{3} \underset{+\infty}{B}$

$0 + 3 < +\infty$

2. релаксируем

$\underset{0}{A} \xrightarrow{1} \underset{+\infty}{C}$

$0 + 1 < +\infty$

3. $\underset{3}{B} \xrightarrow{4} \underset{1}{C}$

$3 + 4 < 1$
не подходит

4. $\underset{1}{C} \xrightarrow{1} \underset{3}{B}$

$1 + 1 < 3$

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

AB, AC, BC, CB

– нет улучшений

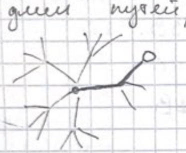
Ответ:

A	B	C
0	2	1

Корректность Алгоритма

Th В конце массива d содержится расстояния от A

Д-во После i-го цикла релаксации всех рёбер, d хранит числа $d(v) \le \min$ длин путей, в которых $\le i$ рёбер



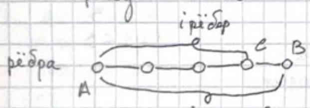
Действительно. База индукции: $i=0$ $\min$ (пути из 0 рёбер)

– только A-A

$d(A)=0$

$d(u)=+\infty$

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



По предположению $d(C) = d_B + (A, C)$

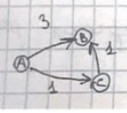
ребро CB

– длина пути $\underbrace{A - C}_{d(C) = dist(C)} - B = dist(C) + вес(CB)$, $dist(C) = d(C)$

Проверка:

$d(C) + вес(CB) \le d(B)$ – верно, т.к. путь оптимален

$\Rightarrow d(B) = d(C) + вес(CB)$



(путь после второй релаксации будет равен 2)

Почему $n-1$ этап?

Оптимальный путь не содержит цикл

$\Rightarrow$ —//— $\le n-1$ ребро. ч.т.д

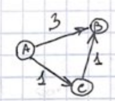
Замечание Мы вычисляем только расстояния, но путь неизвестен. Как восстанавливать пути?

Будем сохранять информацию об успешных релаксациях.

Prev – массив вершин

Если релаксация $u \to v$ успешная, то

$Prev[V] = u$ (Оптимальный путь в V лежит через u)



d	A	B	C
	0	$+\infty$	$+\infty$
AB:	0	3/A	$+\infty$
AC:	0	3/A	1/A
CB:	0	2/C	1/A

Восстанавливаем путь в B

$A \to C \to B$, $A = prev(C)$, $C = prev(B)$

В общем случае путь $A \to v$ это

$A \to \dots prev(prev) \to prev(v) \to v$

Алгоритм Дейкстры

В отличие от ФБ требуется, чтобы веса $w(e) \geq 0$

Алгоритм. Дан граф $G = (V, E)$, $A \in V$

Найти расстояние до всех вершин $d(u) = dist(A, u)$

Алг. $d(A) = 0$ $d(u \neq A) = +\infty$ – обработанные вершины

Повторяй n раз: $(n = V)$ – вершины

Выбрать $u \in V \setminus P$, где $d(u) \to \min$ (из необраб. $\min(d)$)

for $e \in$ ребра из u, $e(u, v)$

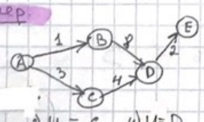
релаксируем ребро e

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

d:

A	B	C	D	E	
0	$+\infty$	$+\infty$	$+\infty$	$+\infty$	$P = \varnothing$
	1	3	$+\infty$	$+\infty$	$P = \{A\}$
		...			
		3	9	$+\infty$	$P = \{A, B\}$
			7	$+\infty$	$P = \{A, B, C\}$
				9	$P = \{A B C D\}$

Пример.



1) $u = A$

$A \xrightarrow{1} B$
0 $+\infty$

$A \xrightarrow{3} C$
0 $+\infty$

2) $u = B$

$B \xrightarrow{?} D$
1 $+\infty$

3) $u = C$

$C \xrightarrow{4} D$
3 9

4) $u = D$

Эффективность: $|V| \times |E| \times \log|V|$

выбор min

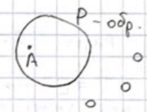
Корректность:

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

обр. вершины

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

Переход:



P – обр.

