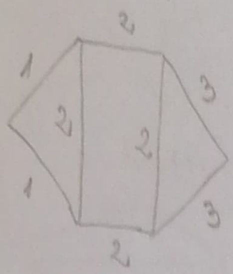
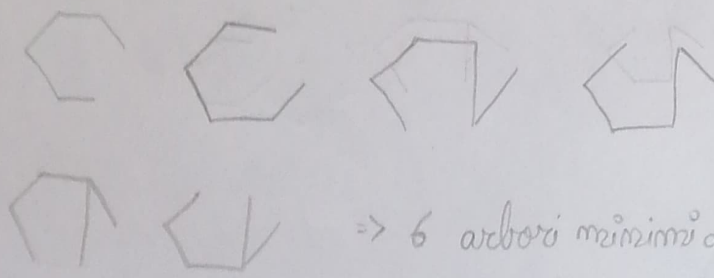
MĂRIRE

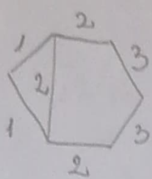
ARBORE MINIM ACOPERIRE

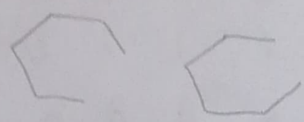
ARBORE MINIM ACOPERIRE

1. 

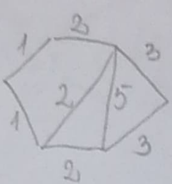


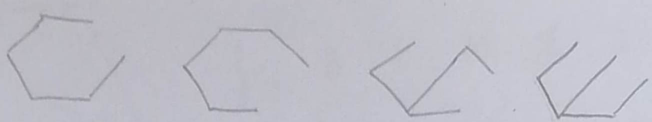
=> 6 arbori minimi de acoperire

2. 

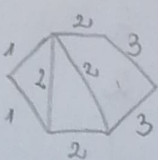


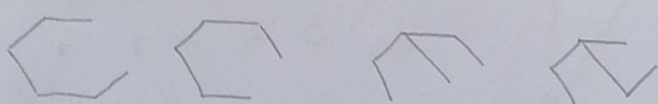
=> 2 ama

3. 

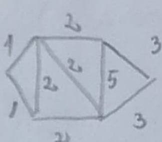


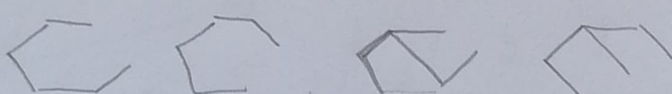
=> 4 ama

4. 



=> 4 ama

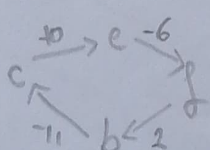
5. 



=> 4 ama

(iei unul dintre minimele J-te și nu duci muchii spre ce ai deja conex)

BELLMAN-FORD

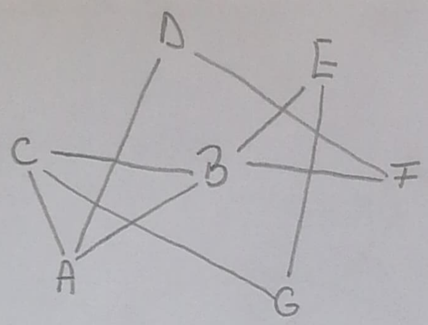
1. false, avem ciclu negativ (dacă mergem pe margini)
2. false, avem 

3. Modificarea algoritmului constă în înlocuirea primului for cu un while, care se repetă până nu mai putem îmbunătăți (cu o variabilă booleană).

4.

BFS

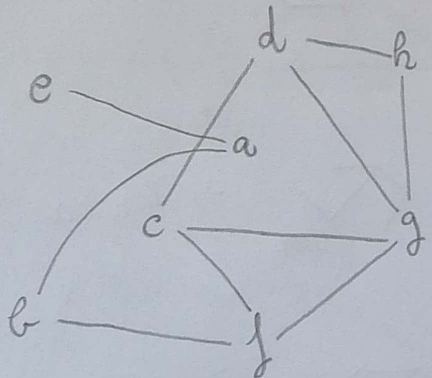
1.



nod:	A	B	C	D	E	F	G
d:	0	1	1	1	2	2	2
π:	nil	A	A	A	B	B	C

(începând cu A)

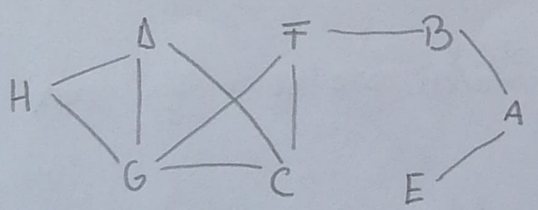
2.



nod:	a	b	c	d	e	f	g	h
d:	3	2	0	1	4	1	1	2
π:	b	f	nil	c	a	c	c	d

(începând cu C)

3.

