Лекция№1

Бинарные отношения

Определение

М-множество $\neq 0$

R$\subset$ MxM- бинарное отношение

Пояснение

MxM- множество пар из элементов R

Допустим M=a,b,c

MxM=(a,a),(a,b),(a,c),(b,a),(b,b),(b,c),(c,a),(c,b),(c,c)

или M= $\mathbb{N}$

Отношение R-это подмножество пар

Обозначение

(x,y)$\in R$ пара (x,y) принадлежит отношению

мы будем писать xRy

вместо (x,y)$\notin$R мы пишем ~~xRy~~

Пример

1.M=R >=R=(x,y): x>y

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

(3,4) $\notin R$ ~~3R4 3>4~~

2.M=$\mathbb{R}$ отношение $\geq$ $\quad 7 \geq 6, 7 \geq 7$

3.M=$\mathbb{R}$ отношение = $\quad \begin{array}{cc} 7 = 7 & 7 \neq 8 \\ (7,7) \in = & (7,8) \notin = \end{array}$

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

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

2#2 $<=> 2^2 > 2$

~~7#8~~

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

4$\vdots$2 10$\vdots$5 ~~7$\vdots$0~~

~~2$\vdots$4~~ ~~10$\vdots$3~~

7. $\equiv_3$ $\quad$ 0$\equiv_3$3 1$\equiv_3$4 1$\equiv$ 7

~~0$\equiv_3$ 2 1$\equiv_3$ 8~~

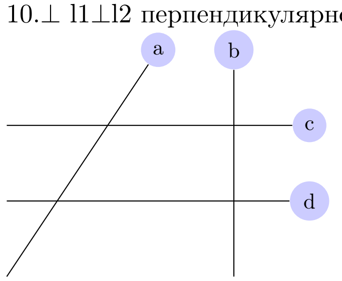
8.M=$\mathbb{N}$ $\quad$ aЦв, если в числе a 'в' цифр

100Ц3 238Ц3 ~~238Ц8~~

9.M=прямые на R^2

||-l1 и l2 если l1 не пересекает l2 или l1=l2

10.$\perp$ l1$\perp$l2 перпендикулярно



$b \perp c$
$d||c$ ~~b⊥a~~
$a||a$
11.Студенты ЛЭТИ
x≻y средний балл за последнюю сессию больше у x
12. M=пользователи одноклассники
x→y, если ’y’ в друзьях у ’x’
Иванов → Петров
~~Петров → Посов~~
Свойства бинарных отношений
1. Определение
Бинарное отошение R на M называют рефлексивным, если $\forall x \in M \quad xRx$
$(x, x) \in R$
Замечание
Отношение не рефлеквсивно <=> $\exists x \qquad xRx-$контрпример
Примеры
=-рефлексивно ∀ x: x=x
≥ $\forall x$: x≥x
≈ $\forall x: \quad x \approx x$, т.к. |x-x|=0<1
$\vdots \quad x \vdots x$
>-не рефлексивно ~~2>2~~
Ц- не рефлексивно 3Ц3
→ ~~Посов→Посов~~
⊥ не рефлексивно ~~a⊥ a~~
Определение
Б отношение R на множестве M называется антирефлексивным, если $\forall x$ ~~xRx~~
Замечание
R-не антирефлексивно <=>
$\exists x$: xRx-контрпример
Примеры
>-антирефлексивно ~~x>x~~
⊥ ~~1⊥1~~
→ нельзя быть в друзьях у себя
Ц- не антирефлексивно 1Ц1
ЗамечаниеЦ - не антирефлексивно и не рефлексивно
2)не бывает R, которое и рефлексивно и антирефлексивно
(рассмотрим a $\subset M \quad \to aRa$=> aRa=> не ар
~~aRa~~ => не р
Симметричность
Определение
Бинарное отношение R на множестве M симметрично, если $\forall x, y$ xRy<=> yRx
Замечание
R- не симметрично, <=> $\exists x, y$: xRy, ~~yRx~~ контрпример

Пример

=- симметрично x=y <=> y=x

≈ - симметрично x≈y |x-y|<1 |y-x|<1

$\vdots$- не симметрично 4$\vdots$2 ~~2$\vdots$4~~

||, ⊥- симметрично

Ц- не симметричен

Определение

Бинарное отношение R на множестве M антисимметрично, если $\exists x \neq y$ xRy, yRx- контрпример

Пример

>: x≠y, x>y => ~~y>x~~

Попробуем построить контрпример

x≠y, x>y, y>x- невозможно

=> нет контрпримера=>антисимметрично

≥ x≠y,$x \geq y$, y≥x- нет контрпримера

= x≠y,x=y, y=x- нет контрпримера

$\equiv_3$ 1≠ 4,1 $\equiv_3$ 4, 4$\equiv_3$1- контрпример

$\vdots$ над $\mathbb{N}$ x≠y, $x \vdots y$, y$\vdots$x нет для $\mathbb{N}$

$\vdots$ над $\mathbb{Z}$ 4≠-4, 4$\vdots -4$, -4$\vdots$4 не антисимметрино

Лекция№2

Антисимметричность

$\vdots$ на $\mathbb{Z}$- не антисимметрично

$$\left.\begin{array}{r} -2\vdots 2 \\ 2\vdots -2 \end{array}\right. \text{-контрпример}$$
$$2 \neq -2$$

$\vdots$ на $\mathbb{N}$- антисимметрично $x \neq y \quad x\vdots y \quad y\vdots x$=> такого не бывает

$x \neq y \quad x\vdots y \quad => \quad y \not\vdots x$

Определение

R- бинарное отношение на M- ассиметричность, если

$\forall x, y$ xRy=> ~~yRx~~

($x \neq y$- у антисимметричность)

контрпример- xRy, yRx

Утверждение

R- симметрично <=> R-антисимметрично и антирефлексивно

Пример

>- асиметрично $\forall x, y$ x>y=> ~~y>x~~

□(пустое)- асиметрично (пусто, когда R=0)

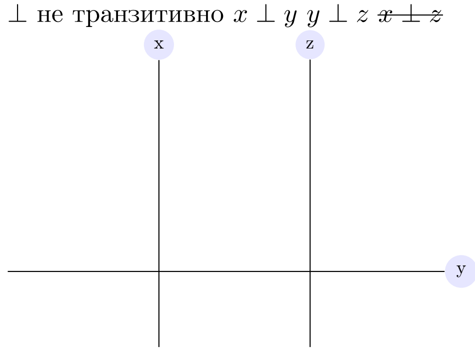
"выше асиметрично

"начальник"x нач y =>~~y нач x~~

R-бинарное отношение транзитивно, если

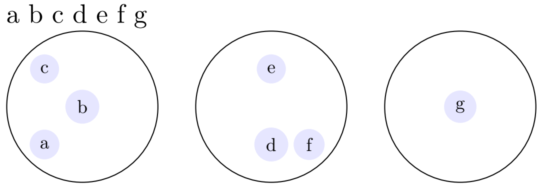
$\forall x, y, z$ x xRy, yRz => xRz

Контрпример
> трназитивно x>y, y>z=> x>z
$\geq$ транзитивно
$\vdots$ транзитивно $x\vdots y$, y$\vdots z$=> x$\vdots$ z
$\perp$ не транзитивно $x \perp y$ $y \perp z$ $\cancel{x \perp z}$

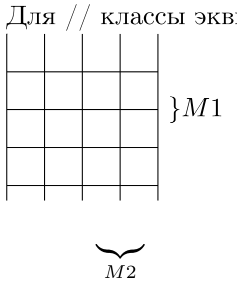


Ц(кол-во цифр) 100Ц3 3Ц1
не транзитивно 100Ц1
Определение
Отношение R называется отношением эквивалентности, если
R-рефлексивно, симметрично, транзитивно
Пример
= на $\mathbb{R}$(или $\forall$ другом множестве)
$\forall$x x=x- рефлексивно
$\forall$x,y x=y => y=x- симметрично
$\forall$x,y,z x=y, y=z => x=z- транзитивно
= -это ОЭ
||- параллельность
$\equiv_3$-сравнение
$\geq$- не ОЭ
т.к. не симметрично
$x \geq y$=> y$\geq$x
z$\geq$1, 1$\geq$z
$\approx$- не ОЭ(по транзитивности)
отношение $\uparrow$ на $\mathbb{N}$ x$\uparrow$y, если у x и y поровну цифр
$2 \uparrow 5$ $35 \uparrow 100$
$12 \uparrow 42$
ОЭ $x \uparrow x$- рефлексивно
$x \uparrow y$ => $y \uparrow x$- симметричность
$x \uparrow y$, $y \uparrow z$=> $x \uparrow z$- транзитивность
=,||,$\equiv_3$,$\uparrow$- ОЭ
Определение
R-ОЭ на множестве M
x$\in$ M, класс элемента x
M$_x$=y|xRy
Пример

= M_5=5
$\equiv_3$ M_2=2,5,8,11...
// M_e=////////...
Утверждение
R-ОЭ на M
$\forall x, y \in M$ $M_x = M_y$ или $M_X \cap M_y$=0
Доказательство
$\supset$ $M_X \cap M_y$ $\neq$ 0=> $\exists$z$\in$ M_x, z$\in$ M_y =>xRz,
yRz=>(симм.)=>zRy=>(транз.)=>xRy
Теперь проверим, что класс $M_x = M_y$
Возьмем u$\in$ M_x, проверим, что u$\in$ M_y
u$\in$ M_x => xRu
xRy=>yRx=>yRu=>$u \in M_y$
Следствие R-ОЭ не M
тогда M разбито на несколько классов эквивалентности
M=$M_1 u ... u M_n$
$M_i \cap M_j$=0
= на ℕ=1 и 2 и 3...
$\equiv_3$ на ℕ=0,3,6,9,...
1,4,7,10,...
2,5,8,11,...
Замечание
Если есть M=0 разбитое на M_i=0
M=M_1u...uM_n и $M_i \cap M_j$
Тогда можно ввести отношение R
xRy, если $\exists M_i$: x,y$\in$ M_i
a b c d e f g

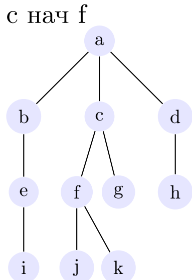


aRb bRc ~~aRd~~ gRy ~~gRa~~
Для // классы эквивалентности



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

R-бинарное отношение
R- транзитивно, антисимметрично
1)рефлексивно-нестрогий порядок
2)антирефлексивно- строгий порядок
обозначения обычно $\succeq$ нестрогий $\succ$строгий
$a \succ b$ b$\succ$ $c =>$ a$\succ$ c
антисимметрично a$\succ$ b
b$\succ$ a
Примеры
$>$ на $\mathbb{R}$- строий порядок
$\geq$ $\mathbb{R}$- не строгий порядок
$\vdots$ на $\mathbb{N}$- не строгий порядок
нач
a нач b
a нач c
b нач f
c нач f



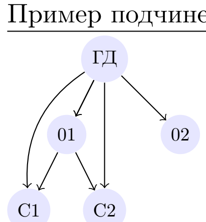
Определение
$\supset$ R-строгий или нестрогий порядок
R-линейный, если $\forall x \neq y$ xRy или yRx
R- частичный иначе ($\exists x \neq y$ ~~xRy~~ ~~yRx~~)
Примеры $>$, $\geq$-линейный порядок
$\vdots$- частичный ~~$2 \vdots 3$~~ ~~$3 \vdots 2$~~
нач- частичный
Утверждение
R- порядок(строй или нестрогий) на M- конечное |M|$<\infty$
Тогда $\exists x-$минимальный, т.е. $\forall y : x \succ$ y
Лекция 3
Утверждение R-отношение порядка (строгое или нестрогое)
тогда $\exists$ x-минимальный, т.е. $\forall$ y$\neq$ x
Пример
$\geq$ на 1, 2, 3, 4, 5
1- min т.к. $\forall$y 1 $\geq$y
Пример
$\vdots$ на 2, 3, 4, 5, 6

2- min $\forall$y $\cancel{2 \vdots y}$

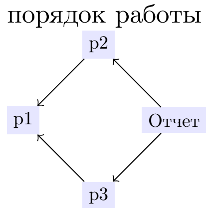
3-min $\cancel{3 \vdots y}$

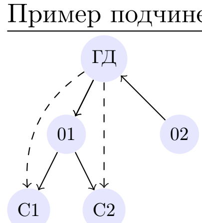
5-min $\cancel{5 \vdots y}$
4,6- не min

$4 \vdots 2$ $6 \vdots 2$
<u>Докозательство</u>
берем x_1- $\forall$ элемента множества
если он не min => $\exists x_2 \neq x_1$ $x_1 \succ x_2$($\succ$-это R)
если x_2 не min=> $\exists x_3 \neq x_2$ $x_2 \succ x_3$
если x_3 не min=> $\exists x_4 \neq x_3$ $x_3 \succ x_4$
если не можем найти min элемент
=> т.к. множество M конечно
в какой-то момент $x_i = x_j$(первый повтор)
$x_i \succ x_{i+1} \succ x_{i+2} \quad \succ x_{j-1} \succ x_j = x_i$
$\succ$- транзитивно => $x_i \succ x_{j-1} \quad x_{j-1} \succ x_i \neq x_i$ невозможно, см. антисимм
<u>Определение</u>
Отношение R_1 на множестве M рассматриваетR_2 на M, если $R_2 \subset R_1$
Замечание. R_1 доставляет пары где xRy
Замечание. xR_2y => xR_1y
<u>Теорема</u>
о топологической сортировке
Если $\succ$ - отношение порядка на M(множество конечное)
то $\exists \succ\succ$- отношение лин порядка на M т.к. $\succ\succ$ расширяет $\succ$
<u>Пример подчинения</u>

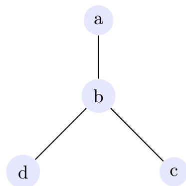


не линейный порядок
ГД-01-02-C1-C2
ГД-01-C1-C2-02
ГД-01-C1-02-C2
<u>Докозательство</u>
Нахождение min элемента отношения $\succ$
$\supset$ это x$\in$M
удалим x_1 из M теперь имеем $\succ|$ M-x_1
очевидно, новое отношение тоже антирефлексивно, рефлексивно, антисимметрично, транзитивно
в нем тоже есть min элемент, это x_2
удаляем x_2 из M и продолджаем
итого, имеем последовательность $x_1, x_2, x_3, ...$
Вводим новый порядок $x_i \prec x_j$ для i<j...