необр. вершины

Выбираем $u = \min$ вершине из $V \setminus \{P\}$

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

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

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

$dist(u)$

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

Корректность

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

пути

$u - \bigcirc - v$

от 1 до k

База: $k=0$

$d(u,v) = \min (u - \bigcirc - v)$

нет

по индукции по k

Переход $k \to k+1$

$u - \bigcirc - v$

вершины от 1 до $k+1$

$\exists$ есть опт. т. путь

из $u - \bigcirc - v$

$k+1$

1) $\longrightarrow$ в нем нет $k+1$, => его длина это $d(u+v)$

2) $\longrightarrow$ есть $k+1$ $u - \bigcirc - (k+1) - \bigcirc - v$

его длина $d(u, k+1) + d(k+1, v)$

это ровно проверка из нашего цикла

- меньший вариант запишется в d

В конце $\forall$ вершин

$d(u,v) = \min(u \overset{\forall}{-} v)$

Замечание

чтобы восстанавливать путь, можно ввести массив.

through: if $d(u,v) > d(u,k) + d(k,v)$

$\Rightarrow d(u,v) =$ —//—

$through(u,v) = k$

Для восстановления пути:

1) $A = \dots I \dots B$

$= th(A, B)$

2) $A \dots I \quad B$

$I \quad a$

$= th(A, I) \quad = th(I, B)$

3) и т.д. если $th(x,y)$ нет записи ->

ребро $x-y$ - это оптимальный путь

Замечание

Алг. Флойда ищет транзитивное замыкание бинарных отношений.

$\exists$ R - бин. отн. на M

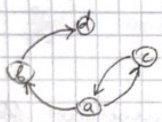
$\exists$ $\bar{R}$ - транз. замыкание R, если

1) $\bar{R}$ - транз.

2) $\forall \bar{\bar{R}}$ $\bar{\bar{R}} \supset \bar{R} \supset R$

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

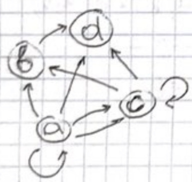
Пример.



– не транзитивно

$a R b$, $b R d$, но $a \not R d$

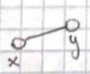
станет транзитивным (минимальные добавл. рёбер)



Th $\exists R$ – бин. отн. $\exists G = (M, R)$ – граф отношения

тогда $\overline{R}$ – это $x \overline{R} y \Leftrightarrow$ есть путь $x - y$ в G

Д-во 1) $\overline{R} \supset R$ т.к. если $x R y \Rightarrow$ есть путь из 1 ребра $\Rightarrow x \overline{R} y$



2) $\overline{R}$ – транзитивно, т.к. $x \overline{R} y$, $y \overline{R} z$ $\Rightarrow x \overline{R} z$ (путь $x - z$ тоже есть)

3) $\exists\ \overline{\overline{R}} \supset R$, $\overline{\overline{R}}$ – транзитивно

$\exists$ есть путь x в y $x \to x_1 \to x_2 \to x_3 \cdots \to y$

$x R x_1 \Rightarrow x \overline{\overline{R}} x_1$

$x_1 R x_2 \Rightarrow x_1 \overline{\overline{R}} x_2$ $\Rightarrow$ $x \overline{\overline{R}} x_2$, $x_2 \overline{\overline{R}} x_3$ $\Big| \Rightarrow x \overline{\overline{R}} x_3 = \ldots \Rightarrow x \overline{\overline{R}} y$

$\Rightarrow \overline{\overline{R}} \supset \overline{R} \supset R$

Применим Флойда к графу $G = (M, R)$

d_0	y
x	-1

$d_0(x, y) = 1$, если $x R y$

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

После конца алгоритма:

	y
x	∞

7

Замыкание, $x \overline{\overline{R}} y$, если $d(x, y) < \infty$

<u>на практике:</u>

Алг. транз. замык.

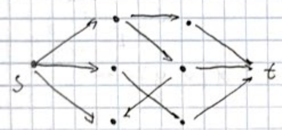
```
R̄ := R
for k ∈ M
  for x ∈ M
    for y ∈ M
      if xRk & kRy
        R̄ ← (x,y) т.е. сделать xR̄y
```

Потоки в сетях

Опр. Сеть – это $G = (V, E)$ – граф ориент.