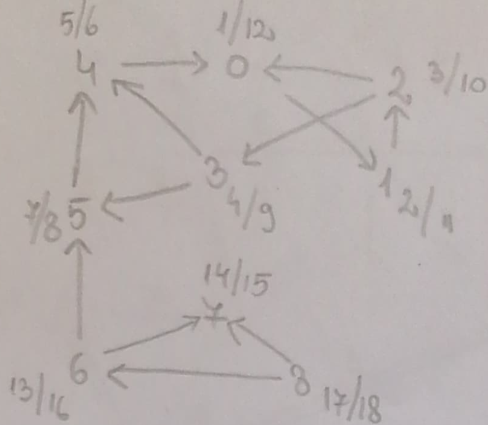


nod:	A	B	C	D	E	F	G	H
d:	0	1	3	4	1	2	3	4
π:	nil	A	F	C	A	B	F	G

(începând cu A)

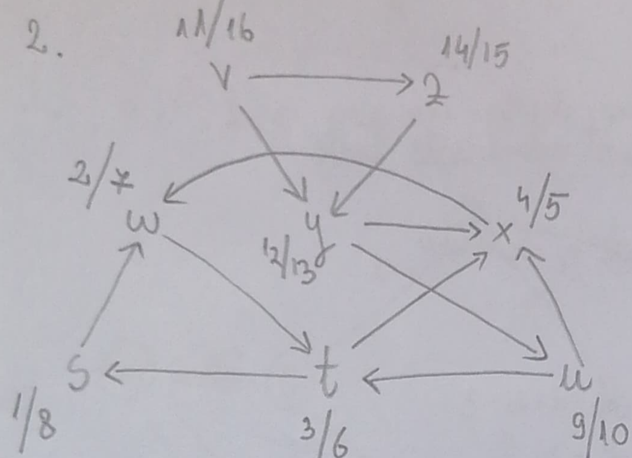
DFS

1.



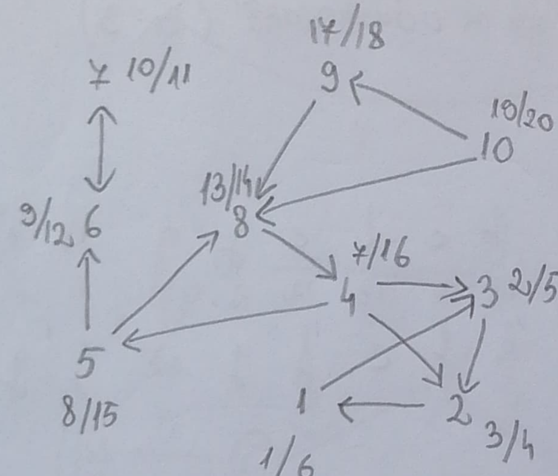
nod:	1	2	3	4	5	6	7	8
d:	1	3	4	5	7	13	14	17
f:	12	10	9	6	8	16	15	18

2.



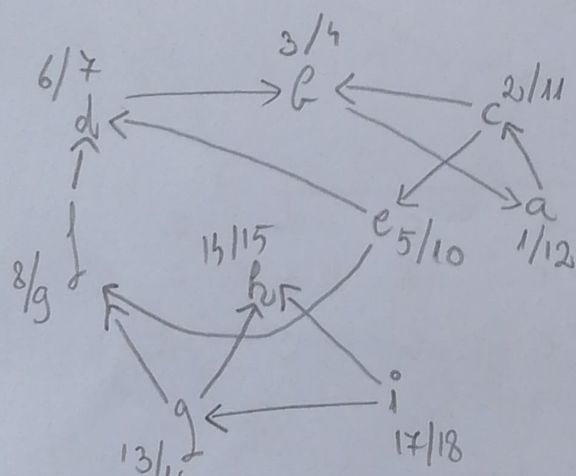
nod:	s	t	u	v	w	x	y	z
d:	1	3	9	11	2	4	12	14
f:	8	6	10	16	7	5	13	15

3.



nod:	1	2	3	4	5	6	7	8	9	10
d:	1	3	2	7	8	9	10	13	17	19
f:	6	4	5	16	15	12	11	14	18	20

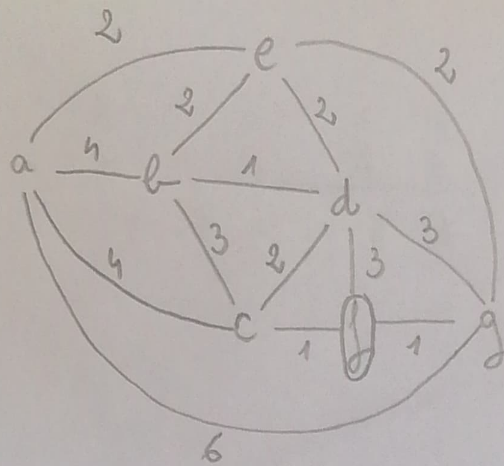
4.