$x_1 \prec\prec x_2 \prec\prec ... \prec\prec x_n$
почему $\prec\prec$ расширяет $\prec$
Если x$\prec$ y => x был удален раньше y
=> $x \prec\prec y$
Замечание этот алгоритм(поиск min и удаления) не самый эффективный, лучше - сделать поиск в глубину с обратной нумерацией
Замечание топологическая сортировка практически важная задача
порядок работы


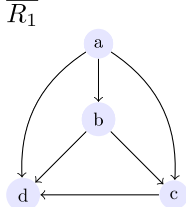
Транзитивное замыкание
был порядок- расширили до линейного (Топологическая сортировка)
было отношение- расшилири до транзитивного(Транзитивность замыкается)
Пример
подчиненное отношение
Пример подчинения


H.R 01
01 R C1
но ~~H.R C1~~
Если добавить в отношение, что H R C1, H R C2- станет транзитивным
Теорема
$\supset$ R- бинарное отношение на M
$\exists\overline{R}$ отношение на M
1)$\overline{R}$ расширяет R(R$\subset$ $\overline{R}$)
2)$\overline{R}$ транзитивно
3)$\overline{R}$ min транз. расширение
т.е. если $\overline{R}$- тр расширяет R, то $\overline{\overline{R}} \supset \overline{R}$
Докозательство(не для алгоритма)
рассмотрим се транзитивные расширения
R_i возьмем $\overline{R}$= $\cap$ $\overline{R}_i$
т.е. ьеем только те пары, которые есть во всех транзитивных расширениях
Пример
M=a,b,c,d

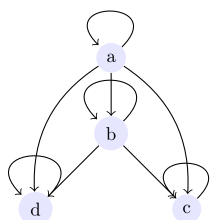


aRb bRc bRd

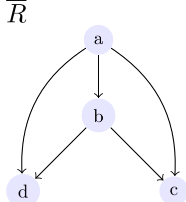
$\overline{R_1}$



стало транзитивным



$\overline{R}$



только те пары, которые есть везде

Проверим, что $\overline{R}$ подходит под улсовия транзитивности

0)$\overline{R}_i$ существует? R_i полное отошение MxM

1)$\overline{R}$ - расширяет

$\supset xRy$ => $\forall\overline{R}$, x$\overline{R}_i$y => x$\overline{R}$y

2)$\overline{R}$-транзитивно? $\supset$ x$\overline{R}$y, $\supset y\overline{R}z$=>

$\forall\overline{R}_i$ x$\overline{R}_i$y y$\overline{R}_i$z => x$\overline{R}_i$z- транзитивно

=> x$\overline{R}$z

3) $\overline{\overline{R}}$=$\overline{R}_i$ >R т.е. R=$\overline{R}_i\cap$...

Графы!!!

Определение

неориентированный граф

G=(V,E) где V- множество(вершины)

E$\subset$ (u,v), где u,v $\in$ V

Замечание как рисовать

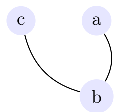
веришны $\bullet$ или $\bigcirc$

ребра $\rightarrow$ линии между $\bigcirc$

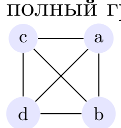
важно, только что реберо соединеяет

Примеры

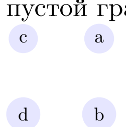
обычный граф



полный граф



пустой граф



Определение

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

Замечание. V=vertex E=edge

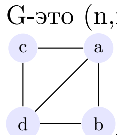
Определение

|G|- размер (порядок) графа |V|- кол-во вершин

|V|=n(обычно)

|E|=m(обычно) количество ребер

G-это (n,m) граф



- (4,5) граф

Определение

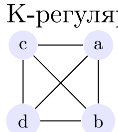
Степень вершины V∈ V-это |(v,u)|(u,u) ∈ E| степ 3

(кол-во ребер с этой вершиной) степ 2

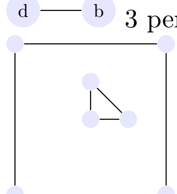
обозначаем deg V

Определение

K-регулярный граф- это граф у которого $\forall V \in V \quad degV = K$



3 рег граф



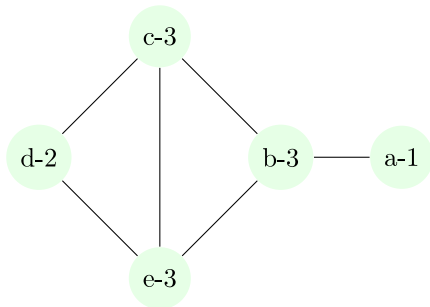
2 рег граф

Лекция 4

Напоминание

Графы- множество вершин и ребер

Степени(напротив названий)

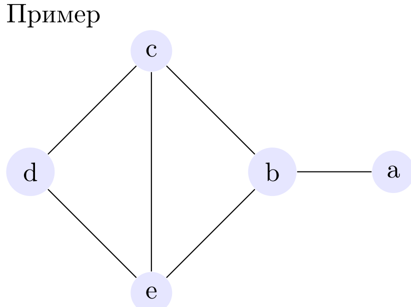


Определение

Пусть в графе G(V,E)- последовательность $v_1 e_1 v_2 e_2 ... v_{n-1} e_{n-1} v_n$

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

Пример



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

2)ab

3)aba

4)abcdecdeba

Определение

Путь замкнутый- если $V_1 = V_n$. "4) замкнутый

не замкнутый- открытые($v_i \neq v_n$)

Определение

Путь простой, если$e_i \neq e_j$ при $i \neq j$

"4)"непростой, т.к. 2 раза de

5)bedce- простой, но не замкнутый

Пути	все ребра разные(простые)	все вершины разные
Замкнутый	простой замкнутый путь	цикл
Открытый	простой открытый путь	цепь

цикл- путь из одной вершины в неё же

путь может проходить через себя же неограниченное количество раз, идёт из одной вершины в другую

простой путь- путь не проходящий по пройденному пути, но имеющий точки пересечения

цепь- путь безпересечений

Теорема

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

Доказательство

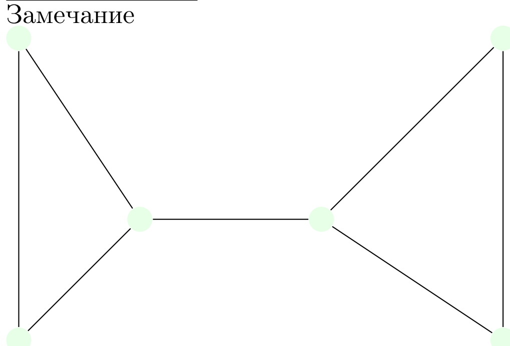
$\supset$ путь u $e_1 v_1 e_2 v_2 ... e_n$v
рассмторим все пути из этих ребер и выберем min, это будет цепь
иначе: $u ... u_i ... u_j ... v \quad \supset v_i = v_j$
укоротим $u ... v_i = v_j ... v$??

Теорема

Если есть простой замкнутый путь через ребро е => есть цепи через е

Доказательство аналогично

Замечание

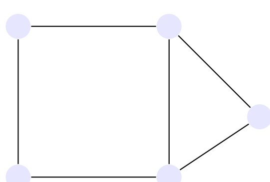
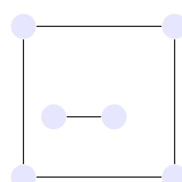
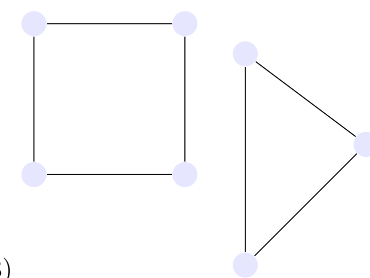


dbacbd- не простой путь(е-повторяется)
цикла через е нет

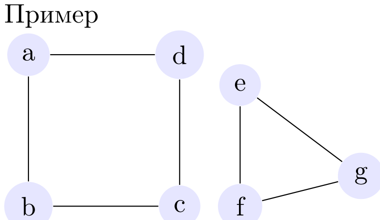
Связность графа

Определение

Граф связан, если $\forall \quad u, v \in V \quad \exists$ цепь(путь) из u в v
пример

1)  2)  3) 

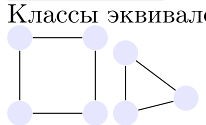
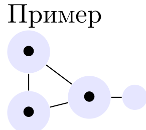
1-связан
2,3-не связаны
Введем отношение $\equiv$ на вершинах графа:
U$\equiv$ V, если $\exists$ путь из U в V
Пример

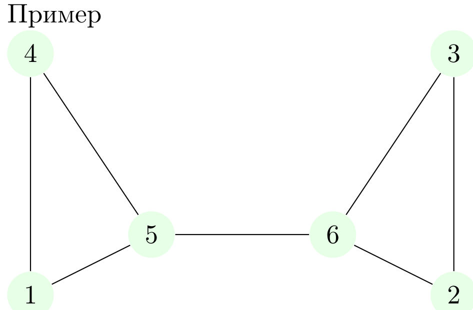


$a \equiv c \quad a \equiv d \quad e \equiv g$ ~~a≡e~~

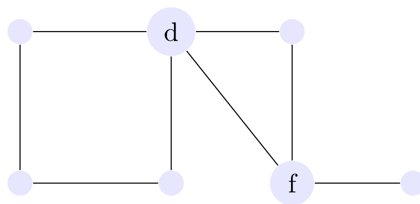
Проверим, что $\equiv$- это отношение эквивалентности

1)рефлексивность $u \equiv u$-верно, путь u
2)симметричность $u \equiv v \quad => \quad v \equiv u$ путь $ue_1v_1...v$ путь $v...v_1e_1u$
3)транзитивно $u \equiv v, v \equiv \omega$ путь $\underbrace{\underline{ue_1...v}\overline{v...\omega}}$ v не повторяется,

а входит в 2 пути
Определение
Классы эквивалентности ≡-это компоненты связности
 - 2 компонента связности
Определение
$G_1 = (V_1, E_1)$- подграф G, если $V_1 \subset$ V $E_1 \subset$ E
Пример


Помеченные • являются подграфом
Замечание
G-свой подгаф
0-пустой граф-подграф чего угодно
Определение
G ребро e называется мостом,
если комп связности G< кол-воткомп. связности (V,Ee)
Пример


56 мост
Определение
Степень связности графа G- это количество ребер,
которые надо выкинуть,чтобы G стал несвязным
Определение
Двусвязный граф- надо выкинуть ≥2 ребер, чтобы он стал несвязным
Замечание: двусвязный <=> нет мостов и связи
Определение
Вершина $v \in V$ называется точкой сочленения,
если количество компонента связности < количества компонента связности
G'=(Vv, E'(u,v)|(v,u)∈ E)



Пример

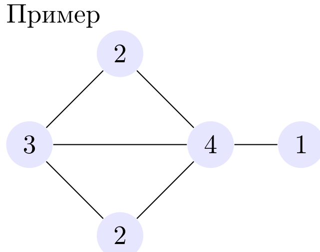
Теорема

В графе G=(V,E)

если deg(u)- степень вершины u

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

Пример



ребер 6=1/2(3+2+2+4+1)=6, верно

Доказательство

deg(v)= количество ребер, выходящих из вершин $\sum\limits_{v\in V}$= все ребра посчитали дважды=2|E|

Следствие:

1)сумма степеней вершин всегда четна

2)вершин нечетной степени- четно

Пример

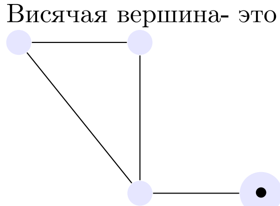
15 инопланетян, по 3 руки у каждого,мгу ли они взяться за руки, чтобы не было свободных рук?

Решение:

нет, эьо граф из 15 нечетных вершин степени 3(нечет.)