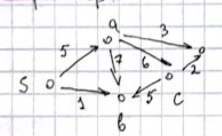
$s \in V \quad \nexists e = (u, s)$ – ничего не входит

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



$c : E \to N$ – пропускн. способность рёбер. целые > 0

Пример

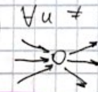


–

Опр. Поток f в сети G – это $f : E \to \mathbb{R}$

1) $0 \le f(e) \le c(e)$ – поток макс. в сети

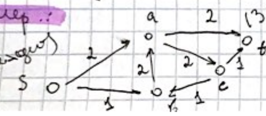
2) $\forall u \ne s, t$



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

Пример:

(3 выходит)



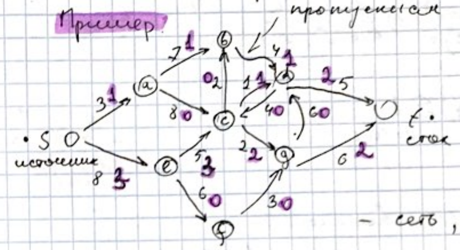
(3 входит)

– в каждую вершину входит и выходит одинаковая сумма

Лекция

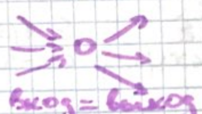
Пример

пропускная способность



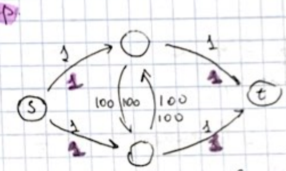
– сеть, черный c

– поток, фиол. f



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

Пример



– поток корректный

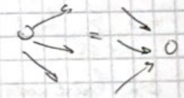
величина потока в примере 1 : 4

в примере 2 : 2

Th. Дана сеть (G, c), поток f на G

Тогда

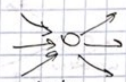
$$\sum_{u : e=(s,u)} f(e) = \sum_{u : e=(u,t)} f(e)$$



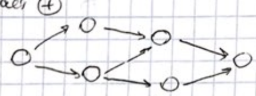
Рассмотрим:

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

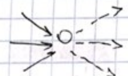
$$\ominus \underbrace{\sum_{e: e=(u,s)} f(e)}_{0} + \sum_{e: e=(u,t)} f(e) +$$



$$+ \sum_{v\in V\setminus\{s,t\}}\ \sum f(e) = \text{вытекает} \ (4)$$



$$(4) \sum_{v\in V\setminus\{s,t\}}\ \sum_{e: e=(v,u)} f(e) \ominus$$



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

$$- \sum_{e: e=(s,u)} f(e) - \sum_{e: e=(v,u)} f(e) - \text{втекает} - 0 =$$

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

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

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

$$\sum_{e: e=(u,t)} f(e) = \sum_{e: e=(s,u)} f(e)$$ – **Опр** величина потока $W(f)$ (размер)

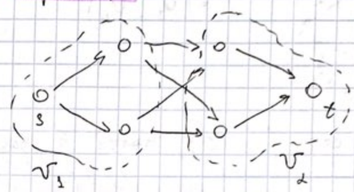
Определение:

Разрез в сети (G, c), $G = (V, E)$. Разрез $G = (V_1, V_2)$

$s \in V_1$, $t \in V_2$, $V_1 \cup V_2 = V$

$V_1 \cap V_2 \neq \emptyset$

Пример



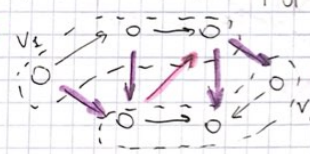
и другие...

Опр. 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:



– прямые E_c^+
– обратные E_c^-

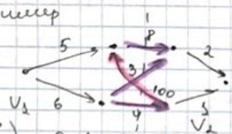
$$E_c = E_c^- \cup E_c^+$$

Определение.