nod:	a	b	c	d	e	f	g	h	i
d:	1	3	2	6	5	8	13	14	17
f:	12	4	11	7	10	9	16	15	18

DIJKSTRA

1.



nod:	a	b	c	d	e	f	g
d:	5	4	1	3	3	0	1
π:	c	c	f	f	g	nil	f

pas 1: $S = \{f\}$ și c, d, g - adiacente

c.d = 1   d.d = 3   g.d = 1   sunt actualizate toate

c.π = f   d.π = f   g.π = f

pas 2: $S = \{f, c\}$ - care are cea mai mică val. a distanței

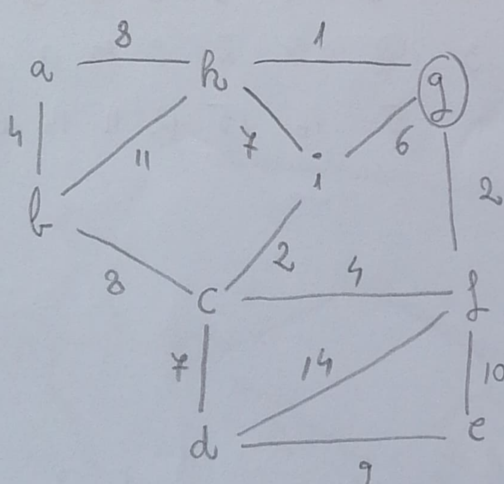
a, b, d - adiacente

a.d = 1+4 = 5   b.d = 1+3 = 4   d.d = 1+2 = 3

a.π = c   b.π = c   d.π = c

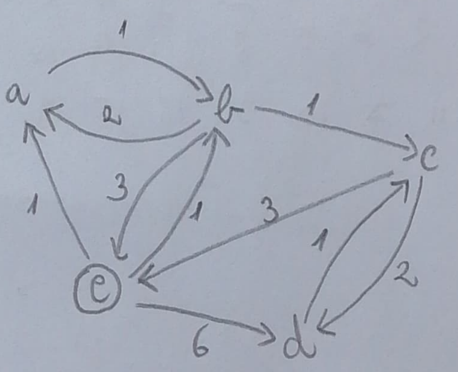
se actualizează   nu se actualizează (3 = 3)

2.



nod:	a	b	c	d	e	f	g	h	i
d:	9	12	6	13	12	2	0	1	6
π:	h	h	f	c	f	g	nil	g	g

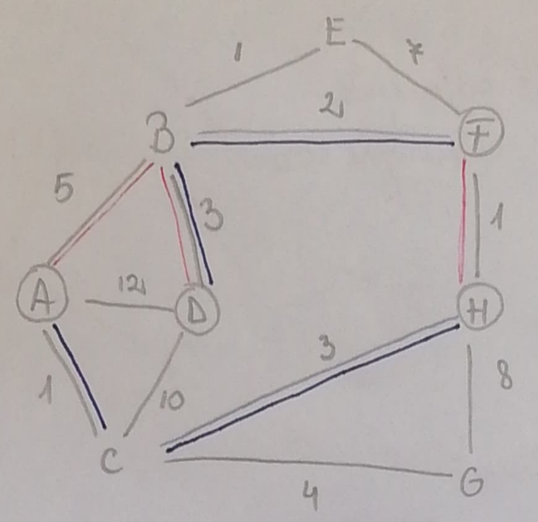
3.



nod:	a	b	c	d	e
d:	1	1	2	4	0
π:	e	e	b	c	nil

EULERIAN

1.



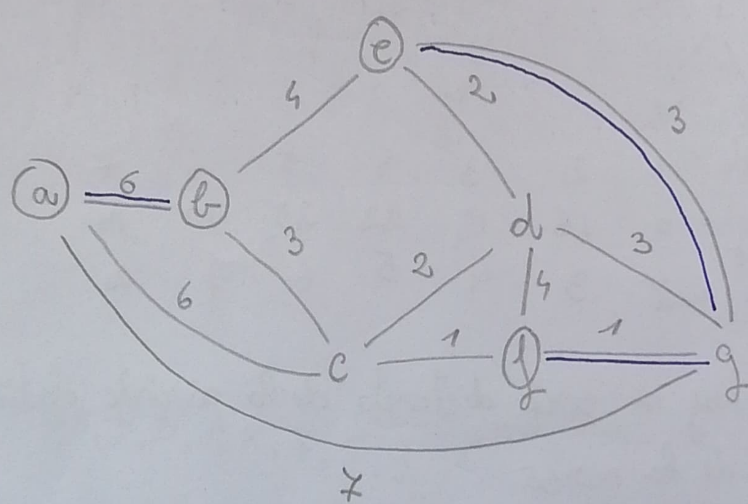
trebuie să facem grad par la toate nodurile

