Графы

Вариант 1

Содержание

[Задание 1 3](#_Toc133824132)

[Задание 2 3](#_Toc133824133)

[Задание 3 5](#_Toc133824134)

[Задание 4 5](#_Toc133824135)

[Задание 5 7](#_Toc133824136)

[Задание 6. 10](#_Toc133824137)

[Задание 7 12](#_Toc133824138)

[Задание 8 14](#_Toc133824139)

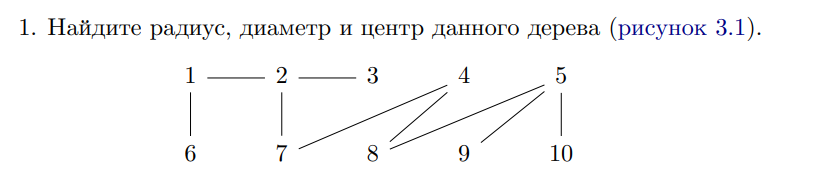
[Задание 9 20](#_Toc133824140)

[Задание 10 22](#_Toc133824141)

[Задание 11 28](#_Toc133824142)

[Задание 12. 33](#_Toc133824143)

# Задание 1



Решение.

Строим дистанционную матрицу D(u,v) и эксентриситет e(u)=maxvD(u,v) по дереву.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| vi vj | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | e(vi) |
| 1 | 0 | 1 | 2 | 3 | 5 | 1 | 2 | 4 | 6 | 6 | 6 |
| 2 | 1 | 0 | 1 | 2 | 4 | 2 | 2 | 3 | 5 | 5 | 5 |
| 3 | 2 | 1 | 0 | 2 | 5 | 3 | 2 | 4 | 6 | 6 | 6 |
| 4 | 3 | 2 | 2 | 0 | 2 | 4 | 1 | 1 | 3 | 3 | 4 |
| 5 | 5 | 4 | 5 | 2 | 0 | 6 | 3 | 1 | 1 | 1 | 6 |
| 6 | 1 | 2 | 3 | 4 | 6 | 0 | 2 | 4 | 7 | 7 | 7 |
| 7 | 2 | 2 | 2 | 1 | 3 | 2 | 0 | 2 | 4 | 4 | 4 |
| 8 | 4 | 3 | 4 | 1 | 1 | 4 | 2 | 0 | 2 | 2 | 4 |
| 9 | 6 | 5 | 6 | 3 | 1 | 7 | 4 | 2 | 0 | 1 | 7 |
| 10 | 6 | 5 | 6 | 3 | 1 | 7 | 4 | 2 | 1 | 0 | 7 |

Радиус дерева

r(T)=min(e(u))=min(6,5,6,4,6,7,4,4,7,7)=4;

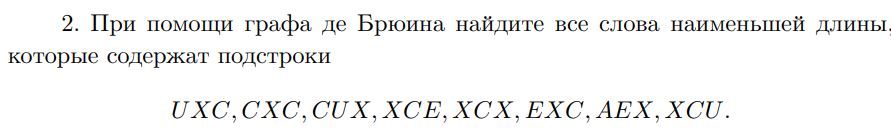
Диаметр графа

d(T)=max(e(u))=max(6,5,6,4,6,7,4,4,7,7)=7;

Центр дерева

C(T)={u:e(u)=r(T)=4}={4,7,8};

# Задание 2



Построим граф де Брюина.

UX

XC

CX

CU

CE

EX

AE

Построить минимальные пути, содержащие все дуги.

Таких пути 2:

1. AE->EX->XC->СX->XC->CU->UX->XC->CE
2. AE->EX->XC->CU->UX->XC->CX->XC->CE

Эти пути минимальны, так как каждую дугу графа содержат ровно 1 раз.

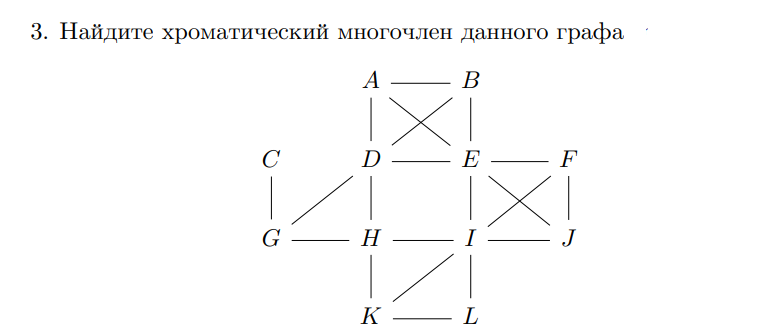
Других путей нет, т.к. начальный отрезок пути до вершины XC определяется однозначно, конечный отрезок из XC в CE также однозначно, а ориентированный цикл, содержащий вершину XC можно пройти двумя способами в двух направлениях.

Получаем два варианта кратчайшего слова

AEEXXCСXXCCUUXXCCE;

AEEXXCCUUXXCCXXCCE;

# Задание 3



Пусть используется t цветов. Найдем количество раскрасок центрального квадрата DEIH, для каждой правильной раскраски центра DEIH количество способов раскраски оставшейся части графа, очевидно, одинаково, т.е. оставшиеся части не связаны между собой.

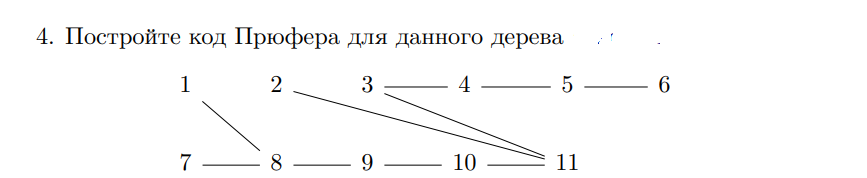
Вершины D и I можно раскрасить способами и для каждого такого способа вершины H и E можно докрасить одним и тем же количеством способом, итого центральный квадрат можно раскрасить способами. Для каждого способа раскраски центрального квадрата вершины A,B можно раскрасить (t-2)(t-3) способами, вершины F,J также (t-2)(t-3) способами, вершины K,L- (t-2)2 способами, Вершины C,G- (t-1)(t-2) способами.