Определение

Висячая вершина- это вершина степени 1



•- висячая

Теорема

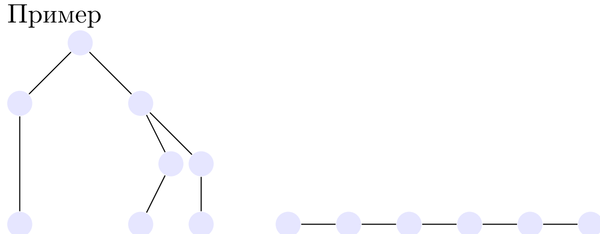
Если в графе есть ребра, но нет висячих вершин, то ∃ цикл

Доказательство

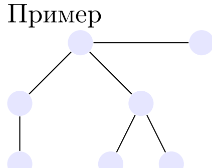
Берем ребро e=$(u1U2)$

$u_1 -^e -u_2 -^{e_1} -u_3$

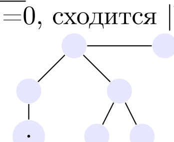
u_2- не висячая вершина=> из неё есть еще ребро(u_2u_3)
u_3- не висячая вершина=> из неё есть еще ребро(u_3u_4)
...
Продолжаем, пока очередность u_n не будет равна u_i
путь u_1 $u_{i+1}...u_n$-цикл(ребра разные, вершины разные)
Определение
Дерево- связный граф без циклов
Пример



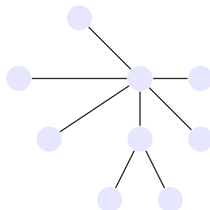
Теорема
В любом дереве ≥ 2 висячих вершины
Докозательство
Берем $\forall$ вершину, если она не висячая, идём по ребру,
если опять не висячая, есть ребро и т.д. циклов нет=>конец
Чтобы найти вторую, надо начать из первой
Теорема
Если G- дерево, то |V|=1+|E|
Пример



7 вершин, 6 ребер
Докозательство по индукции(кол-во вершин)
Б. |V|=1 0 |E|=0, сходится |V|=|E|+1

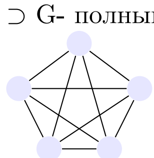


П. $\supset$ |V|=n+1, .- висячая
найдем висячую и удалим с её ребром
G'=(V'v,E'e), тоже дерево, т.к. связан, нет циклов=> |V'|=1+|E'|,
но |V|=|V'|+1 |E|=|E'|+1
отсюда следует |V|=1+|E|
Лекция 5
Напоминание
Дерево- связной граф



|E|+1=|V|

$\supset$ G- полный граф, $\forall$ u $\neq \subset$ V соединены ребром



Если n вершин(|V|=n), то ребер

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

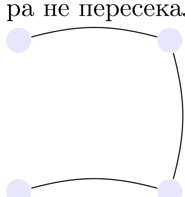
2)степени всех вершин n-1

$\sum\limits_{v\in V} deg(v)$=2|E|=>(n-1)n=2|E| Ответ: $\frac{n(n-1)}{2}$

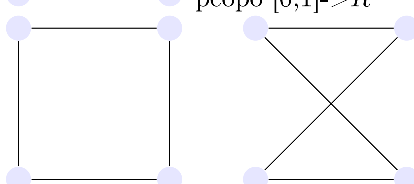
<u>Планарные графы</u>

<u>Определение</u>

G-планарный граф, если его можно нарисовать а плоскости так, чтобы ребра не пересекались

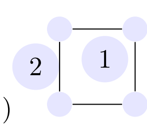


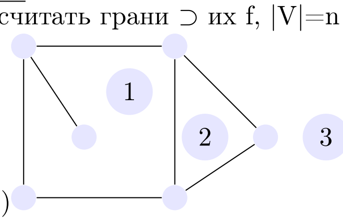
ребро [0,1]->R^2



- планарный, но неправильно нарисованный

<u>Формула Эйлера</u> Если планарный граф G=(V,E) нарисован на плоскости у него можно посчитать грани $\supset$ их f, |V|=n |E|=m

1) 

2) 

Тогда: n-m+v=2

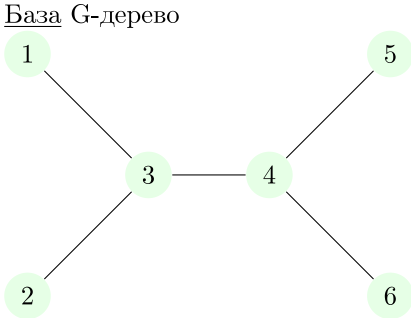
проверим:

1) 4-4+2=2 2) 6-7+3=2

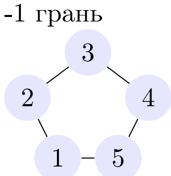
<u>Докозательство</u>

Индуция по количеству ребер

<u>База</u> G-дерево



-1 грань

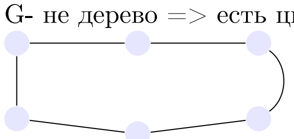


вокруг грани есть цикл, а дерево без циклов

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

переход G- не знаем, верно ли (G,G’-связная планарные графы), если G’ имеет меньше ребер=> верно

G- не дерево => есть цикл, берем ребро цикла



-вокруг него 2 грани, удалим ребро, получим G’ тоже связен и планарен

n’ m’ f’- вершины, ребра, грани G’

$n'=n$

$m'=m-1$

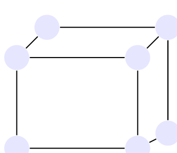
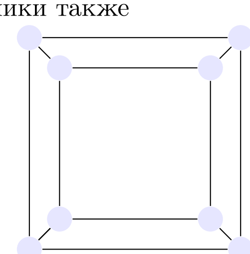
$f'=f-1$

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

Следствия

1) неважно, как рисовать планарный граф, количество граней постоянно

2) про многогранники также

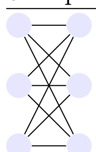



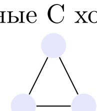
8-12+6-2

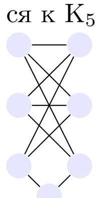
3) Если G планарный(не обязательно связный) то $n-m+f=1+|$комп. связности G$|$

Докозательство упражнение

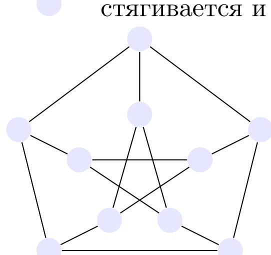
4. $\supset$ У каждой рани вокруг $\geq$ 3 ребра
3f$\leq \sum_{y \in grani}$ количество ребер вокруг y $\leq$ 2m каждое ребро посчитно 1 или 2 раза=> 3f$\geq$ 2m
но n-m+f=2, умножим на 3
3n-3m+3f=6 => 3n-3m+2m$\geq$6=> 3n-m$\geq$6=> m$\leq$3n-6
Итого m$\leq$ 3n-6 в связном планарном графе
Следствие
полный граф при n=5- не планарен
<u>Докозательство</u>
n=5
$m = \frac{5 \cdot 4}{2} = 10$
10$\leq$ 3*5-6=9??
<u>Замечание</u>
K_5- полный граф n=5
<u>Утверждение</u> граф $K_{3,3}$ тоже не планарный



<u>Докозательствро</u>
n=6 m=9 9$\leq$ 3*6-6 нет противоречия
Сколько граней, если полярный 6-9+f=2 =>f=5 граней
В $K_{3,3}$ все циклы цетные С ходим лево-право или прав-лево
=> у грани $\leq$ 4 ребра  невозможно
4f $\leq \sum$ребер грани g$\leq$ 2m=> m$\geq$2f но ~~9≥ 2*5~~
<u>Теорема</u> Понтрягина-Курлотовского
Связный граф планарен <=> если не содержит полуграфов стягивающих-ся к K_5 и $K_{3,3}$



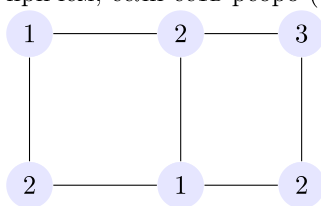
стягивается и $K_{3,3}$ не планарен



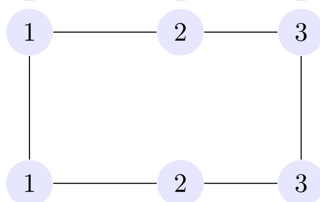
не планарен, в нем есть K_5
<u>Хроматизм</u>

Определение

$\supset$ G=G(V,E)- граф раскраска графа
G в к цветов это функция c: V->1..K
причем, если есть ребро (U,V), то C(U)$\neq$C(V)

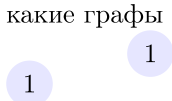


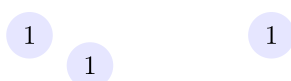
-раскраска



-не раскраска

какие графы можно раскрасить в 1 цвет



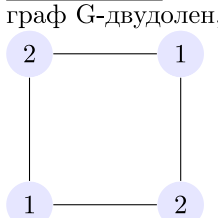


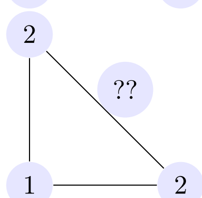
-это граф без ребер

какие графы можно раскрасить в 2 цветах

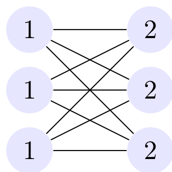
Определение

граф G-двудолен, еслли его можно раскарсить в 2 цвета?



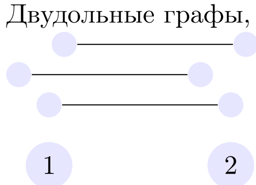


$K_{3,3}$-двудольный



Замечание

Двудольные графы, часть рисуют из 2ух частей(долей)

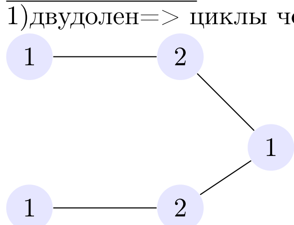


Теорема

Двудолен <=> вс циклы G имеют четную длину

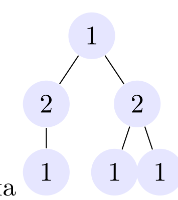
Доказательство

1)двудолен=> циклы четные



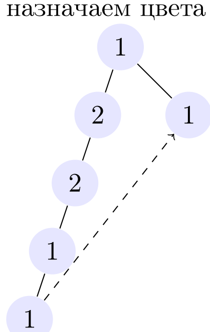
цикл поровну цвета 1 и цвета 2

2)циклы четные ?=> двудолен

"подвесим граф за вершину"∀ вершина 

ребра назад не рисуем

назначаем цвета по уровням

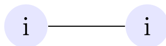


-обратное

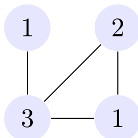
Почему обратное ребро не соединяет оинаковые цвета? Потому что иначе цикл нечетный

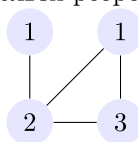
Напоминание

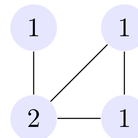
К-раскраска графа k цветов у вершин



нельзя ребром один цвет



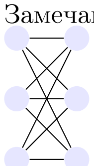
и


- раскраски, но не раскраска:


граф имеет 1 раскраску <=> граф бз ребер

граф имеет 2 раскраски- двудольный граф(по определению)

Замечание

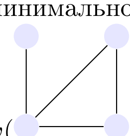
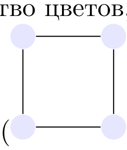

граф двдолен<=> все циклы четные

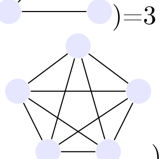
Определение

G=(V,E)- граф

ϕ(G)- хроматическое число графа

минимальное количество цветов, в котром его можно раскрасить

ψ()=3 ψ()=2

ψ()=5

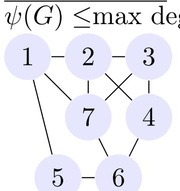
все должны быть разные $\psi(K_n) = n$ полный граф n вершин

Замечание

Если $k \geq \psi(G)$, то G можно покрасить в k цветов

Утверждение

$\psi(G) \leq$max deg v+1



это степени вершины $\psi(G) = 5 + 1$ max deg=5

Доказательство

Индукция по количеству вершин

$n = 0$
$n = 1$ верно

max deg =0

$\psi(G) \leq 1$

П G v- вершина max deg, уберем её, получим G'=G deg v

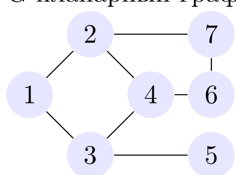
max deg G'$\geq$ max deg G=$\triangle$

раскрасим G' в A+1 цвет

цвет запрещены $\leq \triangle$ цветов => $\geq$ 1 цвет можно

Утверждение

G-планарный граф => ψ(G)$\leq$5



Докозательство

1)в G есть вершина степени $\leq$ 5 |V|=n

Если нет => deg V $\geq$ 6 => $\sum$ deg v$\geq$ 6n

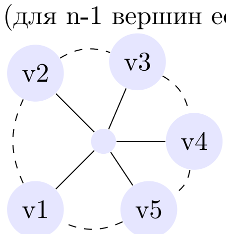
=> 2m$\geq$ 6n=> m$\geq$ 3nб но в планарном G m$\leq$ 3n-6