-> sunt 2 variante de transformare

roșu: adaugă 9

albastru: adaugă 9

2. a, b, f, e nu au grad par



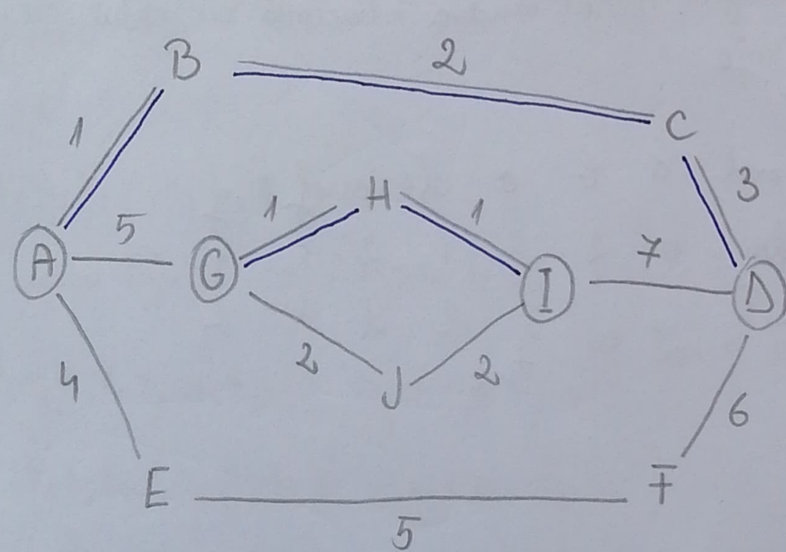
se adaugă 10

-> așa e și ex 5, dar avem

AC, CB, EG, GF

5<6

3.

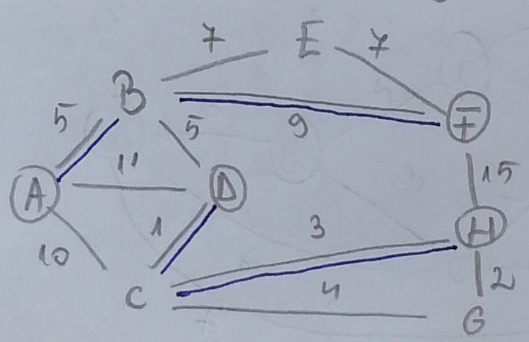


dacă aș face AG, ID => 12

dar AB, BC, CD, GH, HI => 8

-> așa e și ex 6

4.



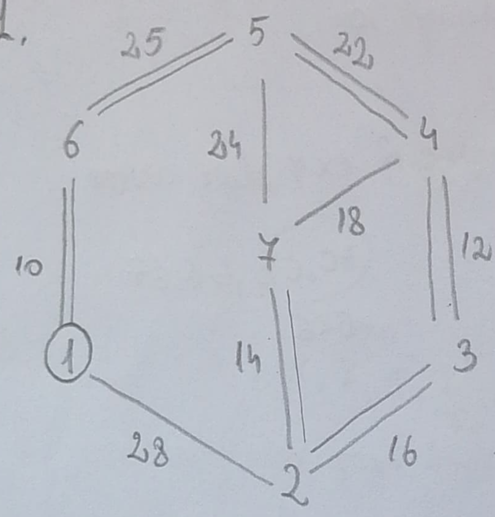
AB, BF, DC, CH => 18

IZOMORFE

1. nu: nodurile cu grad 2 ar trebui să fie adiacente
2. da: verific pe rând gradurile nodurilor și adiacentele lor
3. da: destul de evident
4. da: iar e evident
5. da: am luat iar pe grade și adiacente
6. nu: nodurile cu grad 3 nu sunt legate direct
7. nu: nodul cu grad 4 are adiacente 2 cu grad 1 și 2 cu grad 2 (nu 3)

PRIM

1.

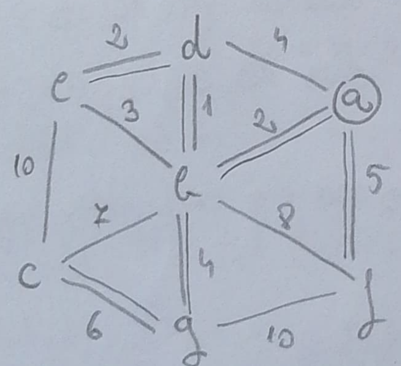


nod =	1	2	3	4	5	6	7
key =	0	16	12	22	25	10	14
π =	nil	3	4	5	6	1	2

→ la key se scrie distanța de la părinte efectiv, nu de la sursă

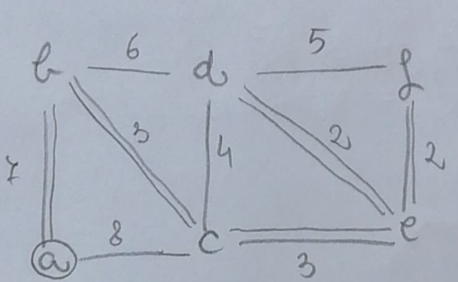
→ alegem mereu muchia adiacentă cu costul cel mai mic