Получаем (t-2)(t-3) (t-2)(t-3) (t-2)2 t-1)(t-2)=(t-1)∙(t-2)4∙(t-3)2 способов, окончательно получаем хроматический полином

P(G,t)= (t-1)∙(t-2)4∙(t-3)2 =

= (t-1)∙(t-2)6∙(t-3)2, ясно, что хроматическое число графа

# Задание 4

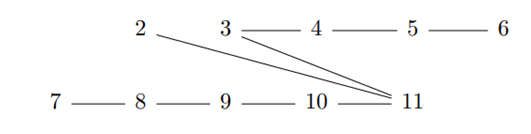


Код Прюфера строится следующим образом. Будем n-2 раза проделывать процедуру: выбираем лист дерева с наименьшим номером, удаляем его из дерева, и добавляем к коду Прюфера номер вершины, которая была связана с этим листом. В конце концов в дереве останется только 2 вершины, и алгоритм на этом завершается (номер этих вершин явным образом в коде не записываются).

Шаг 1.

Лист vmin=1, ребро (1,8), код =8

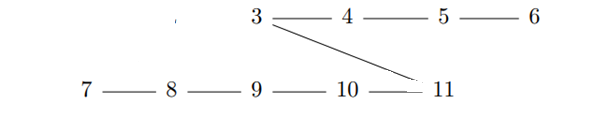
Остаточное дерево



Шаг 2.

Лист vmin=2, ребро (2,11), код =8,11

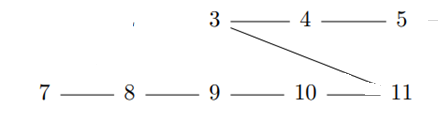
Остаточное дерево



Шаг 3.

Лист vmin=6, ребро (5,6), код =8,11,5

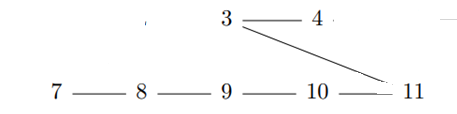
Остаточное дерево



Шаг 4.

Лист vmin=5, ребро (4,5), код =8,11,5,4

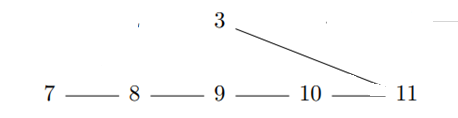
Остаточное дерево



Шаг 5.

Лист vmin=4, ребро (4,3), код =8,11,5,4,3

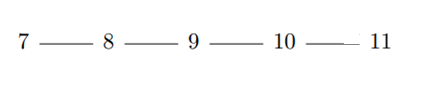
Остаточное дерево



Шаг 6.

Лист vmin=3, ребро (11,3), код =8,11,5,4,3,11

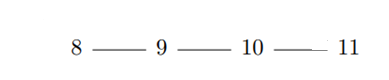
Остаточное дерево



Шаг 7.

Лист vmin=7, ребро (7,8), код =8,11,5,4,3,11,8

Остаточное дерево



Шаг 8.

Лист vmin=8, ребро (9,8), код =8,11,5,4,3,11,8,9

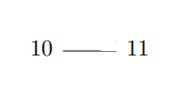
Остаточное дерево



Шаг 9.

Лист vmin=9, ребро (9,10), код =8,11,5,4,3,11,8,9,10

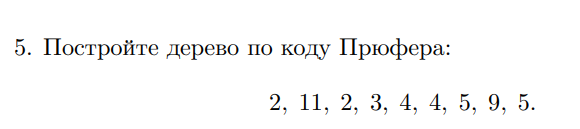
Остаточное дерево



Осталось 2 вершины, работа алгоритма закончена.

Код Прюфера: 8,11,5,4,3,11,8,9,10

# Задание 5



Решение.

s=(1,2,3,4,5,6,7,8,9,10,11)- список всех номеров вершин.

**Шаг 1.**

i1=1- первый номер в s, который отсутствует в коде p. Добавляем ребро (1,p1)=(1,2), номер 1 удаляем из s, а номер 2 из p.

**Шаг 2.**

s=(2,3,4,5,6,7,8,9,10,11);

p=(11,2,3,4,4,5,9,5);

текущее дерево



i2=6

p2=11

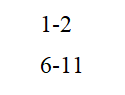
r2=(6,11)

**Шаг 3.**

s=(2,3,4,5,7,8,9,10,11);

p=(2,3,4,4,5,9,5);

текущее дерево



i3=7

p3=2

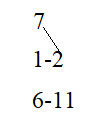
r3=(7,2)

**Шаг 4.**

s=(2,3,4,5,8,9,10,11);

p=(3,4,4,5,9,5);

текущее дерево



i4=2

p4=3

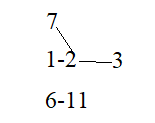
r4=(2,3)

**Шаг 5.**

s=(3,4,5,8,9,10,11);

p=(4,4,5,9,5);

текущее дерево



i5=3

p5=4

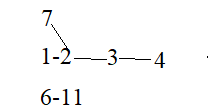
r5=(3,4)

**Шаг 6.**

s=(4,5,8,9,10,11);

p=(4,5,9,5);

текущее дерево



i6=8

p6=4

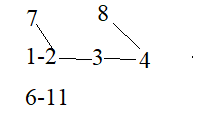
r6=(8,4)

**Шаг 7.**

s=(4,5,9,11);

p=(5,9,5);

текущее дерево



i7=4

p7=5

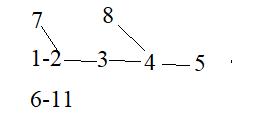
r7=(4,5)

**Шаг 7.**

s=(5,9,11);

p=(9,5);

текущее дерево



i7=11

p7=9

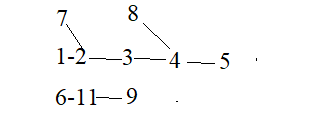
r7=(11,9)

**Шаг 8.**

s=(5,9);

p=(5);

текущее дерево



i8=9

p8=5

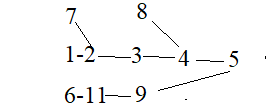
r8=(9,5)

**Шаг 8.**

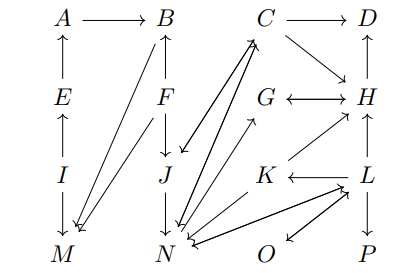
s=(5);

p=();