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

Обозначение $c(C)$

Например



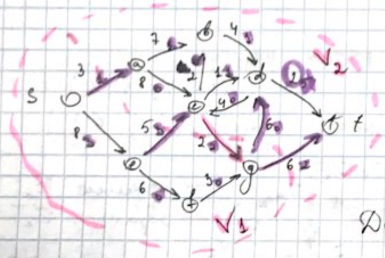
$C = (V_1, V_2)$

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

Утв. Пусть есть сеть (G, c), поток f, разрез ~~разделение~~ $C = (V_1, V_2)$

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

Проверим.



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

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

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

Да! $4 = 6 - 2$

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

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

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

2) $\sum_{\substack{e=(u,v)\\ u \in V_1,\, v \in V_2}} (f(e) - f(e)) + \underbrace{\sum_{e \in E_c^+} f(e) + \sum_{e \in E_c^-} [-f(e)]}_{\text{см. условие}} =$

$= 0 +$ величина из условия ч.т.д.

Обозначение:

$w(C, f)$ – величина (размер) потока через разрез

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

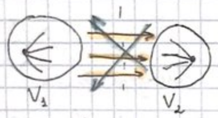
Замечание $\forall C \;\; w(f) = w(C, f)$ – по теор.

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

Утв. Дано G, c – сеть, C – разрез

Тогда $w(f) \le c(C)$

Доказательство



$w(f) = w(C, f) =$

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

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

Утверждение

В сети G $\quad w(f_{max}) \leq c(C_{min})$

где $w(f_{max}) = \max\limits_{f\text{-поток}} w(f) \qquad c(C_{min}) = \min\limits_{C\text{-разрез}} c(C)$

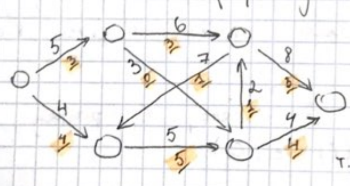
Th Форда - Фалкерсона $\quad w(f_{max}) = c(C_{min})$

в сети (G, c), $c(e) \in N_0$

$G = (V, E)$

для простоты считаем, что пропускающие способности целые

Дополнительный граф для потока



$\overline{G}$ имеет $\overline{V} = V$

$\overline{E}$:

если $f(e) < c(e)$ $\quad e = (u, v)$

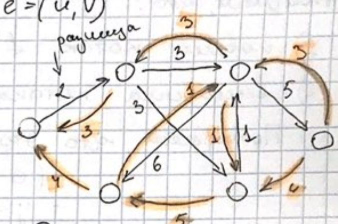
т.е. $e' = (u', v')$

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

если $0 < f(e)$ $\quad e = (u, v)$

т.е. $e'' = (v', u')$

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



- Доп. граф

Д-во. Начнем с 0-го потока и будем его постепенно увеличивать.

Построим доп. граф $\overline{G}$ и найдем в нём путь из s в t.

$s \to \circ - \ldots \to \circ \to t$

Найдем $\min g(e)$ на этом пути $\exists$ это x

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

$\circ \to \circ \to \circ \to \circ$

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

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

2) $f(e) - x \qquad f'(e) := f(e) - x$

Поймём, что:

1) новый поток f' остался потоком
2) величина потока увеличилась на x

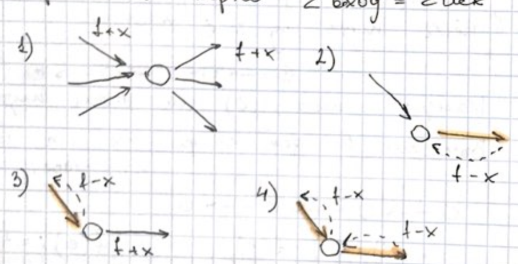
Проверяем, что это поток.

$(0 \leq f'(e) \leq c(e))$

- уменьшаем по обратному пути
- увеличиваем по прямому пути

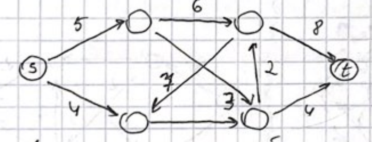
$c(e) - (f(e) + x) \geq 0$

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



Итого f' – поток. ч.т.д.