2.



nod =	a	b	c	d	e	f	g
key =	0	2	6	1	2	5	4
π =	nil	a	g	b	d	a	b

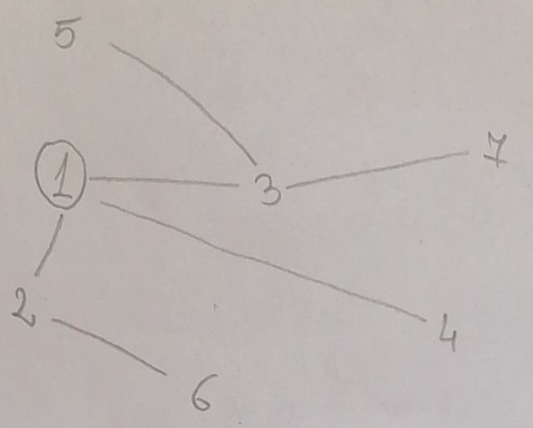
3.

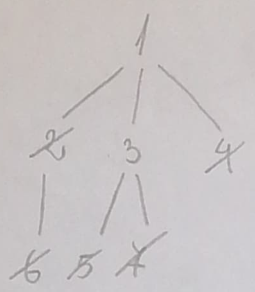


nod =	a	b	c	d	e	f
key =	0	7	3	2	3	2
π =	nil	a	b	e	c	e

PRUFER

1.

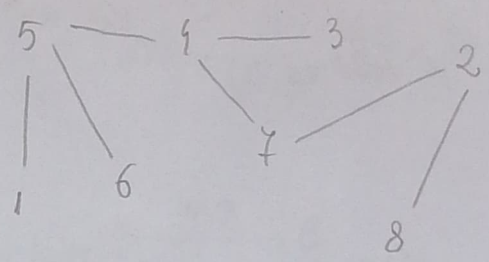


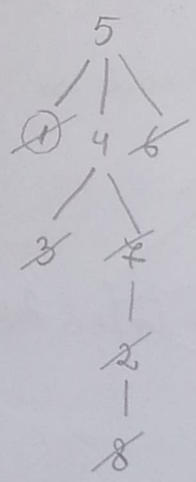


1, 3, 2, 1, 3, 1

→ căutăm frunza minimă, punem părintele ei în secvență și o tăiem

2.





=> 5, 4, 5, 2, 7, 4, 5

3. 1, 5, 3, 3, 1, x, 7, 5, 7 rădăcina

→ fiecare vârf are grad impar => rădăcina apare de nr. impar de ori și restul de nr. par

=> x = 7

4. 3, 4, 3, x, 4, 2, 1 => x = 2

1 apare deja de nr. impar de ori

5. 1, 1, 7, x, 3, 3, 7, 6, 7 => x = 6

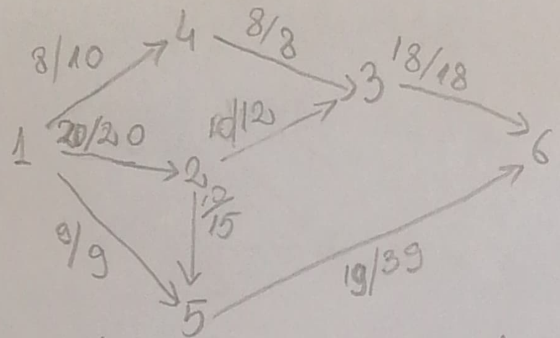
6. 1, 5, 3, 3, 1, x, 7, 5, 7 => x = 7

7. 10, x, 3, 3, 1, 7, 7, 1, 1 => x = 10

8. 2, 1, 2, x, 1, 7, 5 => x = 7

9. 1, 2, x, 3, 2, 1, 4, 4, 1 => x = 3

FORD FULKERSON

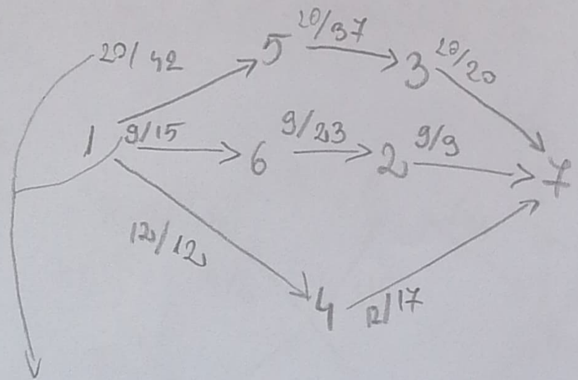
1. 