**Построили дерево по коду Прюфера:**



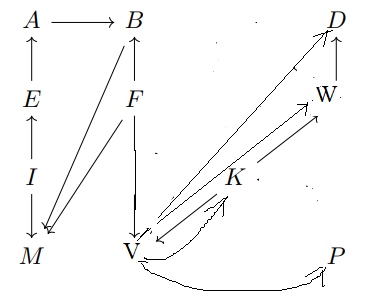
Задание 6. При помощи алгоритм Косарайю и Шарира найдите компоненты сильной связности данного графа



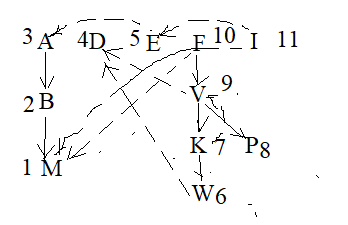
Из за наличия двойных стрелок и того что отношение сильной связности являются отношением эквивалентности имеем N,

G. Вводим три супер-вершины V={N,L,O,C,J}, W={G,H}

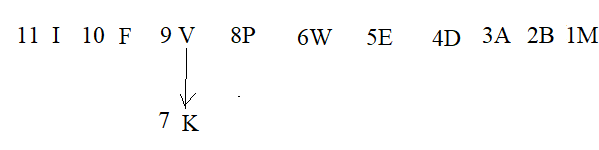
Граф приобретает вид



Строим лес DFS(Числа- DFS порядок завершения вершин)

**

Строим DFS лес для обратного направления ребер в порядке убывания DFS номеров вершин(поперечные ребра можно не наносить)



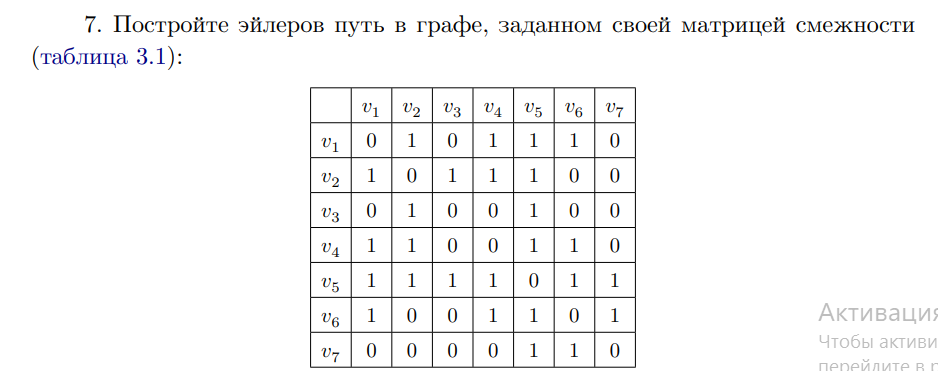
Получили одну сильную компоненту в сжатоv графе {V,K}

равную { N,L,O,C,J.K} в исходном графе:

Остальные сильно-связные компоненты:

{I},{F},{P},W={G,H},[E},{D},{A},{},{M}.

# Задание 7



Решение.

Строим данный граф.

1

2

3

4

5

6

7

1. Идем из первой вершины в соседнюю вершину с наименьшим номером по не пройденному ребру, удаляя при переходе ребро, пока это возможно. Процесс обязательно закончится в вершине 1

1

2

3

4

5

6

7

1

2

3

4

6

5

7

5

9

10

Имеем

частичный эйлеров цикл С1= 1- 2-3-5-1-4-2-5-4-6-1

и остаточный граф

1

2

3

4

5

6

7

1. В остаточном графе берем любую вершину, входящую в остаточный граф, например 5 и применяем тот же алгоритм в остаточном графе:

3

1

2

3

4

5

6

7

1

2

Получили второй эйлеров цикл c2=5-6-7-5, сцепленный с первым циклом по вершине 5, и остаточный граф, не имеющий ребер, все ребра пройдены, причеем только один раз.

1. Преобразуем первый цикл, делая вершину сцепления 5 начальной

С=5-1-4-2-5-4-6-1-2-3-5

1. Замещаем вершину 5 в первом цикле на второй цикл

C=C1+C2=5-6-7-5-1-4-2-5-4-6-1-2-3-5, получили замкнутый эйлеров путь.

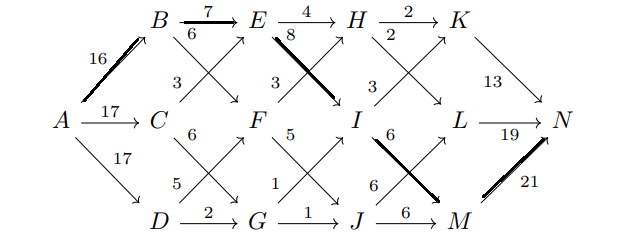
# Задание 8



Решение.

F0=0- начальный поток.

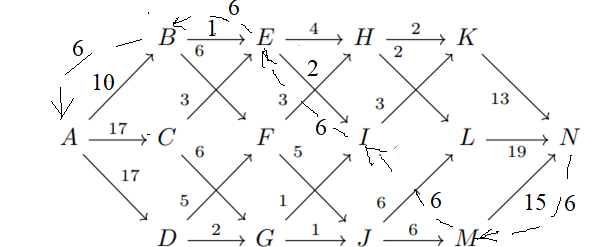
Первый аугментальнвй путь.