Лекция

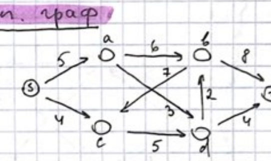


(граф с прошлой лекции)

Пример. Строим пути

- сначала поток = 0

Доп. граф



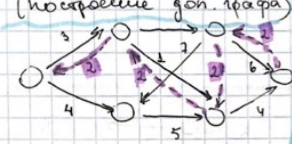
Путь идём из s в t

1) $s - a - d - b - t$
   5 3 2 8 ⇐ min=2

- добавляем к потоку +2 на каждое из этих рёбер

1) $c - f$

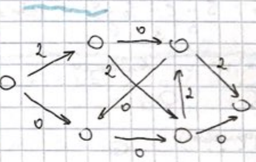
(построение доп. графа)



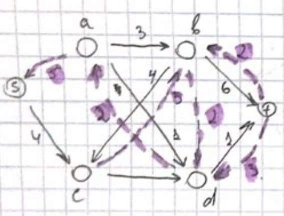
путь $s \overset{3}{-} a \overset{6}{-} b \overset{7}{-} c \overset{5}{-} d \overset{4}{-} t$

– min 3

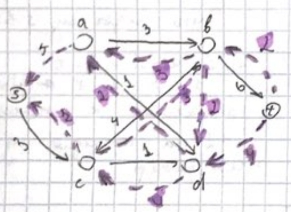
Поток.



Доп. граф.

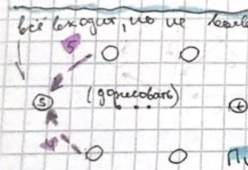


Путь $S - c - d - t$
$4 \quad 2 \quad 1 - \min 1$



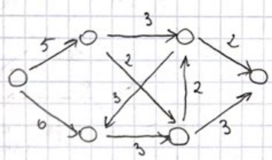
Путь: $S - c - b - t \leftarrow \min 3$
$3 \quad 3 \quad 6$

всё входит, но не выходит

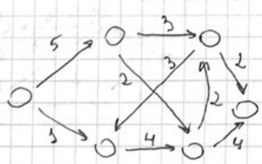


Путь: $s - t$! нет!

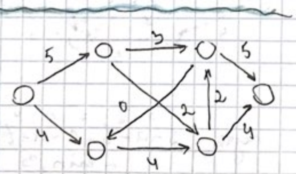
Добавляем к потоку



+ min 3



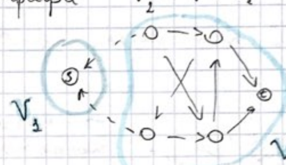
+ min 1



Продолжаем д-во th. Форда – Фалкерсона

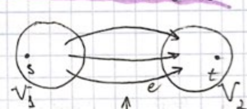
? Если пути нет, то поток оптимален?

$\exists V_1$ – вершины, достижимые из s по рёбрам доп. графа $V_2 = V + V_1$



$t \in V_2$ т.к. нет пути $s \to t$

– Получим разрез!



рёбра E_c^+

рёбер нет в доп. графе

⇓

в доп. графе веса $= 0$

$c(e) - f(e) = 0$

- $c(C) = \sum\limits_{e \in E_c^+} c(e) = \sum\limits_{e \in E_c^+} f(e) = \sum\limits_{e \in E^+} f(e) - \overbrace{\sum\limits_{e \in E^-} f(e)}^{0} = c(f)$

– Течёт ли по обр. рёбрам?

– нет, иначе V_1 неверн

Итого мы нашли разрез C:

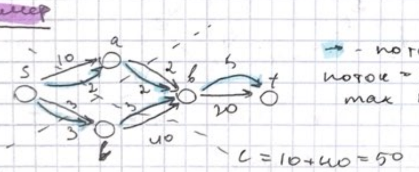
$c(C) = c(f)$

В прошлый раз мы показали: $\forall C$ – разрез, $\forall f$ – поток

$c(C) \geq c(f)$

Получается C min разрез. f – max поток

Пример



→ – поток

поток = 5

max поток

$c = 10 + 40 = 50$

$c(C) = 2 + 3 = 5$

min разрез

Замечание Метод ФФ строит min разрез и max поток

Утверждение Если каждый раз искать путь с min кол-вом рёбер, то время поиска max потока $\sim V^2 E$

– без д-ва

Утверждение Для плоской сети эффективно искать верхние пути

Задача о паросочетаниях

– Дан двудольный граф $G = (U, V, E)$

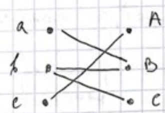
Пример

Опр. паросочетание в G – это $P \subset E$, где рёбра из P не имеют общих вершин

Определение Максимальное паросочетание, это $P \subset E$

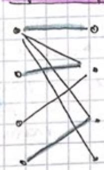
$|P| \to$ max из возможных

Пример.