$8+10+10+9=37$

basically avem de găsit fluxul maxim

2. $20+20=40$

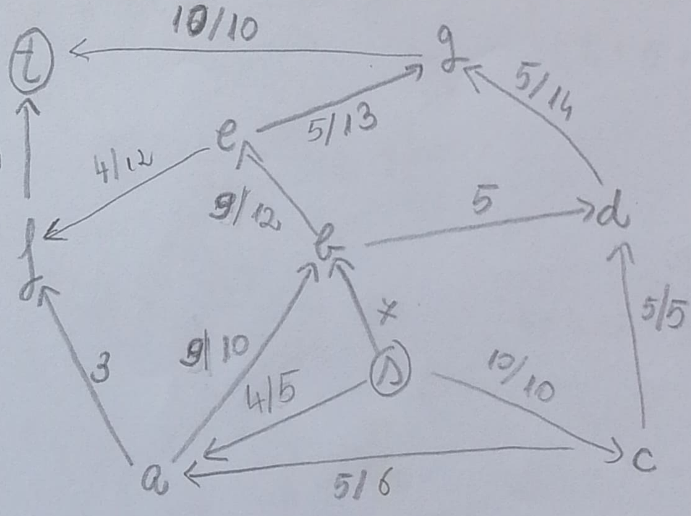
3. $10+10=20$ (pe jos n-am pe unde să-i trimit pe ultimele 5)

4. 

$20+9+12=41$

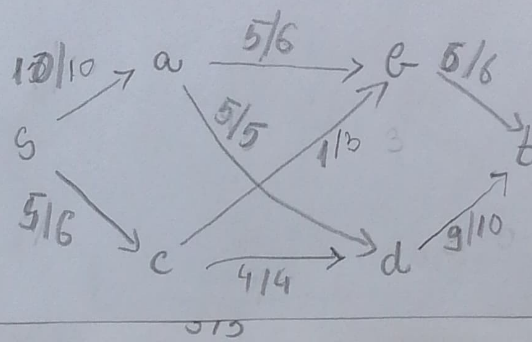
nu am pe unde să trimit ultimele unități

FLUX

1. 

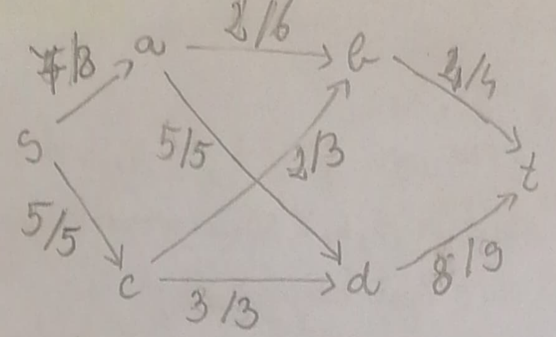
$10+4=14$

luăm pe rând și vedem cât putem duce

2. 

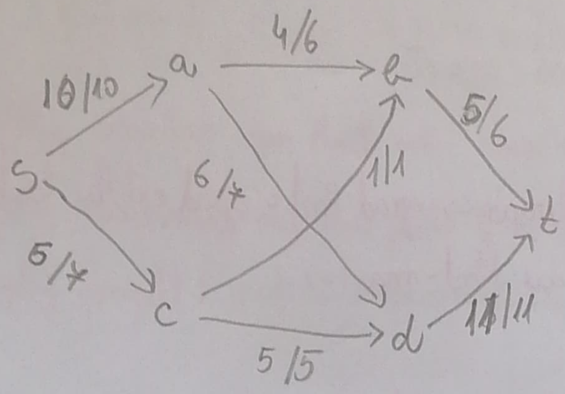
$4+5+5+1=15$

3.



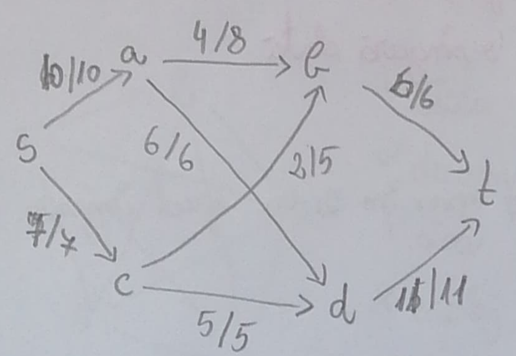
$5+5+2=12$

4.



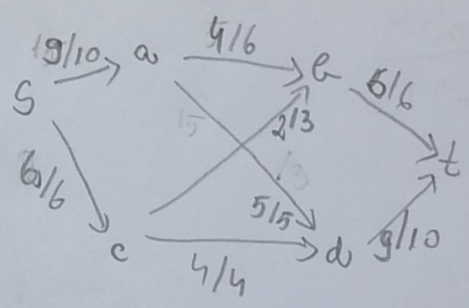
$5+6+1+5=17$

5.