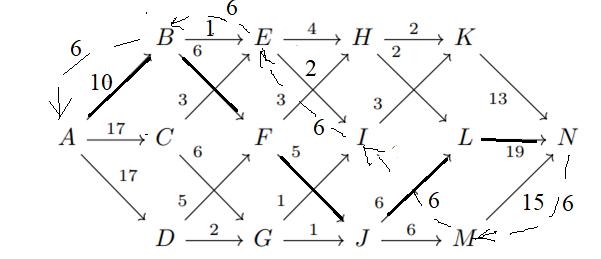
dF0=min(16,7,8,6,21)=6;

F1=F0+dF0=6

Остаточный граф



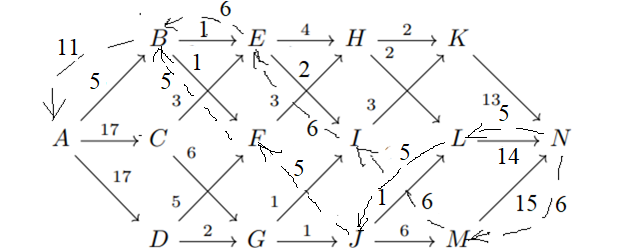
Второй аугментальный путь



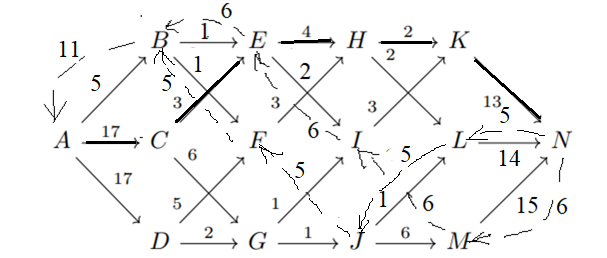
dF1=min(10,6,5,6,19)=5;

F2=F1+dF1=11

Остаточный граф



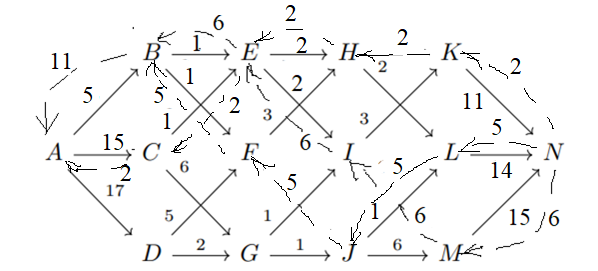
Третий аугментальный путь



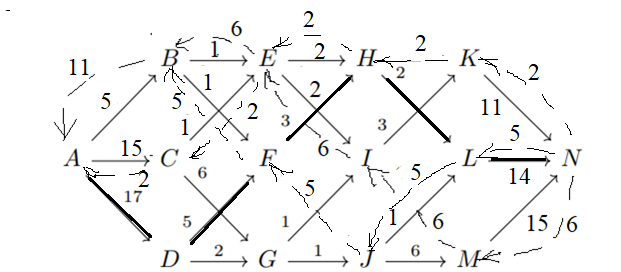
dF2=min(17,3,4,2,13)=2;

F3=F2+dF2=13

Остаточный граф

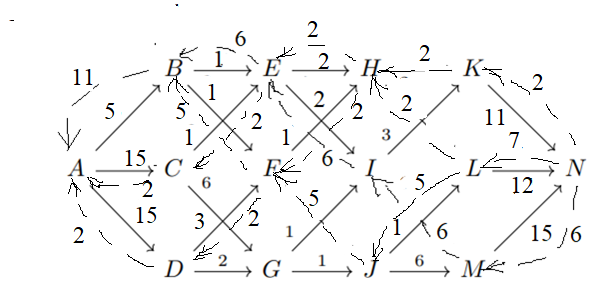


Четвертый аугментальный путь

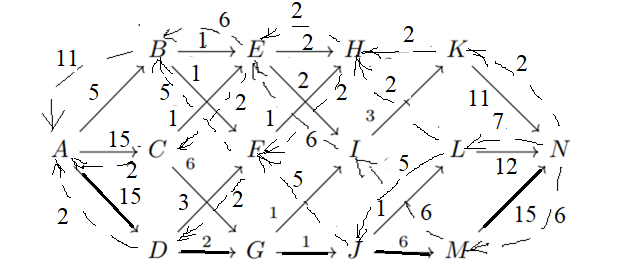


dF3=min(17,5,3,2,14)=2;

F4=F3+dF3=15

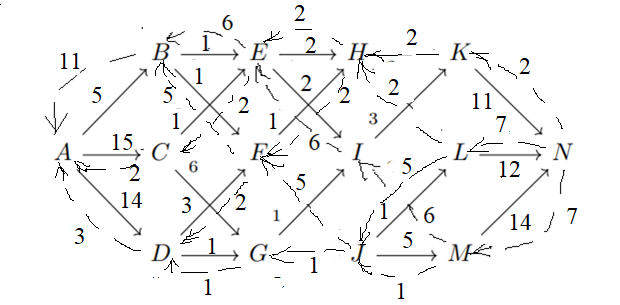


Пятый аугментальнвй путь

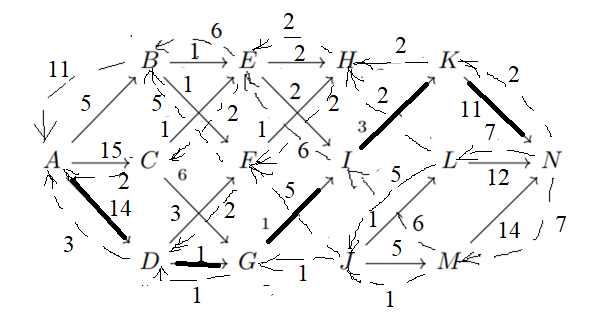


dF4=min(15,2,1,6,15)=1;

F5=F4+dF4=16

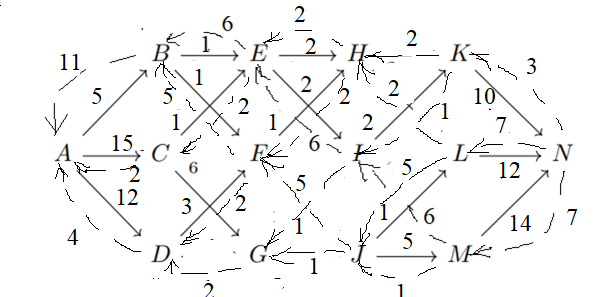


Шестой аугментальный путь:

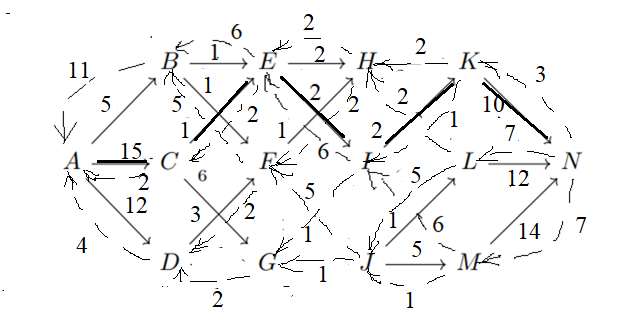


dF5=min(14,1,1,3,11)=1;