Утверждение

Раскрасим в 5 цветов по индуцкции

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

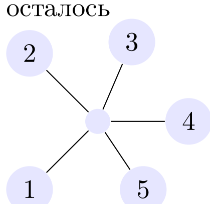
$\supset$ у нас n вершин (для n-1 вершин есть раскраска)

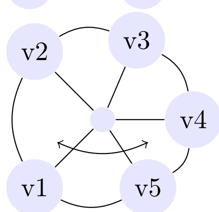


Берем v: deg v$\leq$ 5

раскрасим G' без v

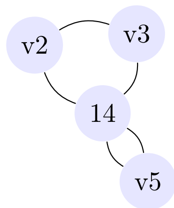
если соседи v_i исчезнут $\leq$u цветов => для V есть цвет осталось





стягиваем 1 граньбез V сделаем $\overline{G}$

$\overline{G}$



$\overline{G}$ -n-2 вершины => можно раскрасить

вернемся у G, a и d имеют один цвет => есть для V

если $v_i \quad v_j$ все соединены ребром => есть K_5=> не планарен

<u>Утверждение</u>

$\psi(G) \leq 4$(проблема 4х красок)

<u>Хроматические многочлены</u>

$\supset \quad \psi(G,K)$-это функция, сколько способов расрасить G в k цветов

$$\psi(0-0,k) = \begin{cases} k=0 & 0 \\ k=1 & 0 \\ k=2 & 2 \\ k=3 & 6 \\ k=4 & 12 \end{cases}$$

иначе k(k-1)

$\psi(0-0,k)$=k(k-1)

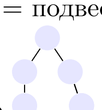
$\psi(0 \quad 0,k) = k^2$

<u>Утверждение</u>

1)$\psi(0_n,k) = k^n$ граф из n вершин без ребер

2)$\psi(K_n$(полный), k)=$\underbrace{k(k-2)(k-1)...(k-n+1) = k^n}_{\text{множители}}$

3)$\psi(T_n,k)$ = подвесим дерево $\forall$ вершину

T_n- дерево  k-1 цвет выполнен

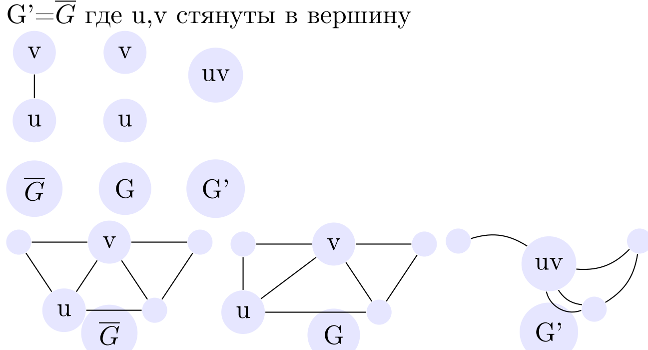
4) $\overline{G}$- граф

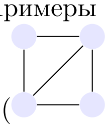
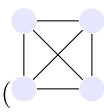
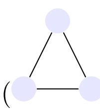
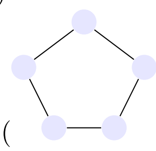
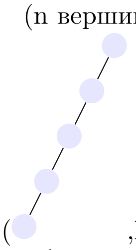
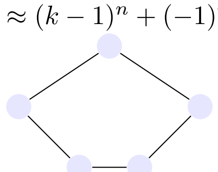
u,k вершины с ребром (u,v)

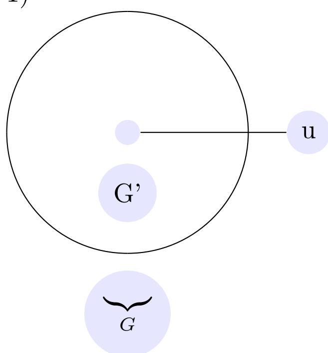
G=$\overline{G}$(u,v)

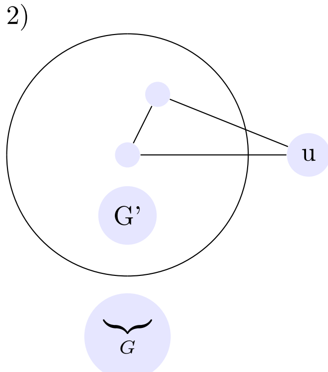
без ребра

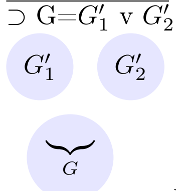
G'=$\overline{G}$ где u,v стянуты в вершину



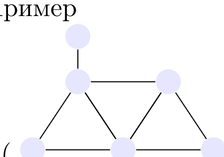
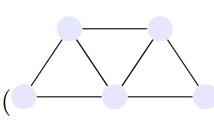
$\psi(G,k)=\psi(\overline{G},k)+\psi(G',k)$
способы раскрасить G, где u и v- разные цвета
способы
раскраска G, где u,v- один цвет
Следствие
$\psi(\overline{G},k)=\psi(G,k)-\psi(G',k)$
Примеры
$\psi($  ,k)=$\psi($  ,k)=$\psi($  ,k)=
=k(k-1)(k-2)(k-3)+k(k-1)(k-2)
5)
$\psi($  (n вершин),k)=?
$\psi(C_n,k)=\psi($  ,k)-$\psi(C_{n-1},k)=k(k-1)^{n-2}-\psi(C_{n-k},k)$
=>C_n=k(k-1)$^{n-1}-C_{n-1}=k(k-1)^{n-1}-k(k-1)^{n-2}+C_{n-2}=k(k-1)^{n-1}-k(k-1)^{n-2}+k(k-1)^{n-3}+$
$\pm k(k-1)^1+(-1)^{n-1}C \qquad c=\psi(0,k)=k$
геометрическая прогрессия
$q=(-1)^n k(k-1)\frac{q^n-1}{q-1}+(-1)^{n-1}k=(-1)^n k(k-1)\frac{(-1)^n(k-1)^{n-1}-1}{k-1}+(-1)^{n-1}k$
$\approx (k-1)^n+(-1)^n$

- C_5 цикл из 5 вершин
$\psi(C_n,K)$- несколько способов раскрасить C_n в к цветов
$\psi(T_n$(дерево),k)=$k(k-1)^{n-1}$
$\psi(K_n$(полный),k)=$k^n=k(k-1)...(k-n+1)$
$\psi(C_n,k)=k(k-1)^{n-1}-k(k-1)^{n-2}+k(k-1)^{n-3}..(-1)^n k(k-1)+(-1)^{n+1}k-C_1$
=геом прогрессия
начало:$(-1)^{n+1}k$
множитель:$-(k-1)$
членов:n штук
S=$(-1)^{n+1}k\frac{q^{n+1}-1}{q-1}$
$=(-1)^{n+1}k\frac{(-1)^k(k-1)^n-1}{-k}=(-1)^n((-1)^n(k-1)^n-1)=(k-1)^n-(-1)^n$

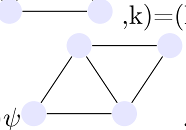
Утверждение
$\supset$G имеет висячую вершину и
1)


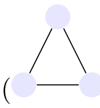
$\psi(G,K)=\psi(G',k)x(k-1)$- расчитали
2)


u соединены с v_1 и v_2 $(v_1,v_2)\in G$ $\psi(G,k)$
$=\psi(G',k)x(k-2)$
Утверждение
$\supset$ G=G_1' v G_2' нет ребер между G' и G_2'


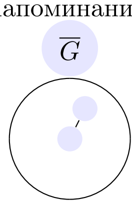
нет ребер $\psi(G,k)=\psi(G_1',k)\psi(G_2',k)$
Пример

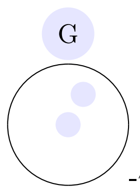
$\psi($  ,k)=(k-1)$\psi($  ,k)=

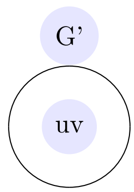
=(k-1)(k-2)ψ  ,k)=

=(k-1)$(k-2)^2\psi($)=(k-1)$(k-2)^2\cdot k(k-1)(k-2)$

Напоминание

ψ 

$=\psi$ 

-ψ 

Утверждение

$\psi(G,k)$- это многочлен

1)старший коэффициент=1

2)степень= n(количество вершин)

3)знаки чередуются

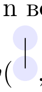
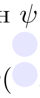
4)младший коэффициент=0

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

Докозательство

Индукция по количеству вершин при равном количестве вершин: количество ребер

База. пустой граф ищ n вершин $\psi(::,k)=k^n\pm 0k^{n-1}$

переход $\overline{G}$ с ребером $\psi($,k)=$\psi($,k)(*1)-$\psi($,k)(*2)

*1-мало ребер, количество вершин

*2-мало вершин

работает индукционное предположение

1) ст. коэффициент $(1k^n)-(k^{n-1})$

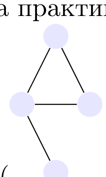
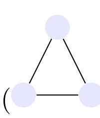
2)степень=n

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

4)младший коэффициент

5)ребер Gk^{n-1}=-(количество ребер G+1)k^{n-1}=-ребер Gk^{n-1}

на практике

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

Раскроем скобки

$k^4-4k^3+5k^2-2k$

Утверждение

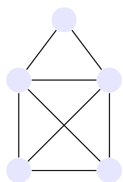
$\psi(G)$-хроматическое число(минмальное число цветов для раскраски)

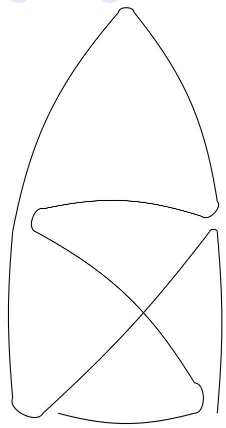
$\psi(G,k)$ k=0,1,2,...$\psi(G)-1$- корни многочлена

$\psi(G)$- не корень

Эйлеровы графы

Нарисовать не проводя дважды по одному ребру





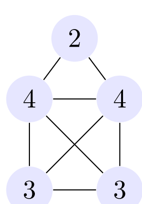
Определение

Эейлеров путь- цикл, содержащий все ребра не проходящие дважды по ребру

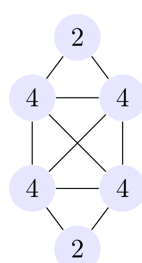
Утверждение

G содержит Эйлеров цикл <=> G связан и все степени вершин четные deg v-чет $\forall v \in V$

Пример



- нет цикла



- есть Эйлеров цикл

Докозательство

представим граф имеющий одну веришну с длинным ребром пересекющимся самого себя, но входящим и исходящим из одной вершины- такой граф связен.

Количество входов= количество выходов=> deg четн.

В каждой вершине по пути

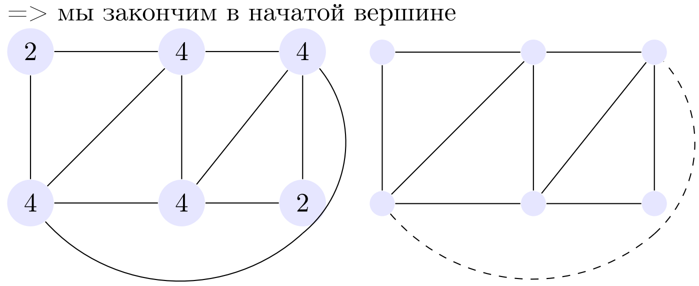
Использовано чет. ребер(к вход, к выходу)

+1 реберо, через которые вошли

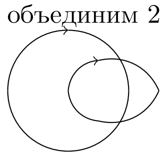
=> использовали нечет ребер

=> есть еще одно, по нему можно уйти, кроме начальной, из неё вышли на 1 раз больше

=> мы закончим в начатой вершине



граф с 1 вершиной и ребром входящим и исходящим из неё
Построена часть в остатке, все степени четные, т.к. G связен из начальной вершины x можно попасть в $\forall$ вершину и ребро
Повторим процесс из v$\in$ 1 циклу, из которой ведет новое ребро
объединим 2 цикла

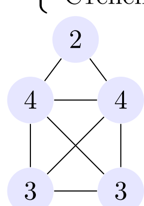


Продолжаем пока все ребране объединятся в 1 цикл

Теорема

G содержит Эйлеров путь <=>
1)связен

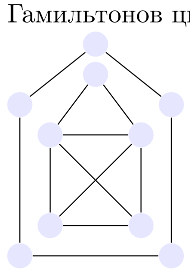
$$2)\begin{cases}\text{Степени всеъ вершин четны}\\ \text{Степени всех вершин кроме 2ух- чет}\end{cases}$$

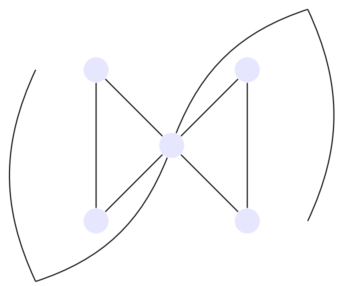


граф с двумя вершинами и ребром выходящим из одной и входящей в другую
В этом случае нечет вершины- это начало и конец