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

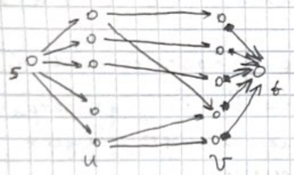
Пример.



4 нельзя!

3 можно

Сводим к задаче о потоке



рёбра из $s \to U$

$V \to t$

рёбра из $U \to V$ слева направо

$c(e) = 1$

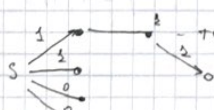
Утверждение

- Каждому потоку (из $f = 0, 1$) соответствует паросочетанию

Доказательство

- Рёбра с $f(e) = 1$ это рёбра паросоч.

=> поток

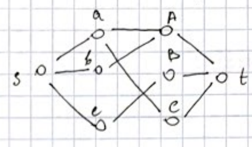


- только одно из рёбер = 1

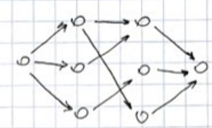
<= Паросочетанию соответствует поток, где $f(e) = 1$ для ребер паросочетаний

Следствие Размер max паросоч. = размер max потока

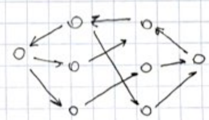
Строим паросочетание методом ФФ.



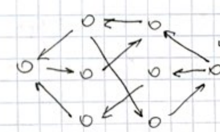
Строим доп. граф, по без чисел (всегда = 1)



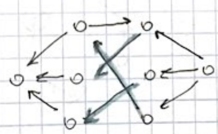
путь s a A t



путь s c B t



путь s b A a C t

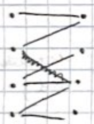


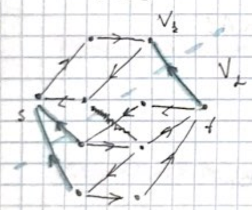
больше путей нет

Ответ: aC, bA, cB (обр. рёбра)

Лекция

Пример паросочетаний:



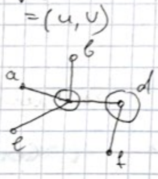


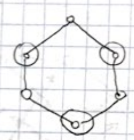
Задача о максимальном контр. мн-ве

Опр. $\exists\ G=(V,E)$ $C\subset V$ – контролирующее мн-во, если $\forall e\in E$ $= (u,v)$ $u\in C$ или $v\in C$

Пример

$C=\{c,d\}$



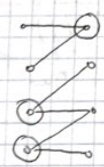


Замечание $C=V$ – контролирующее мн-во

Задача

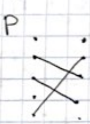
– Найти КМ min размера
(Будем решать для двудольного графа)

* в примере сверху *



Утв. В двудольном графе $G=(U\cup V,E)$
$\exists\ C$ – к. мн-во
$\exists\ P$ – паросоч. Тогда $|C|\geq|P|$

Док-во P



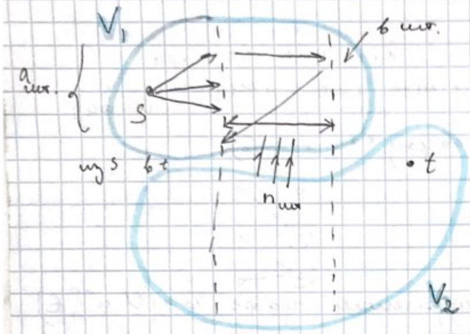
– у каждого ребра $e\in P$ $= (u,v)$ есть вершина u или $v\in C$