F6=F5+dF5=17

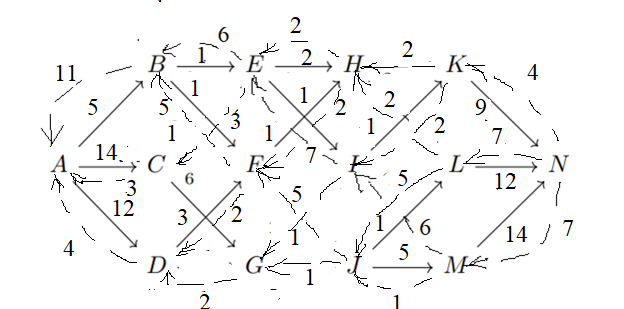


Седьмой аугментальный путь

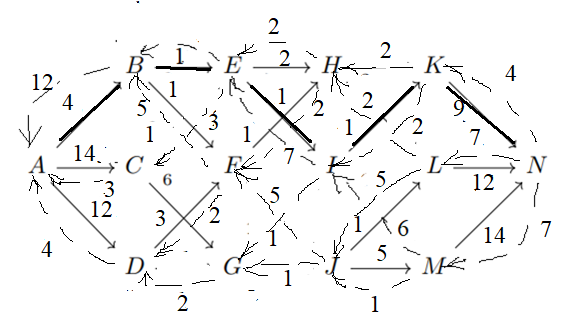


dF6=min(15,1,2,2,7)=1;

F7=F6+dF6=18;



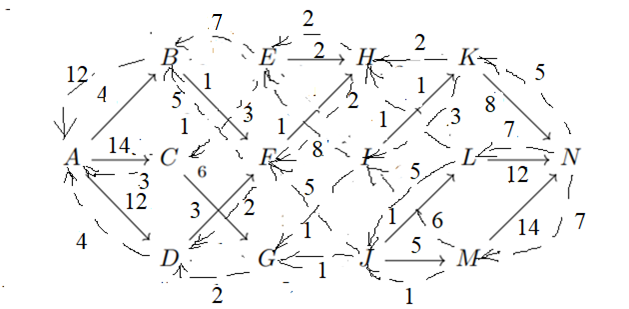
Восьмой аугментальный путь



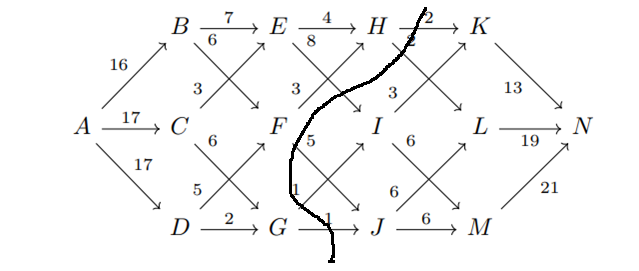
dF7=min(5,1,1,2,9)=1;

F8=F7+dF7=19;

Остаточный граф

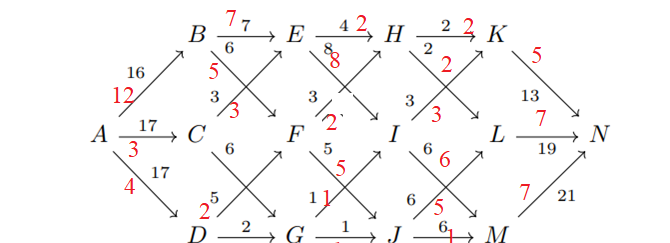


Новых аугментвльных путей нет, поток F8=19=Fmax является максимальным. В этом можно убедиться по тереме о минимальном разрезе.

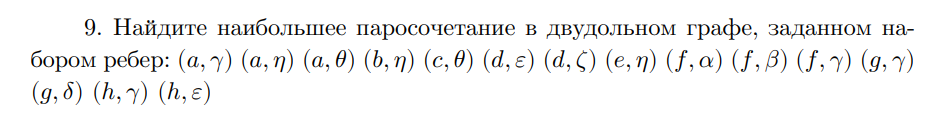


Имеется разрез с пропускной способностью 2+2+8+5+1+1=19=Fmax, следовательно поток Fmax=19 , следовательно поток является максимальным, а разрез минимальным.

Распределение потоков по лугам находим по весам обратных луг.



# Задание 9



Решение.

Граф:

a

b

c

e

f

g

h

ϒ

η

θ

ε

d

Ϛ

α

β

δ

Начальное паросочетание имеет мощность 7, максимально возможная мощность паросочетания =8.

a

b

c

e

f

g

h

ϒ

η

θ

ε

d

Ϛ

α

β

δ

Альтернирующее дерево из вершины e

e

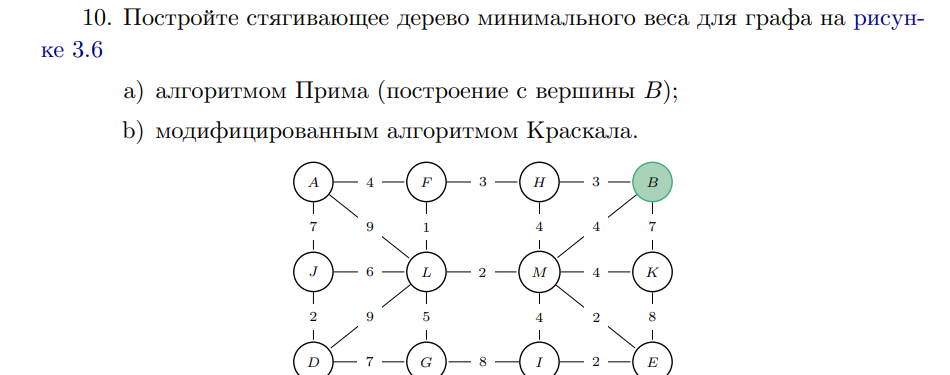
η

b

Альтернирующее дерево является венгерским, поэтому начальное паросочетание является максимальным.