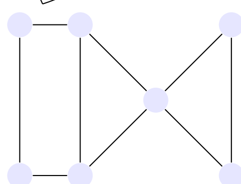
Определение

Гамильтонов цилы- полсьые цепи/циклыпо всем вершинам





Г. путь



нет Г. пути

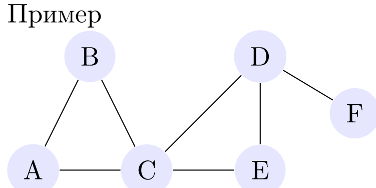
В прошлый раз(по всем вершинам) гамильтоновы пути
(по всем ребрам) эйлеровы циклы

Длины путей в графе

Определение

Длина пути в графе- количество ребер в пути

Пример



ABCDF- путь от A до F- длина 4(4 ребра)
ACEDF- длина 4
ACDF-длина 3
ABCEDF- длина 5

Определение

Расстояние между вершинами- минимальная длина пути между вершинами или $+\infty$, если пути нет

Обозначение d(x,y)- растояние от X до Y

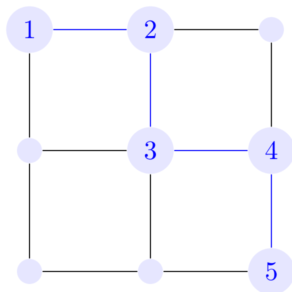
Пример d(A,F)=3

Определение

Диаметр графа- минимальное расстояние между вершинами графа

Пример

В примере выше для него =3(достигается на др)



-диаметр 4

Все другие расстояние ≤ 4

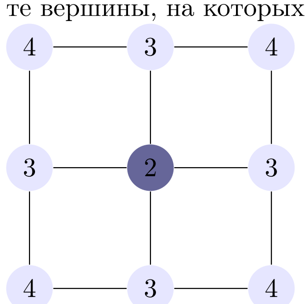
Определение Для каждой вершины графа G=(V,E) можно посчитать max расстояние до других вершин

r(v)=maxd(v,s)| s∈V

Радиус

r(G)=minr(v)|v∈ V

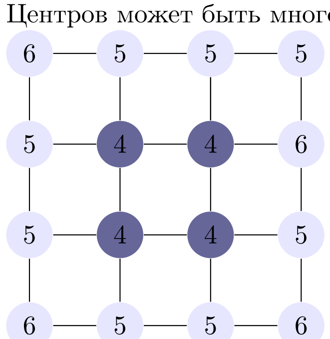
те вершины, на которых достигается min- это центр



2-центр

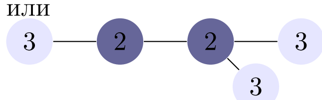
r(6)=2-радиус графа

Центров может быть много



-4 центра r(6)=4

или

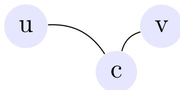


r(G)=2 2 центра

Утверждение в G=(V,E) d(G)$\leq$ 2 r(G)

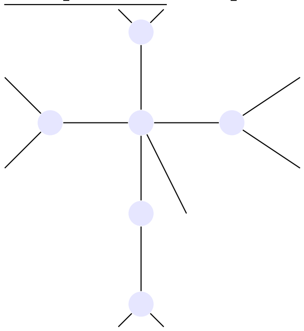
Докозательство

$\supset$ c-центр графа u,v$\in$ V

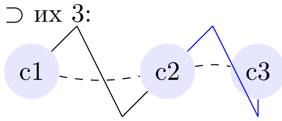


d(c,u)≤ r
d(c,v)≤ r
=> d(u,v)≤ 2r=> d(G)= max d(u,v)≤ 2r

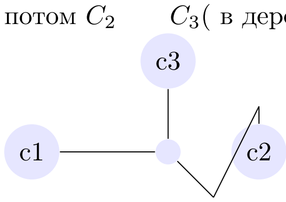
<ins>Утверждение</ins> В дереве ≤ 2 центров



⊃ их 3:



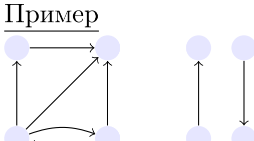
Построим пути между C_1 C_2
потом C_2 C_3(в дереве ровно 1 путь между вершинами)



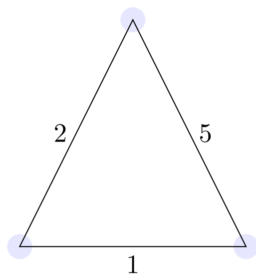
$r(c_0) < r(c_1) = r(c_2) = r(c_3) = r(G) = r$

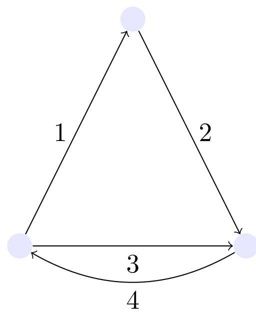
<ins>Замечание</ins>
Будем далье иногда использовать ориентированиые графы G=(V,E)
(ребра в ориентированном графе иногда называют дугами)
E ⊂(u,v)-упорядоченная пара

<ins>Пример</ins>

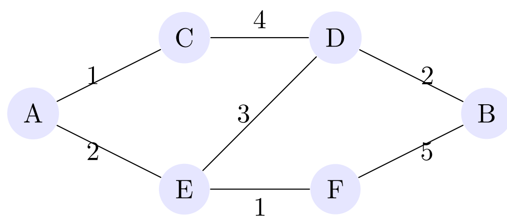


<ins>Замечание</ins> У рёбер будут весы
G=(V,E) вес- это f:E → R
то есть у числа каждому ребру

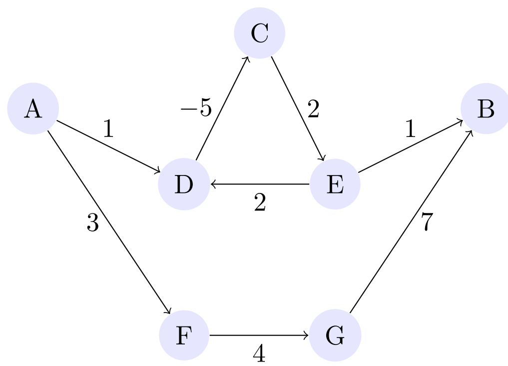




Расстояние на графе с весами считается как
min $\sum$ весов по всем путям



d(A,B)=?
d(ACDB)=1+4+2=7
d(ACDEFB)=1+4+3+1+5=14
d(AEFB)=2+1+5=8
d(AEDB)=2+3+2=7
min=7
=> d(A,B)=7
<u>Замечание</u> расстояние во взвешенном графе не всегда существует



d(F,G)=4

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

d(A,B)=?

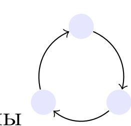
d(ADCEB)=1-5+2+1=-1

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

и т.д. min=-∞

<u>Утверждение</u> В графе есть все расстояния <=> в графе нет цикла отрицательной длины

<u>Докозательство</u>

Если есть цикл <0 => $\forall$ две вершины  этого цикла не имеют расстояние(или $\rightarrow \infty$

Если нет расстояния, то есть для u,v есть пути сколь угодно малеикие $\supset$ есть путь длинее n=|v| ребер => повтор ведущих в пути

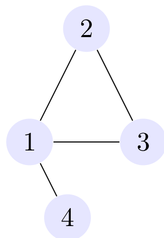
- это и будет отрицтальный цикл

Как хранить графы в компьютере (представление графа в компьютере)

1. Матрица смежности: таблица ведущих вершин

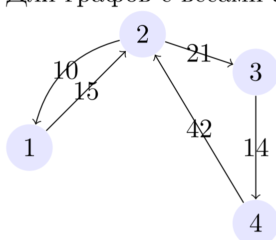
a(i,j)=$\begin{cases} 0\text{если нет ребра} \\ 1\text{если есть ребро} \end{cases}$

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



-симметичная для неориентированного 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
3	$+\infty$	42	$+\infty$	$+\infty$

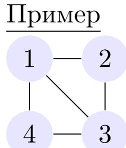
объем памяти $n^2 = |V|^2$

2. Списки смежности

- для каждой вершины хранит список соседей

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

Пример



1:234

2:13

3:24

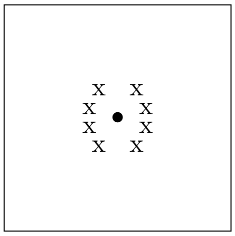
4:13

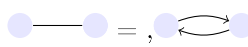
Память $\approx$ |E| количество ребер

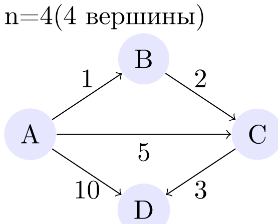
3.Неявные способы умеем вычислять всех соседей $\forall$ вершины

Пример

Задача обход конем шахматной доски

 граф: вершины=клета 64 шт
ребра- вершины перел ходом коня
Можно для $\forall$ клетки(вершины) посчитать, куда можно попасть
Задача обхода конем= гамильтонов 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 то де самое, что иметь путь от r до всех вершин.
Алгоритмы Форда-Беллмана
Дано G=(V,E)
u$\in$ V, найти расстояния 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($v_1 v_2$)<d(v_2)=> d(v_2)=d(v_1)+f($v_1 v_2$)
Алгоритм: Повторить n-1 раз: перебрать все ребра e и каждое релексировать
(в неориентированом графе  то есть две релаксации на ребро)
Пример
n=4(4 вершины)

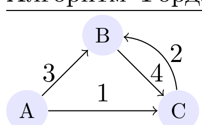


$, \psi$

ребра: A:B(1)C(5)d(10) B:C(2) C:D(3) D:
Шаг 1

	A	B	C	D
	0	$+\infty$	$+\infty$	$+\infty$
AB	0	1	$+\infty$	$+\infty$
AC	0	0	5	$+\infty$
AD	0	1	5	10
AB	0	1	3	10
CD	0	1	3	6

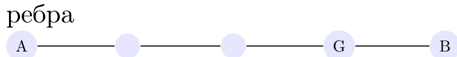
Шаг 2
AD—//— C->D 3+3<10
AC—//—
AD—//—
AB—//—
CD—//—
Шаг 3
AD—//—
AC—//—
AD—//—
AB—//—
CD—//—
ответ: d(A)=0 d(B)=1 d(C)=3 d(D)=6
Время работы $\approx$ |V|*|E|$\leq |V|^3$.
<u>Алгоритм Форда-Беллмана</u>

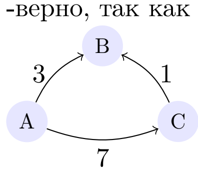


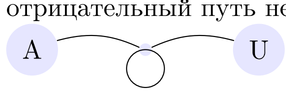
A:3B, 1C
B:4C
C:2B
Путь из A

	A	B	C
	0	∞	∞
сначала d:	0	3	∞
	0	3	1
	0	2	1

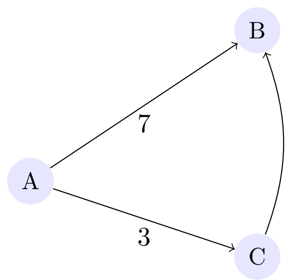
Релаксируем
A $\xrightarrow{3}$B
0 $\quad\infty$
0+3<∞
A $\xrightarrow{1}$C
0+1<∞
B $\xrightarrow{4}$C
3 $\quad$ 1
3+4<1
C $\xrightarrow{3}$B

1+1<3
n=1=>n-1=2 раза цикл релаксирующий
AB,AC,BC,CB= нет улушений ABC
Корректность алгоритма
<u>Теорема</u>
В конце массив d содержит расстояния от A
<u>Докозательство</u>
После i-го цикла релаксации всех ребер, d хранит числа d(v)$\leq$ min длин путей, в которых $\leq$ i ребер
Дейститвтельно, i=0(База индукции)
min (по пути из 0 ребер) только A-A d(A)=0 d(u)=∞
$\supset$ есть оптимальный путь из i+1
ребра


от A до B i+1 ребер
от A до G i ребер
По предположении d(C)= dB + (A,C)
длина пути $\underbrace{A-C-B}_{d(c)=dist(c)}$=dB +(C)+вес(CB)
проверка
d(C)+вес(CB)$\leq$ d(B)
-верно, так как путь A-C-B оптимален => d(B)=d(C)+вес (CB)


Почем путь n-1 этап?
отрицательный путь не содержит цикл


=> -//- $\in$ n-1 ребро
<u>Замечания</u>
Мы вычисляем только расстояния, но путь неизвестен
Как восстановить путь?
Будем сохранять информацию об успешных релаксациях
prev-массив вершин
Если релаксация u v успешна, то Prev[v]=u оптимальный путь в v лежит через u

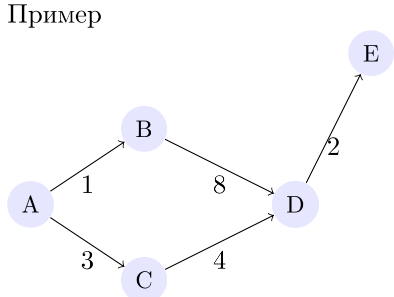


d	A	B	C
AB	0	∞	∞
AC	0	3/A	1/A
CB	0	2/C	1/A

восстановить путь в B A→C→B
A=prev(C) C=prev(B)