Утв. $\exists\ G=(U\cup V,E)$ – двуд. граф

– Размер макс. паросоч. = размеру min КМ

~~Д-во Построим макс. паросоч по Форду-Фалкерсону~~

Д-во: Построим max поток по ФФ и Δ разрез



$|u| = x$

$|v| = y$

$|u \cap V_1| = a$

$|v \cap V_1| = b$

- $c([V_1 V_2]) = \sum 1$
  разрез
  e - ребро исх.
  u, v $u \in V_1$
  $v \in V_2$

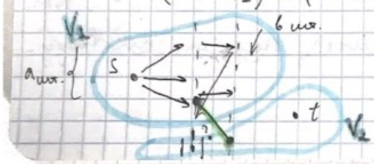
- $c([V_1 V_2]) = \sum 1 = (x - a) + b + n$

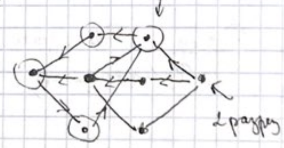
Итого

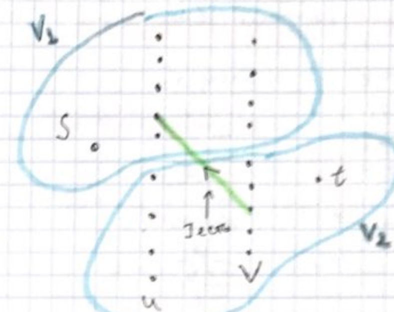
$m = c([V_1 V_2]) = x - a + b + n \leq x - a + b + m$

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

- Возьмём в качестве КМ
  $c = (u \cap V_1) \cup (v \cap V_1)$





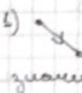


КМ - это

$u \cap V_2$ $\quad v \cap V_1$

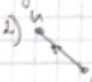
$\exists$ есть ребро $e = (u$

1)



значит $v \in V_2$

2)



как попасть в u?
только из v

- невозможно

Значит, в min разрезе нет ребер между $u \cap V_2$ и $v \cap V_2$

Вывод 1. C - КМ $(u \cap V_2) \cup (v \cap V_1)$

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

$\Downarrow$

$|P| = |C|$

Поиск в глубину, ширину

1) структура данных для хранения вершин
D - стек или очередь
$v \to D$ положить v в D
$v \leftarrow D$ посмотреть $D \to v$

стек: первый вошел, последний вышел
очередь: —//— первый вышел

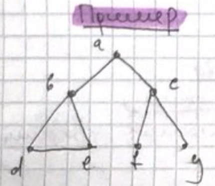
Пример

	стек:	очередь
a → D	a	a
b → D	ba	ba
c → D	cba	cba
∠D		
D →	ba − c	a
∠D		cb
		b

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

D ← V_0 нач. вершина
пока D не пуст
$u = \angle D$
если есть ребро (u, V) $\overline{\text{Были}(V)}$
тогда V → D
Были (V) = 1
иначе достать D → u

Пример



Вглубь:

D: a, ba, dba, edba → dba, ba, a, ca → gca, ca, a

Лекция.

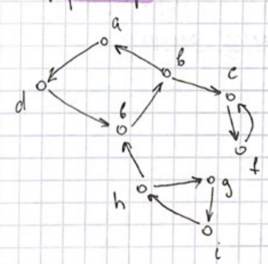
Поиск в глубину / ширину

D – стек D – очередь

Алгоритм поиска:

– Дана нач. вершина
D ← u Used(обр) ← ∅ (обработанные)
пока D ≠ ∅
V ← peek D (смотрим)
если есть ребро V – w, где w ∉ Used
иначе D ← w; Used = Used ∪ {w}
← D (убираем вершину из D)

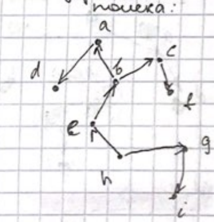
Пример:



в глубину из h

стек
h
he
heb
heba
~~hebad~~
hebc
h~~ebc~~
hg
~~hgi~~

дерево поиска:



В ширину (очередь)

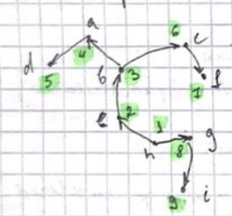
h → he → heg → egb → gbi → bia c → acd → cdf