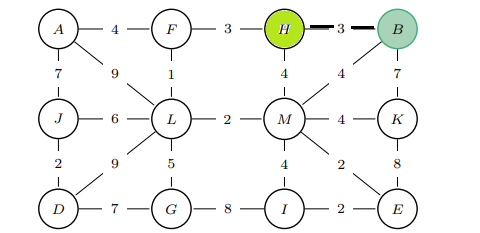
Mmax={(a,γ),(b,η),(c,θ),(f,α),(g,δ),(h,ε),(d,ζ)};

# Задание 10

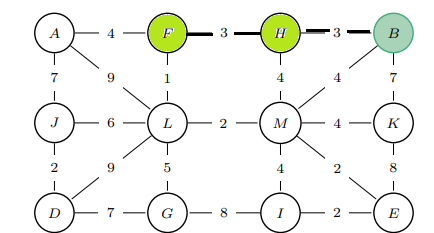


1. Алгоритм Прима

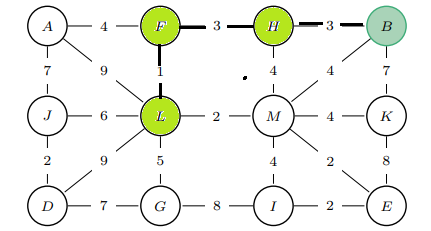
Шаг 1



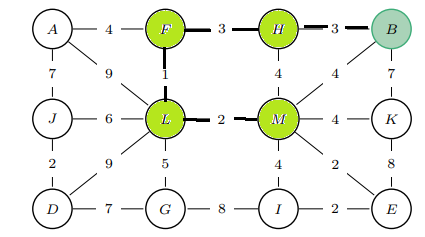
Шаг 2.



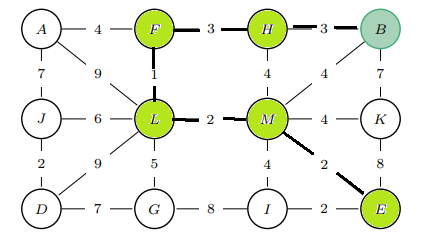
Шаг 3



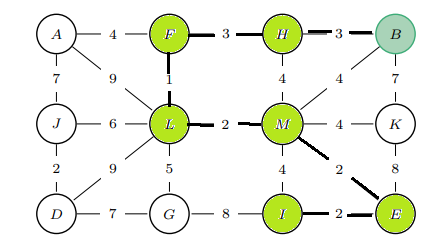
Шаг 4



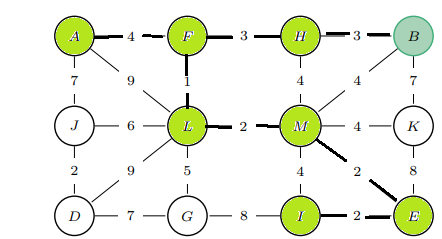
Шаг 5



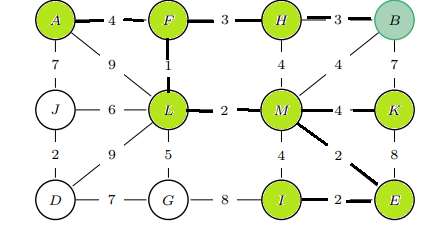
Шаг 6



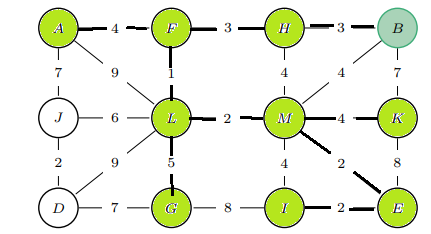
Шаг 7



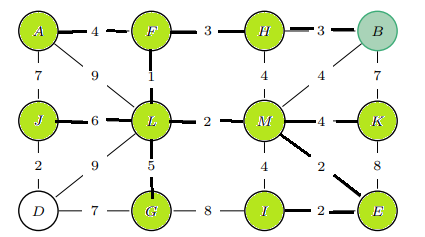
Шаг 8



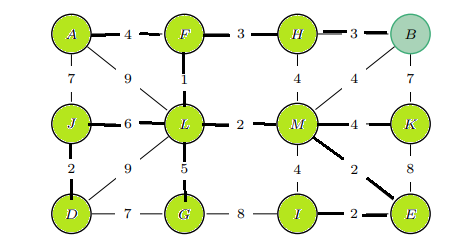
Шаг 9



Шаг 10



Шаг 11



Tmin={(B,H),(H,F),(F,L),(A,F),(L,M),(M,K),(M,E),(J,L),(J,D),(L,G)}

Wmin=3+3+4+1+6+2+4+2+5+2+2=34;

б)

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ КРАСКАЛА Упорядочим все ребра по возрастанию (неубыванию) весов. Присвоим вершинам графа различные номера (раскрасим вершины в разные цвета):

ЦИКЛ-ПОКА не кончились ребра

Рассмотреть очередное ребро

ЕСЛИ концы ребра окрашены в разные цвета

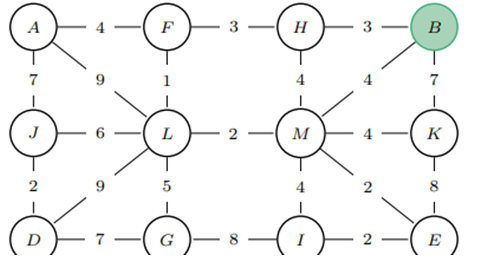
ТО добавить ребро в остовное дерево и перекрасить все вершины, номер которых совпадает с номером одного конца добавленного ребра в цвет второго конца этого ребра (для определенности можно перекрашивать в цвет вершины с большим номером)

КЦ

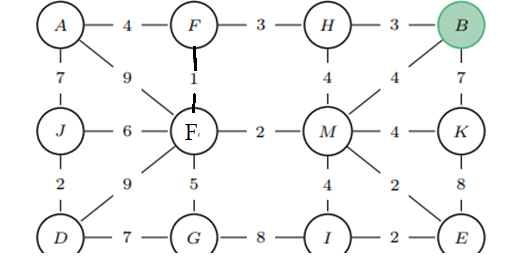
1. Сортировка ребер

(FL,1),(ME,2),(LM,2),(JD,2),(IE,2),(BH,3),(FH,3), (AF,4),(KM,4),(IM,4),(BM,4),(LG,5),(LJ,6),(DG,7),(BK,7),(AJ,7).(KE,8),(IG,8),(LD,9),(AL,9);