<u>Алгоритм Дейкстры</u>

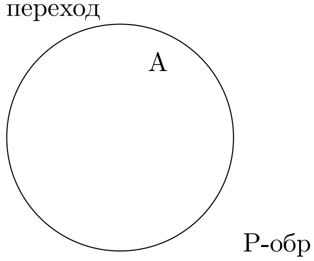
В отличие от ФБ требует, чтобы веса w(e)$\geq$0
Алгоритм
Дан граф G=(V,E) A$\in$ V
Найти расстояния до всех вершин d(u)=dist(A,u)
Алгоритм p=0 d(A)=0 d(u$\neq$A)=∞- обаботанные вершины
for n раз (n=v)
выбрать u$\in$ V / p, где d(u) $\to$ min (из необработ. min d)
for e$\in$ ребра из U, e=(u,v)
рефлексируем ребро e
p=p $\cup$ u
Пример



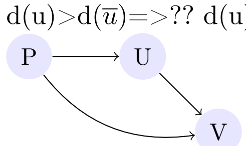
d	A	B	C	D	E
	0	∞	∞	∞	∞
		1	3	∞	∞
			3	9	8
				7	∞
					9

U=A
$A \xrightarrow[1]{} B$
0 $\quad\infty$
$A \xrightarrow[3]{} C$
0 $\quad\infty$
U=B

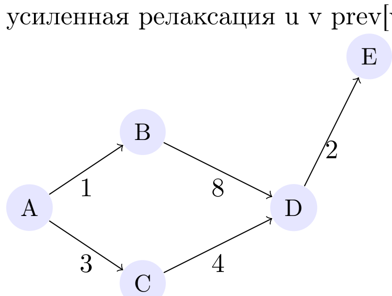
B $\underset{8}{\to}$D
1 ∞
U=C
C $\underset{4}{\to}$D
3 9
4=D
Эффективность |V|x|E|xlog|V|
log|V|- выбор мин
Корректность
идея- на каждом шаге d(u)=min путей
База шаг=0 d(A)=0 d(U)=∞
переход



Выбрали u=min вершин из V/{ p}
$\supset$ есть оптимальный путь в u $\underbrace{A---\overline{u}}_{\text{обр}}$—u
dist($\overline{u}$)=dist(u)-x
По предложению dist($\overline{u}$)=d($\overline{u}$)
d(u)>d($\overline{u}$)=>?? d(u) был min



путь через P
$\supset$ оптимальный путь в v идет через u
dist(A,U)+w(u,v)=dist(A,v)
=> релаксация u $\to$v успешна и d(v) получит оптимальное расстояние
для восстановления пути нужен аналогичный prev
усиленная релаксация u v prev[v]=u



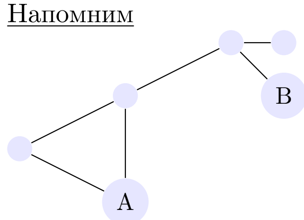
A	B	C	D	E
0	∞	∞	∞	∞
	1/A	3/A	∞	∞
		3/A	9/D	8
			7/C	∞
				9/D

A→C→D→E

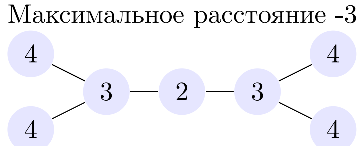
Утверждение

В дереве 1≤ центров ≤2

Напомним

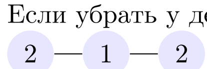


Максимальное расстояние -3



Доказательство

Если убрать у дерева все висячие вершины, расстояние изменяется на 1



Если повторять убирания висячих

или центры

Алгоритм Флойда

Дан граф G=(V,E)

Вернуть d(u,v)

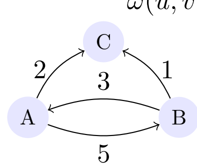
u\ c	v
u	d(u;v)

Алгоритм

составляем матрицу d_0=

d_0:d_0(u,v)=0

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



d_0

	A	B	C
A	0	5	2
B	3	0	∞
C	∞	1	0

for$k \in$ V
for $v \in V$
if d(u,v)> d(u,k)+d(k,v)
=> d(u,v)=-//-
Пример
k=A

	A	B	C
A	0	5	2
B	3	0	5
C	∞	1	0

BC>BA+AB

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

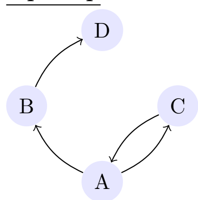
CA>CB+BC

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

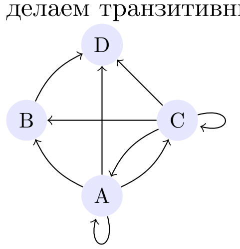
Корректность
Утверждается
после шага k в d(u,v) min d(пути) u- от 1 до k -v
База k=0
d(u,v)=u- нет -v
действительно, в начале d содержит длины ребер, переход u- содержит от 1 до k+1 -v
$\supset$ есть оптимальный путь из u- k+1 -v $\rightarrow$
1)$\rightarrow$ в нем нет k+1 => его длина d(u,v)
2)$\rightarrow$ есть k+1 u- O-k+1-O -v
его длина d(u,k+1) +d(k+1,v)
это ровно проверка цикла, меньший вариант ззанимается в d, в конце
d(u,v)= min(u $\forall$ вершины v)= dist(u,v)
Замечание
Чтобы восстанавливать путь, можно ввести массив through
if d(u,v)>d(u,k)+d(k,v) => d(u,v)= -//- through (u,v)=k
Для восстановления пути
1) A... th(A,i)...I...th(I,B)...B
2)и так далее, если th(x,y) нет, запишем => реберо xy это оптимальный путь
Замечание
В алгоритме Флойда идет транзитивное замыкание бинарных отношений

⊃ R-бинарное отношение на M
⊃ $\overline{R}$- транзитивное замыкание R, если
1) $\overline{R}$ ⊃ R
2) $\overline{R}$-транзитивно
3)∀ $\overline{R}$ $\quad \overline{R} \supset \overline{\overline{R}} \supset R$ не транзитивно

<u>Пример</u>



не транзитивно
aRb, bRd, но ~~aRd~~
aRc, cRa, но ~~aRd~~
делаем транзитивным



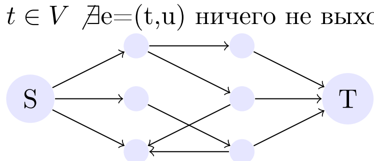
⊃ R-бинарно отношение ⊃ G=(M,R) граф отношения,
тогда $\overline{R}$- это $x\overline{R}y$ <=> есть путь x-y в G

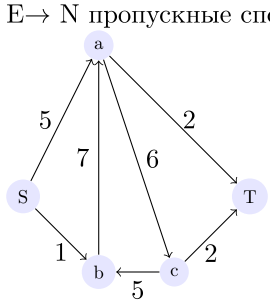
<u>Докозательство</u>

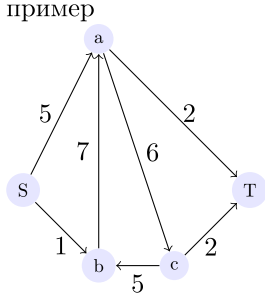
1)$\overline{R}$ ⊃R так как xRy=> есть путь из 1 ребра => $x\overline{R}y$
x-y 2)$\overline{R}$- транзитивно, т.к. x$\overline{R}$y, y$\overline{R}$z => x$\overline{R}$z
3) ⊃ $\overline{\overline{R}} \supset R$, $\overline{\overline{R}}$- транзитивно
⊃ есть путь x в y → x_1 → x_2 → x_3 → y
xRx_1 => $x_1\overline{\overline{R}}x_2$=> x$\overline{\overline{R}}x_2$
$x_2\overline{\overline{R}}x_3$=> x$\overline{\overline{R}}x_3$=>...=> x$\overline{\overline{R}}$y=> $\overline{\overline{R}} \supset \overline{R} \supset R$
Применим алгоритм Флойда к графу G=(M,R)
$d_0(x,y) = 1$, если xRy
$d_0(x,y) = \infty$, если ~~xRy~~
После конца алгоритма

	y
x	∞
	7

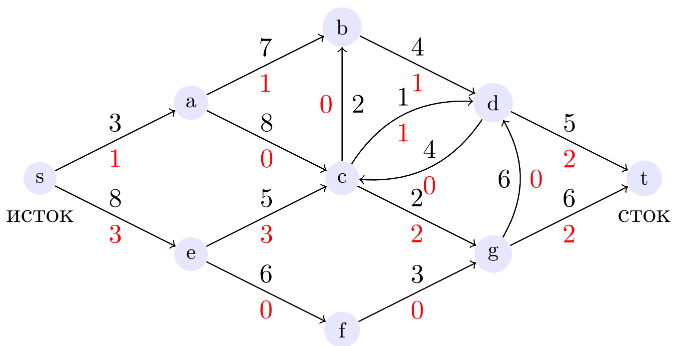
замыкание x$\overline{R}$y, если d(x,y)<∞
Алгоритм транзитивного замыкания
$\overline{R}$=R
for k∈ M

for x∈ M
for y ∈ M
if xRk & kRy
$\overline{\overline{R}} \in (x, y)$, то есть сделать x$\overline{R}$y
Потоки в сетях
Определение
Сеть- граф G=(V,E)
ориентированный
$s \in V$ $\nexists$e=(4,s) ничего не входит
$t \in V$ $\nexists$e=(t,u) ничего не выходит


E→ N пропускные способности ребер целые>0


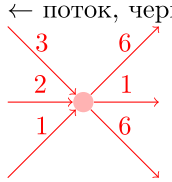
Определение
Поток f в сеть G-это
f: E→ **R** 1) 0≤f(e)≤ c(e)
2) $\forall u \neq S, t$
$\sum\limits_{l=(v,u)\in E}$ f(e)
пример


Поток в сетях
Пример



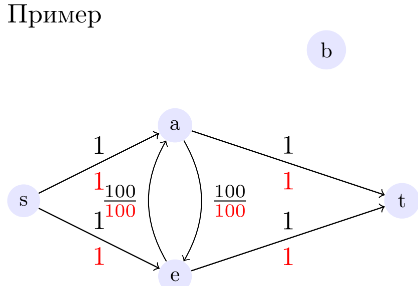
← поток, красные f

← поток, черные c



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

Пример



-поток корректный

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

<u>Теорема</u>

Дана сеть (G=(V,E),c),поток f на G

Тогда

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

Рассмтрим

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

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

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

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

вытекает $+ \sum\limits_{v\in V\{s,\}} \sum\limits_{e:e=(u,v)} f(e)$-втекает- O=

вытекает-втекает$+\sum\limits_{e\in E} f(e) =$
$\Rightarrow$ вытекает- втекает $\Rightarrow$ вытекает=втекает
$\sum\limits_{e:e=(u,t)} f(e) = \sum\limits_{e:e=(s,u)} f(e)$

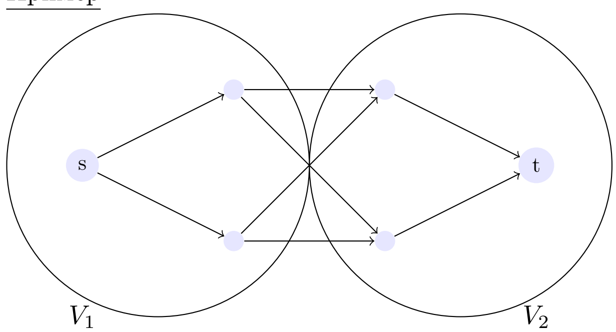
Определение

w(f)- эта величина называется вершиной потока

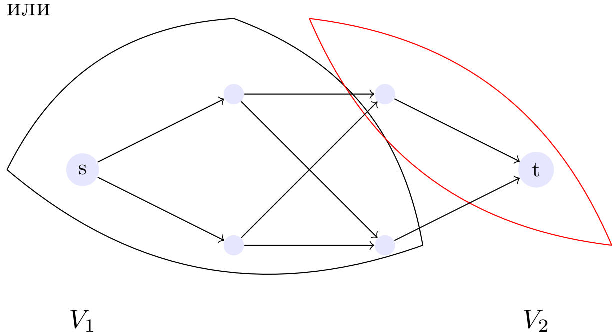
Определение

Разрез в сети (G=(V,E),c) Разрез G=(V_1, V_2)
$s \in V_1 \quad t \in V_2 \quad V_1 \cap V_2 = \emptyset \quad V_1 \cup V_2 = V$

Пример



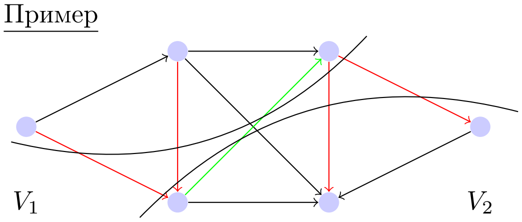
или



Определение

E_c-ребра разреза это все рёбра, которые идут из V_1 в V_2 или наоборот
E_c^+-прямые ребра разреха (из V_1 в V_2)
E_c^+-обратные ребра разреха (из V_2 в V_1)

Пример



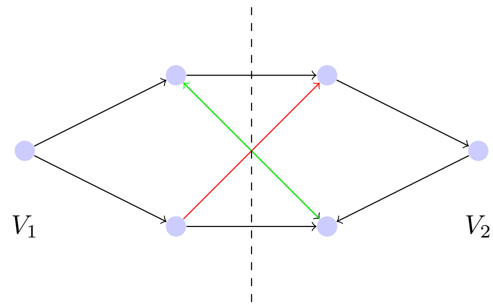
оobратное E_c^-
прямое E_c^+
$E_c = E_c^- \cap E_c^+$