Введем к алгоритму нумерацию:

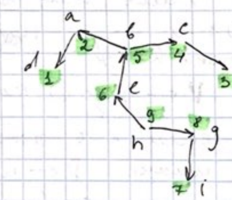
$n(u)$ – номер, какой попала в D

$b(u)$ – обр. номер, какой ушла из D

Рассмотрим на дереве поиска:



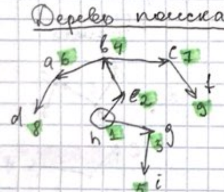
обр. нумерация:



Замечание:

При поиске в ширину $n(u) = b(u)$

Дерево поиска в ширину очередь



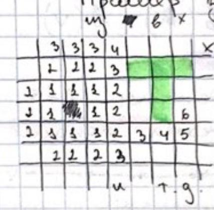
Утверждение:

- Поиск в глубину перебирает вершины в том же порядке, что и в Алгоритме Дейкстры (веса рёбер = 1)

Действительно, добавление вершины в D – это релаксация рёбр $v-w$

Удаление из D – убирание вершин с min расстоянием

Пример задачи:



Полный поиск в глубину

$\boxed{\text{Пока}}$ есть непосещенная вершина u

$\boxed{\text{поиск в глубину}}$ (u)

dfs — deep first search

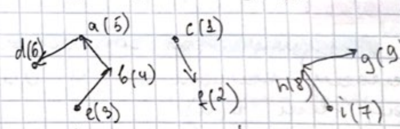
bfs — breadth first search

Пример

dfs(c)

dfs(e)

dfs(i)



(прямая путь)

Утверждение

- 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 попаем раньше?

1) сначала попаем в u

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

В стеке будет $u \dots v$
=> сначала из стека уйдет v, потом u. ■

2) сначала v => закончим поиск, не попав в u

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

=> номер $b(v)$ присвоится раньше, чем $b(u)$ ■

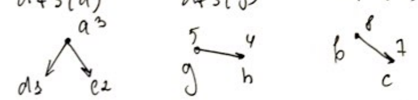
Следствие

Алгоритм топ. сортировки

Делаем полный dfs и мин. порядок задаем через $b(u)$

Пример:

dfs(a) dfs(g) dfs(b)



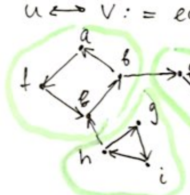
Ответ: d e a h g f e b
1 2 3 4 5 6 7 8

Компонент сильной связности

Напоминание

G - ор. граф. Введем отношение $\leftrightarrow$ на V

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



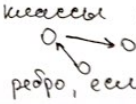
← 3 комп. сильной связности

$c \leftrightarrow f$
$a \leftrightarrow e$
$i \leftrightarrow h$
$b \leftrightarrow c$

Напомним:
$\leftrightarrow$ отн. экв-ти
классы эквивалент. называются комп. сильной связности

Определение: $\exists$ - G - ор. граф.

$G^\circ = (V^\circ, E^\circ)$ - граф, конденсации, если
$V^\circ = V/\leftrightarrow$ (классы экв.)

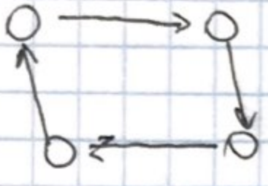
В примере 

E° u° в v° есть ребро, если $\exists e = (u, v)$ где $u \in u^\circ$, $v \in v^\circ$

Продолжение последней лекции

Замечание

G° не имеет циклов



$\Leftrightarrow$ у вершины $\leftrightarrow$

Утверждение

$\exists\, G = (V, E)$ — ор. граф.

G° — граф конденсации G

Делаем полный ~~обход~~ dfs в G

Тогда: Если в G° есть путь из u° в v°,

~~[то $b(u) > b(v)$ для любых $u \in u^\circ$, $v \in V^\circ$]~~

то $\max\limits_{u \in u_0} b(u) > \max\limits_{v \in V_0} b(v)$

Д-во аналогично прошлого утв.

Следствие Поиск компонент сильной связности

1) полный dfs в G
2) Делаем dfs по обратным ребрам G