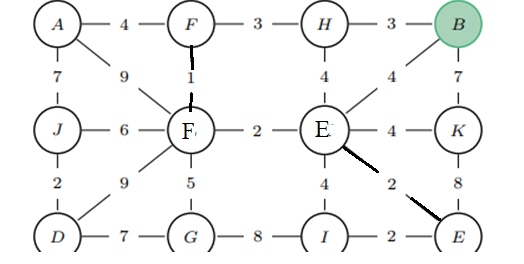
1. Исходная раскраска вершин



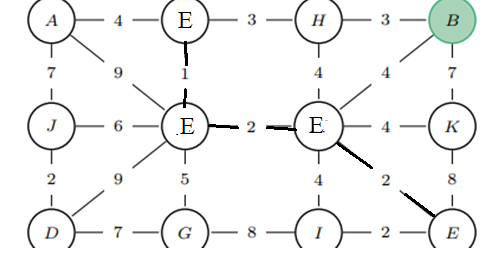
Шаг1 Ребро (FL,1):



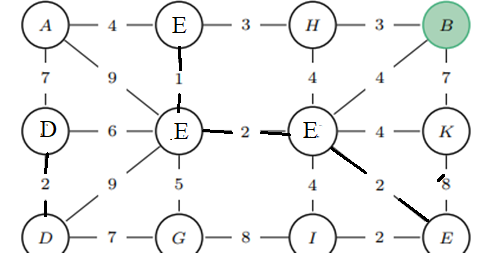
Шаг2 Ребро (ME,2):



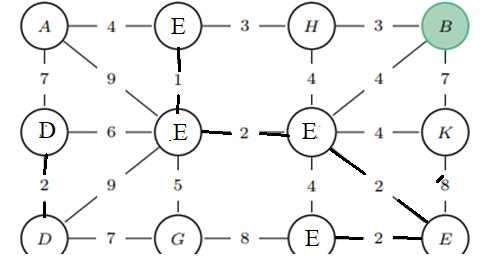
Шаг3 Ребро (ML,2):



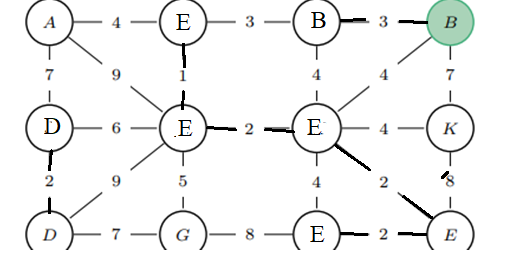
Шаг4 Ребро (JD,2):



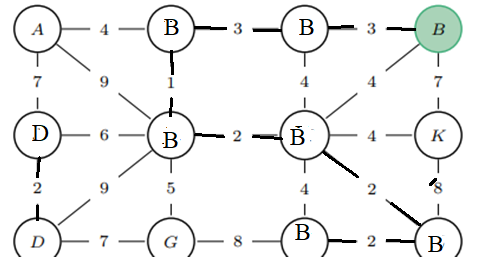
Шаг4 Ребро (AF,4):



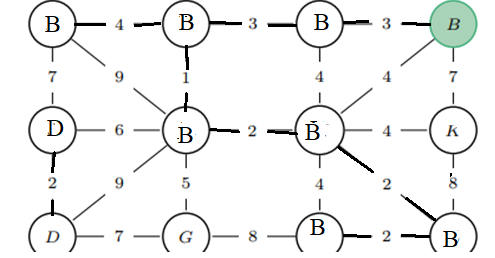
Шаг5 Ребро (BH,3):



Шаг6 Ребро (FH,3):



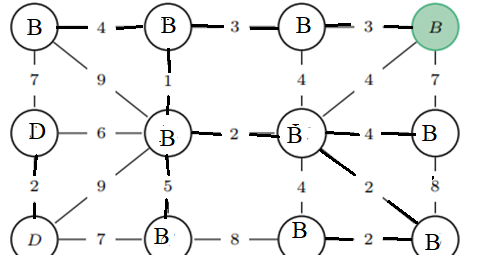
Шаг7 Ребро (FA,4):



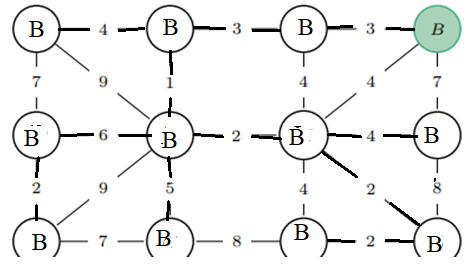
Шаг8 Ребро (IM,4):Пропускаем, соединяются вершины одного цвета:

Шаг9 Ребро (BM,4): Пропускаем, соединяются вершины одного цвета.

Шаг10 Ребро (LG,5):



Шаг11 Ребро (LJ,6):



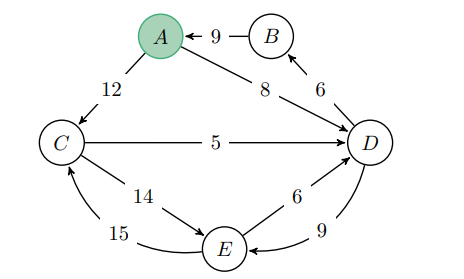
Получено то же самое остовное дерево веса 34, что и методом Прима.

Задание 11. Для графа на рисунке:

a) вычислите длины кратчайших путей от вершины 𝐴 до остальных вершин с помощью алгоритма Дейкстры;

b) c помощью алгоритма Флойда определите кратчайшие пути между всеми парами вершин. Выпишите кратчайший путь от вершины 𝐷 до вершины 𝐶 и его длину;

c) с помощью алгоритма Уоршелла постройте граф и матрицу достижимости.