Определение

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

Обозначение c(G)

Например



C=(V_1, V_2)

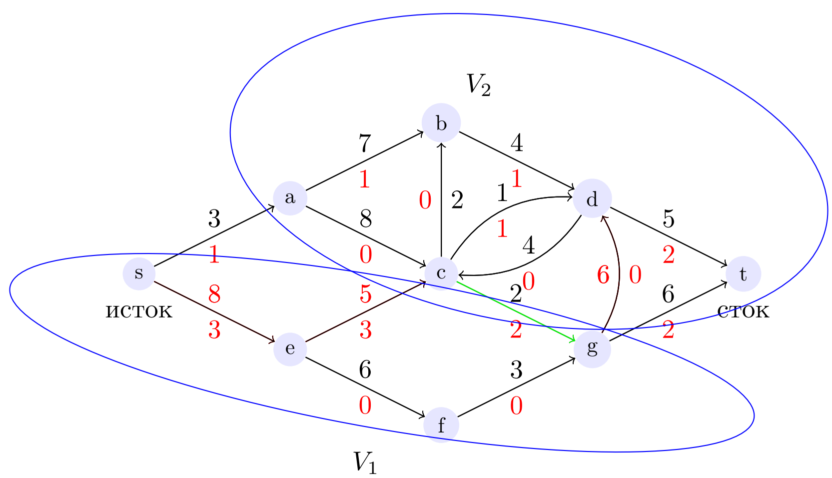
c(C)=8+3+4=15

Утверждение

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

тогда w(f)= $\sum\limits_{e \in E_c^+} f(e) - \sum\limits_{e \in E_c^-} f(e)$

Пример



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

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

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

Да! 4=6-2

посчитаем сумму

$\sum\limits_{v \in V_1} (- \sum\limits_{e:e=(u,v)} f(e) + \sum\limits_{e:e=(v),u} f(e)$

1) для $\forall v \in V_1 \setminus \{s\}$внутреняя $\sum - \sum = 0$
для v=s получается w(f)= $\sum\limits_{e:e=(v),u} f(e)$

2) $\sum\limits_{e=(v),u} (f(e) - f(e)) + \underbrace{\sum_{e\in E_c^+} f(e) - \sum_{e\in E_c^-} +[-f(e)]}_{\text{смотри условие}}$=0+величина их условия

Обозначение

w(c,f)-выличина потока через разрез

$\sum\limits_{e\in E_c^+} f(e) - \sum\limits_{e\in E_c^-} f(e)$

Замечание

$\forall C$ w(f)=w(C,f)- по Теореме

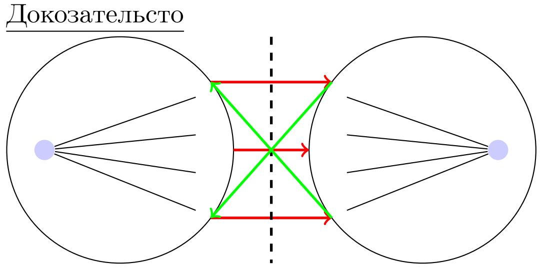
Замечание

Будем решать задачу о максимальном потоке в сети то есть найти f:
w(f)→max

Утверждение

Дано G,c-сеть, C-разрез
Тогда w(f) ≥c(C)

Докозательсто



V_1 V_2

w(f)=w(C,f)= $\sum\limits_{e\in E_c^+} f(e) - \sum\limits_{e\in E_c^-} f(e) \leq$
$\sum\limits_{e\in E_c^+} f(e) \leq \sum\limits_{e\in E_c^+} c(e) = c(C) \Rightarrow$ w(f)≤c(C)

Следствие

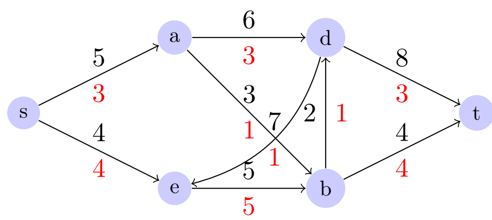
В сети G w(f$_{max}$)≤c(C$_{min}$)
где w(f$_{max}$)=max w(f) f-поток c(C_{min})=min c(C) C-разрез

Теорема Форда-Фалкерсона

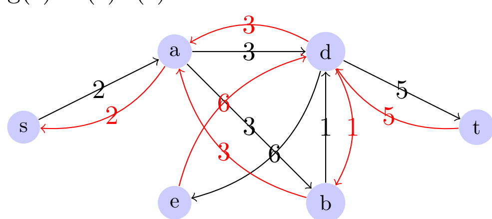
w(f$_{max}$)=c(C$_{min}$)
в сети (G,c) c(e)∈N
- для простоты считаем, что пропускные способности целые

Определение

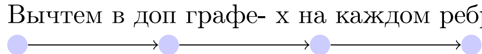
дополнительный граф для потока
$\overline{G}$имеет $\overline{V}$=V
$\overline{E}$:



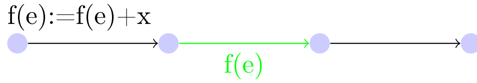
если f(e)<c(e)
e=(u,v)
то есть e'=(e',v')
g(e)=c(e)-f(e)



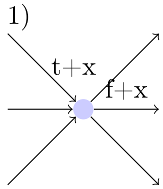
если 0<f(e) e=(u,v)
то есть e"=(v',u')
g(e")-f(e)
Докозательство
Начнем с нулевого потока и будем его постепенно увеличивать
Построим дополнительный граф $\overline{G}$ и найдем в нём путь из s в t
s-O-t
Найдем min g(e) на этом пути $\supset$ это x
Вычтем в доп графе- x на каждом ребре



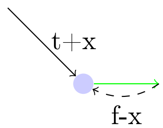
1)c(e)- f(e) $\rightarrow$ c(e)- f(e)-x
f(e):=f(e)+x



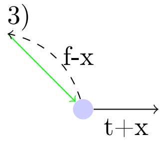
2)f(e)-x f'(e):=f(e)-x
Поймем, что 1) новый поток f' остался потое
2) величина потока увеличилась на x
Проверяем, что это поток $0 \leq f'(e) \leq c(e)$ уменьшаем по обратному, увеличиваем по прямому $c(e)-(f(e)+x) \geq 0$
В вершинах верно $\sum$вход=$\sum$исх
1)



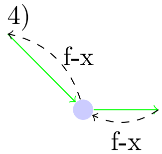
2)



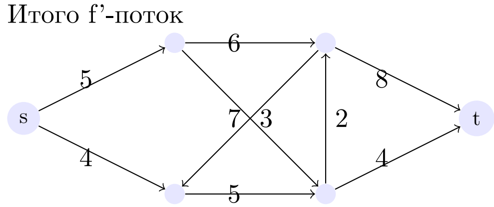
3)



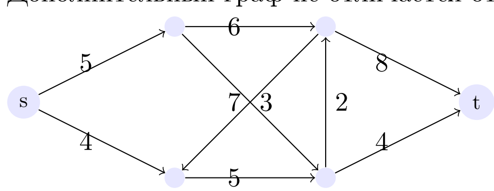
4)



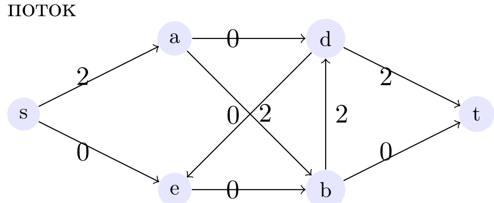
Итого f'-поток

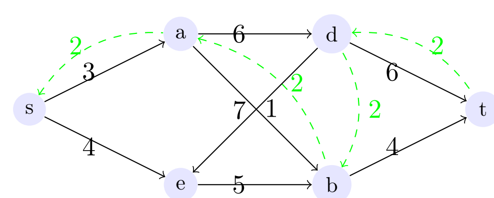


можно построить дополнительный граф
Пример. строим пути
сначала поток=0
Дополнительный граф не отличается от исходного

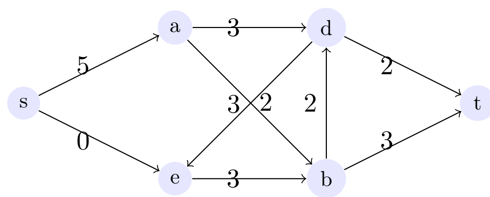


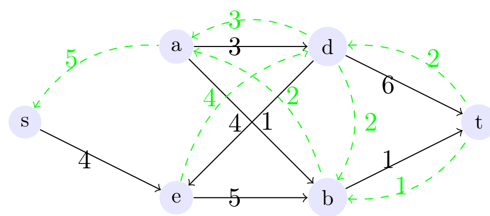
путь из s в t
S-(5)-a-(3)-d-(2)-b-(8)-t
добавим к поток +2 на каждое из этих ребер
поток



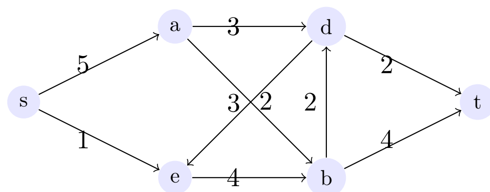


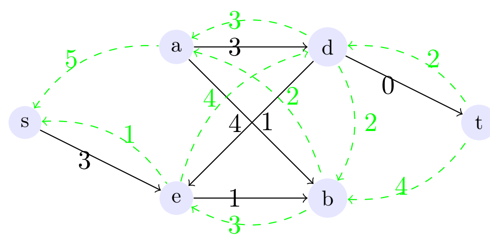
S-(3)-a-(6)-b-(7)-c-(5)-d-(4)-t
min=3



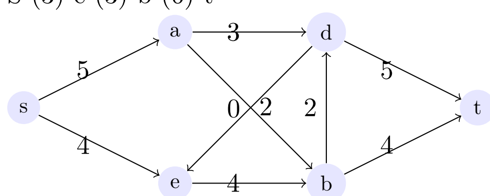


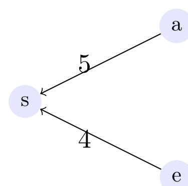
s-(4)-c-(2)-d-(1)-t
min=1



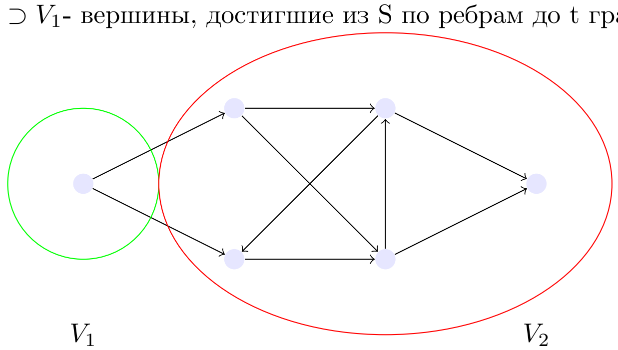


S-(3)-c-(3)-b-(6)-t



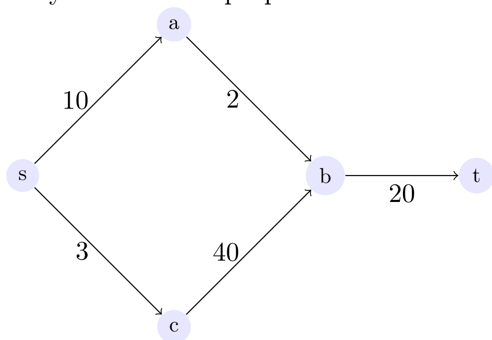


Продолжаем докозательство теорема Форда-Фалкерсона
Если пути нет, то поток оптимальный?
$\supset V_1$- вершины, достигшие из S по ребрам до t графа V_2=V \ V$_1$



$t \in V_2$, так как нет пути S→t получаем разрез исходной сети ребра E_G^+
Ребер нет в дополнительном графе, в дополнительном графе веса=0
c(e)-f(e)=0
c(C)= $\sum\limits_{e \in E_C^+} c(e) = \sum\limits_{e \in E_C^+} f(e)$=c(f)
Течет ли что-то по нечетным? Нет, иначе V_1 неверен
В прошлый раз мы напоминали Vc- разрез $\forall f$-поток
c(C)≥c(f)

получается c-min разрз f-max поток



c(C)=2+3=5

min разрез

Утверждение

Если каждый раз искать путь с min количеством ребер,

то время max поток $\sim V^2E$

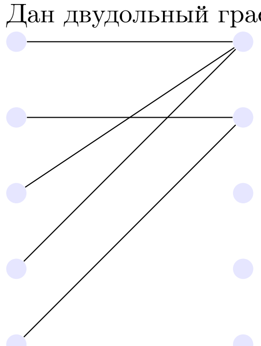
Без докозательства

Утверждение

Для плоской сети(без пересечений ребер)=c(f) эффективно искать верхние пути

Задачи о паросочетаниях

Дан двудольный граф G=(u,v)