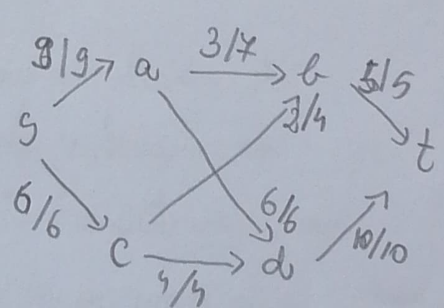
$5+2+6+4=17$

6.



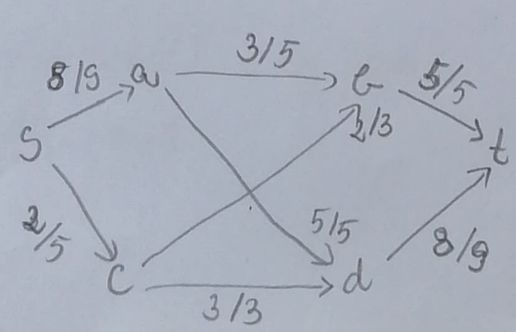
$4+2+5+4=15$

7.



$4+2+6+3=15$

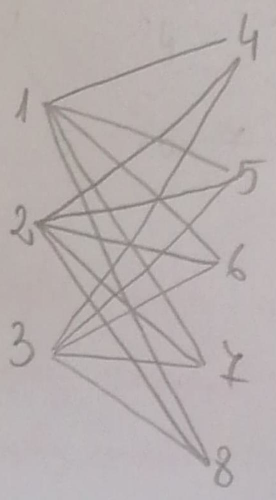
8.



$3+2+5+3=13$

BIPARTITE

1. $K_{3,5}$ conține:



× cuplaj perfect : nu, ar fi 1-4, 2-5, 3-6 și 7,8 fără
× lanț eulerian: nu e, 4, 5, 6, 7, 8 au grad impar
× ciclu hamiltonian: nu e (poți încerca idk)
× ciclu eulerian: dacă nu are lanț, nu are nici ciclu

=> niciun răspuns corect

=> graf orientat eulerian <=> grad int = grad ext la toate sau maxim 2 au |int - max| = 1

=> lanț eulerian: trece prin fiecare o singură dată
=> lanț hamiltonian: trece prin fiecare o singură dată
=> cuplaj perfect: conține toate nodurile
pt. lanț eulerian: toate au grad par sau ∃ maxim 2 cu grad impar

2. drumuri de lungime 2 în $K_{11,12}$ =>
formulă: $C_m^2 n + C_n^2 m$ $\qquad C_n^k = \frac{n!}{k!(n-k)!}$

3. drumuri de lungime 2 în $K_{11,18}$ =>
4. drumuri de lungime 2 în $K_{15,7}$ =>

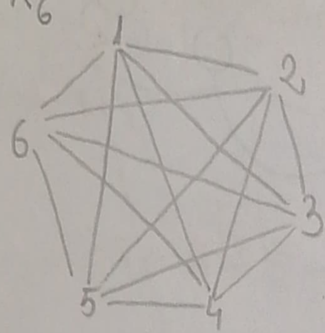
5. lungime 2 în $K_{15,18}$ =>

6. lungime 2 în $K_{21,10}$ =>

7. lungime 2 în $K_{21,12}$ =>

COMPLETE

1. K_6



✓ ciclu hamiltonian: are 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 1

✓ cuplaj maxim

× drum eulerian: nu are, toate nodurile au grad impar

✓ cuplaj complet 1-2, 3-4, 5-6

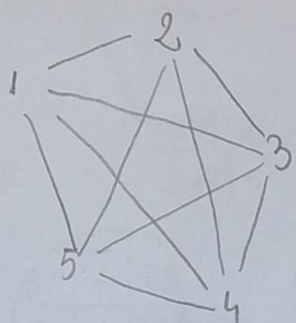
× ciclu eulerian: nu are, nu are nici drum

cuplaj maxim: am luat nr. maxim de perechi efectiv posibile (am ales bine)

cuplaj maximal: nu mai poate fi lărgit (posibilitate de ales prost)