а)

Алгоритм Дейкстры

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Шаг v | A | B | C | D | E |
| 0 | 0,0 | ∞,0 | ∞,0 | ∞,0 | ∞,0 |
| 1 | 0,0 | ∞,0 | 12,A | 8,A | ∞,0 |
| 2 | 0,0 | ∞,0 | 12,A | 8,A | 17,D |
| 3 | 0,0 | 14,D | 12,A | 8,A | 17,D |
| 4 | 0,0 | 14,D | 12,A | 8,A | 17,D |
| 5 | 0,0 | 14,D | 12,A | 8,A | 17,D |

Длины кратчайших путей от вершины А до остальных вершин

|  |  |
| --- | --- |
| v | D(A,v) |
| A | 0 |
| B | 14 |
| C | 12 |
| D | 8 |
| E | 17 |

б) Алгоритм Флойда

Весовая, маршрутная матрицы

W,M=

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | ∞,0 | ∞,B | 12,C | 8,D | ∞,0 |
| B | 9,A | ∞,0 | ∞,0 | ∞,0 | ∞,0 |
| C | ∞,0 | ∞,0 | ∞,0 | 5,D | 14,E |
| D | ∞,0 | 6,B | ∞,0 | ∞,0 | 9,E |
| E | ∞,0 | ∞,B | 15,C | 6,D | ∞,0 |

Шаг 1 v=A- промежуточная вершина путей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | ∞,0 | ∞,B | 12,C | 8,D | ∞,0 |
| B | 9,A | ∞,0 | 21,A | 17,A | ∞,0 |
| C | ∞,0 | ∞,0 | ∞,0 | 5,D | 14,E |
| D | ∞,0 | 6,B | ∞,0 | ∞,D | 9,E |
| E | ∞,0 | ∞,B | 15,C | 6,D | ∞,0 |

Шаг 2 v=B- промежуточная вершина путей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | ∞,0 | ∞,B | 12,C | 8,D | ∞,0 |
| B | 9,A | ∞,0 | 21,A | 17,A | ∞,0 |
| C | ∞,0 | ∞,0 | ∞,0 | 5,D | 14,E |
| D | 15,B | 6,B | 27,B | 23,D | 9,E |
| E | ∞,0 | ∞,B | 15,C | 6,D | ∞,0 |

Шаг 3 v=C- промежуточная вершина путей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | ∞,0 | ∞,B | 12,C | 8,D | 26,C |
| B | 9,A | ∞,0 | 21,A | 17,A | ∞,0 |
| C | ∞,0 | ∞,0 | ∞,0 | 5,D | 14,E |
| D | 15,B | 6,B | 27,B | 23,D | 9,E |
| E | ∞,0 | ∞,B | 15,C | 6,D | 29,C |

Шаг 4 v=D- промежуточная вершина путей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 23,D | 14,D | 12,C | 8,D | 17,D |
| B | 9,A | 23,A | 21,A | 17,A | 26,A |
| C | 32,D | 11,D | 32,D | 5,D | 14,E |
| D | 15,B | 6,B | 27,B | 23,D | 9,E |
| E | 21,D | 12,D | 15,C | 6,D | 15,D |

Шаг 5 v=E- промежуточная вершина путей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 23,D | 14,D | 12,C | 8,D | 17,D |
| B | 9,A | 23,A | 21,A | 17,A | 26,A |
| C | 32,D | 11,D | 32,D | 5,D | 14,E |
| D | 15,B | 6,B | 24,E | 15,E | 9,E |
| E | 21,D | 12,D | 15,C | 6,D | 15,D |

Окончательный результат

Dmin,Mmin=

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | 23,D | 14,D | 12,C | 8,D | 17,D |
| B | 9,A | 23,A | 21,A | 17,A | 26,A |
| C | 32,D | 11,D | 32,D | 5,D | 14,E |
| D | 15,B | 6,B | 24,E | 15,E | 9,E |
| E | 21,D | 12,D | 15,C | 6,D | 15,D |

Кратчайшие пути между всеми парами вершин определяются рассчитанной маршрутной матрицей

Mmin=

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E |
| A | D | D | C | D | D |
| B | A | A | A | A | A |
| C | D | D | D | D | E |
| D | B | B | E | E | E |
| E | D | D | C | D | D |

Длина кратчайшего пути от вершины D до вершины С равна

Dmin(D,C)=24;

Соответствующий кратчайший путь из D в С;

Pmin(D,C)=D-M(D,C)=E-M(E,C)= C=(D,E,C);

1. Алгоритм Уоршела;

Матрица смежности или достижимости за один переход

A=

Матрица достижимости за 2 перехода

A2=sign(A2)=sign(∙

Матрица достижимости за 3 перехода

A3=sign(A2∙A)=sign(∙

Матрица достижимости за 4 перехода

A4=sign(A2∙A2)=sign(∙

Матрица достижимости:

T=sign(E+A+A2+A3+A4)=

граф достижимости

A

B

C

D

E

Задание 12. Отношение 𝑇 задано на множестве целых чисел {60, 25, 01, 71, 68, 50, 20, 24} следующим условием: 𝑥𝑇 𝑦 ⇔ в наборе имеется элемент 𝑧, такой что (𝑥 − 𝑧)(𝑦 − 𝑧) < 0

) проверить, является ли отношение рефлексивным, антирефлексивным, симметричным, асимметричным, антисимметричным, транзитивным;

b) построить матрицу и граф этого отношения;

c) определить, является ли отношение отношением эквивалентности, частичного порядка, линейного порядка;

d) для отношения эквивалентности построить классы эквивалентности;

e) для отношения частичного порядка применить алгоритм топологической сортировки и получить отношение линейного порядка

Решение

(𝑥 − 𝑧)(𝑦 − 𝑧) < 0 означает, что z лежит между x,y, т.е. T- это отношение не быть соседними b и не совпадающими при линейном упорядочении чисел.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| xi | X3 | X7 | X8 | X2 | X6 | X1 | X5 | X4 |
| V(xi) | 1 | 20 | 24 | 25 | 50 | 60 | 68 | 71 |

b)

Матрица отношения

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 |
| X1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| X2 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| X3 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| X4 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| X5 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| X6 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| X7 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| X8 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

Граф отношения

X1

X2

X3

X4

X5

X6

X7

X8

1. Проверка свойств

рефлексивность- **нет**, на диагонали матрицы отношения есть нуди;

антиреылексивность- **да**, на диагонали матрицы все нули:

симметричность- **да**, матрица А симметрична, также по смыслу не быть соседними симметричное отношение;

транзитивность – **нет**, например потому, что соседние числа могут иметь(и имеют в данном случае) общий не соседний элемент.

с) Принадлежность к основным классам

эквивалентность- **нет**, так как отношение не транзитивно;

частичный порядок- **нет**, т.к. не транзитивно.

линейный порядок- **нет**, т.к. не транзитивно

Пункты d,e не применимы.