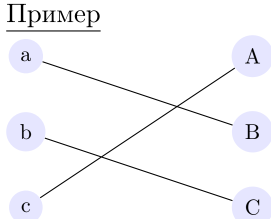
Определение

Паросочетания в G-это P$\in$ E, где ребра из P не имеют общие вершины

Определение

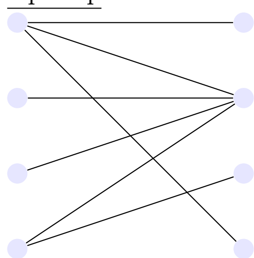
Максимальное паросочетание, это P$\subset$E, |P|$\rightarrow$ max из возможных

Пример



D{cA,bC,aB }

Пример



4-нельзя

Сводим к задаче о потоке

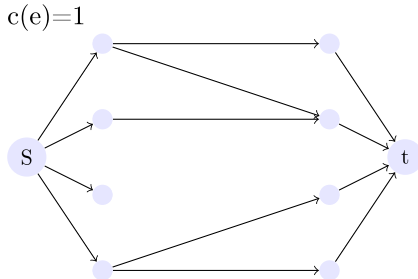
Ребра из S→u

v→t

u→v

слева направо

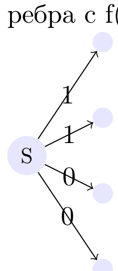
c(e)=1



Утверждение

каждому потоку (из f=0,1) соотвествует паросочетание

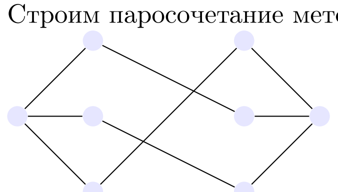
ребра с f(c)=1 это ребра паросочетания⇒ поток



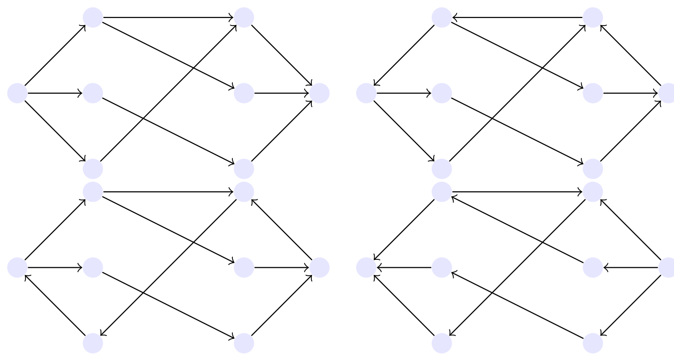
⇐ паросочетание соответвует потоку, где f(e)=1 для ребер паросочетания

Следствие Размер max паросочетаний=размер max потока

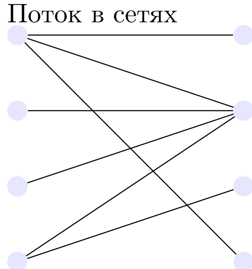
Строим паросочетание методом ФФ

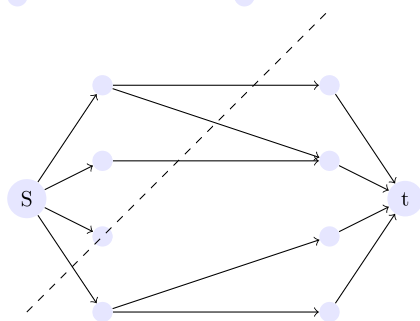


Строим дополнительный граф, но без чисел



Поток в сетях





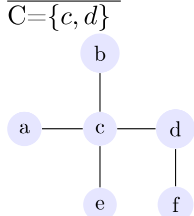
Задача о максимальном контролируемом множестве

<u>Определение</u>

$\supset$ G=(V,E) C$\subset$V- контолирующее множество, если $\forall$ G=(u,v)$\in$ E

<u>Примеры</u>

C=$\{c, d\}$

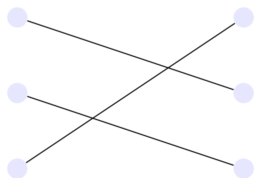


Замечание c=v -контролируемое множество

<u>Утверждение</u>

В двудольном графе G=(u$\vee$v,E) $\supset$ c-контролируемое множество
$\supset$ p-паросочетание, тогда |c|$\leq$|p|

<u>Доказательство</u>



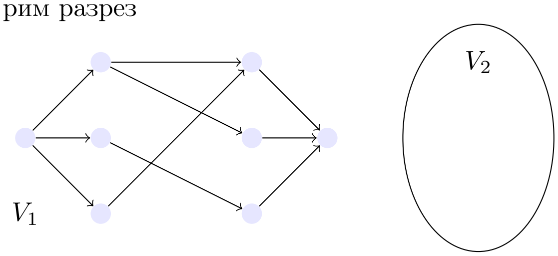
у каждого ребра с$\in$ p есть вершины u или v$\in$

G=(u$\vee$v,E)-двудольный граф

Размер максимального паросочетания= размеру min контролируемого множества

Доказательство

Построим максимальное паросочетание по Форду=Фалкерсон и рассмотрим разрез



|u|=x

|v|=y

|u$\wedge$v$_1$|=a

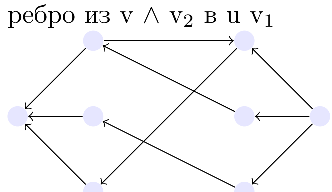
|u$\vee$v$_2$|=b

c($[v_1,v_2]$)-разрез=$\sum$1=е реберо исходное= (x-a)+b+n

Итого m=c((V_1,V_2))=x-a+b+n$\leq$m

x-a+b$\geq$0

Возьмем в качестве контролируещего множества c=(u$\vee$v$_1$) $\vee$ (v $\wedge$v$_1$) есть ребро из v $\wedge$ v$_2$ в u v$_1$



v $\wedge$ v$_2$ v $\wedge$ v$_1$

$\supset$ есть ребро e=(u,v)

1) щначит v$\subset$ v$_1$??

2) как понять в u? только из v?

Вывод 1 c=(u$\vee$v$_1$) $\vee$ (v $\wedge$v$_1$)-контролируемое множество

Вывод 2 c($[v_1,v_2]$)-разрез= $\sum\limits_{\substack{e\in u,v\\ u\in v_1\\ v\in v_1}}$ 1=(x-a)+0+b=|c|$\Rightarrow$|p|=|c|

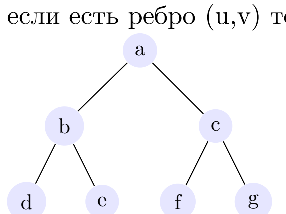
поиск в глубину, ширину

1)структура для хранения вершин данных

D-стек или очередь/ stack or queue
V→D положить в V в D
V∠D посмотреть
стек first in last out
очередь first in first out
<u>Пример</u>

Пример	стек	оередь
a→D	a	a
b→D	ba	ba
c→D	cba	cba
∠	c	a
D→	ba	cb
D→	b	b

Поиск в ширину (D очередь) или глубину (D стек) D∈ V_0 начальная вершина
пока D не пуст
u=∠D
если есть ребро (u,v) тогда v→ D

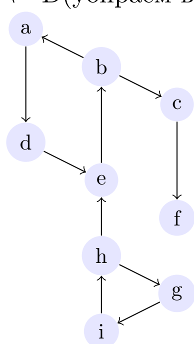


в глубину, иначе достать d→u

в глубину	в ширину
a	a
ba	ba
dba	cba
edba	cb
dba	dcb
a	edcb
ca	edc
fca	fedc
ca	gfedc
a	gfed
-	gfe
	gf
	g
	-

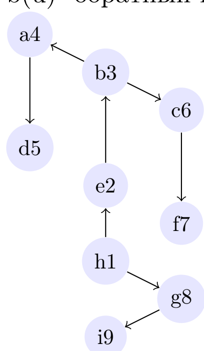
поиск в глубину/ ширину
D-стек/очередь
<u>Алгоритм поиска</u>
Дано: начальная вершина и D←u
used=0(обработанные, то есть были в D)

пока D$\neq$ 0
peek D(смотрим)
если есть ребро v-w, где w$\notin$used
иначе D$\leftarrow$ w_i used=used v{w}
$\leftarrow$ D(убираем вершину из D)

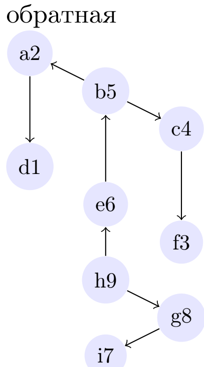


в глубину из h
h
he
heb
heba
hebad
heba
heb
hebc
hebcf
hebc
heb
he
h
hg
hgi
hg
h
0
в ширину
h
he
heg
eg
egb
gb
gbi
bi
bia
biac

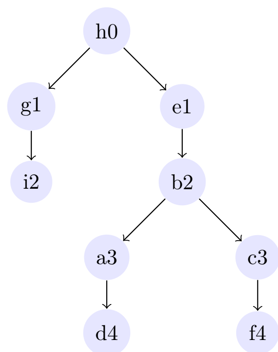
iac
ac
ac
acd
cd
d
0
Введем
n(u)-номер, како попался в D
b(u)- обратный номер, ккаой ушла из D



обратная



Замечание
При поиске в ширину n(u)=b(u)
Утверждение
Поиск в глубину перебирает вершины в том же порядке, что и алгоритм Дейкстры(веса ребер 1)
Действительно, добавление вершинв в D- это релаксация ребер v-w
удаление из D-убирание вершины с min расстоянием



3	3	3		
2	2	2	[]	[]
1	1	1	2	[]
1	*	1	2	[]
1	1	1	3	
2	2	2	4	

полный поиск в глубину

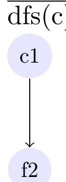
Пока есть непосещенная вершина u:

поиск в глубину(u)

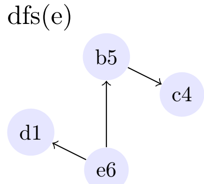
dfs- deep first search

Пример

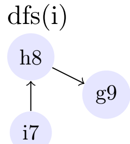
dfs(c)



dfs(e)



dfs(i)

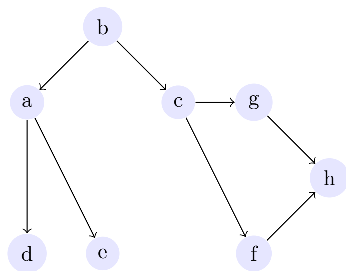


Утверждение

Пусть G-ориентированный граф без циклов

Пусть есть путь u→v

(нет пути v→u, т.к. нет циклов)



Доказательство

Делаем dfs

Куда попали раньше?

1)сначала попали в u

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

в стэк будет u...v$\Rightarrow$сначала из стэка уйдет v, потом u

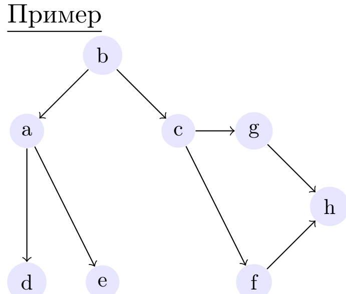
2)сначала в v$\Rightarrow$ значит закончим просмотр не попав в u$\Rightarrow$

номер b(c) присвоится раньше чем b(u)

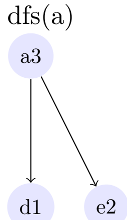
следствие алгоритма топологической сортировки

делаем каждый dfs и линейный порядок,зачтем как b(u)

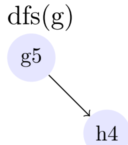
Пример



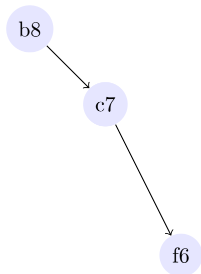
dfs(a)



dfs(g)



dfs(b)



Ответ: deahgfcb

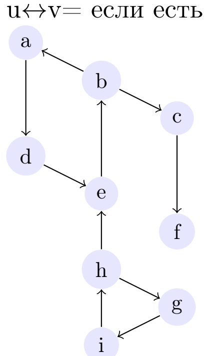
Компонент сильной связности

Напоминание

G-ориентированный граф

Введем отошение $\leftrightarrow$ на V

u$\leftrightarrow$v= если есть путь h$\rightarrow$ v и v$\leftrightarrow$h



abde, cf, hgi- компоненты сильной связности

c$\Leftrightarrow$f

a$\Leftrightarrow$e

e$\Leftrightarrow$h

~~b⇔e~~

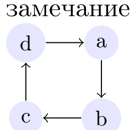
классы эквивалентности называются компоненты сильной связности

Опрделение

$\supset$G=(V,E)- ориентированный граф G^o=(v^o,E^o)-граф конденсации, если V^o=V/

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

замечание G^o не имеет циклов



$\Rightarrow$ у вершины $\leftrightarrow$

Утверждение

$\supset G = (V, E)$- ориентированный граф

G^o-граф конденсации G

делаем полный dfs в G

Тогда

Если в G^o есть путь из u^o в v^o, то b(u)>b(v) для любых u$\in u^o$ v$\in v^o$
max b(u)<max g(v)
u$\in u^o$ v$\in v^o$
<u>Докозательство</u>
Аналогично проломy утверждению
<u>Следствие</u>
Поиск компонент сильной связности
1)полный dfs в G
2)Находим u b(i)$\to$ max, делаем dfs по обратным ребрам G