cuplaj complet: cred că înseamnă cuplaj perfect, adică intră toate nodurile

2. K_5



✓ ciclu eulerian: da, au toate grad par

✓ drum eulerian: da

✓ ciclu hamiltonian: da 1 -> 2 -> 3 -> 4 -> 5 -> 1

× cuplaj complet: 1-2, 3-4, 5 fără

✓ cuplaj maxim: orice graf are

4. K_2

1 — 2

✓ drum eulerian: da

× ciclu eulerian: nu

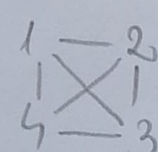
× ciclu hamiltonian: nu

nu are pe unde să se întoarcă

✓ cuplaj complet: o sg. pereche

✓ cuplaj maxim

3. K_4



✓ cuplaj maxim

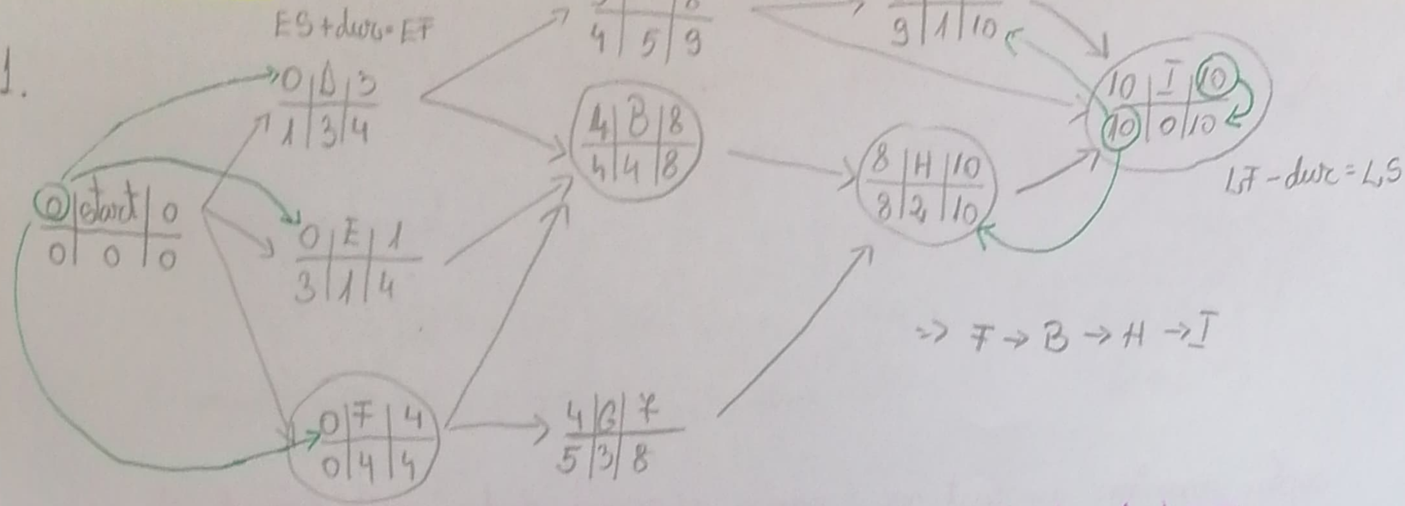
✓ cuplaj perfect: 1-2, 3-4

× ciclu eulerian
× drum eulerian
} avem grad impar la 4 noduri

✓ ciclu hamiltonian: 1 -> 2 -> 3 -> 4 -> 1

DRUM CRITIC

1.



=> F → B → H → I

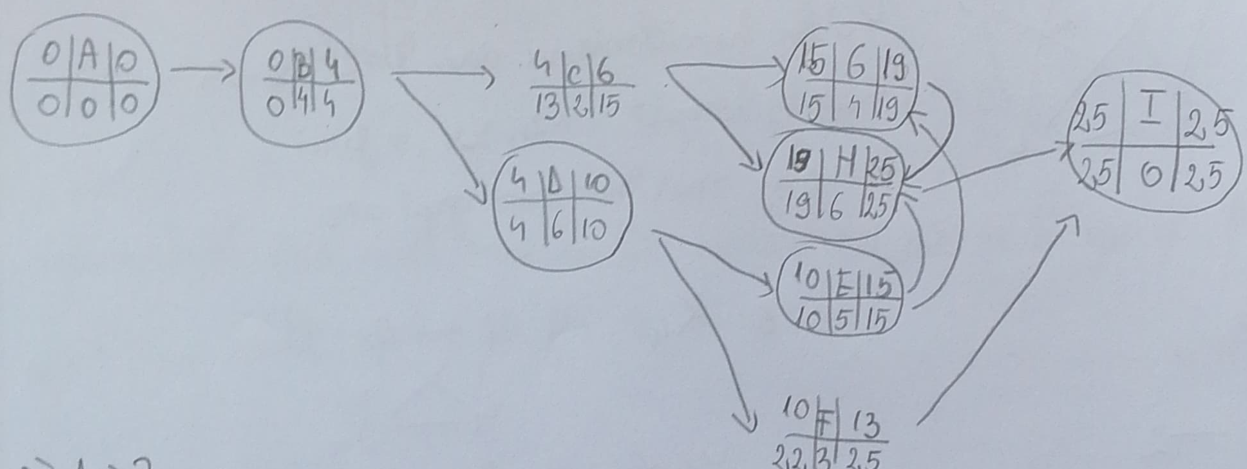
ES	act.	EF
LS	dur.	LF

-> dacă depinde de mai multe (ca și B) iau EF cel mai mare

-> $LF_u = \min(LS_v)$ unde $v \in adj[u]$ (nodurile de după →)

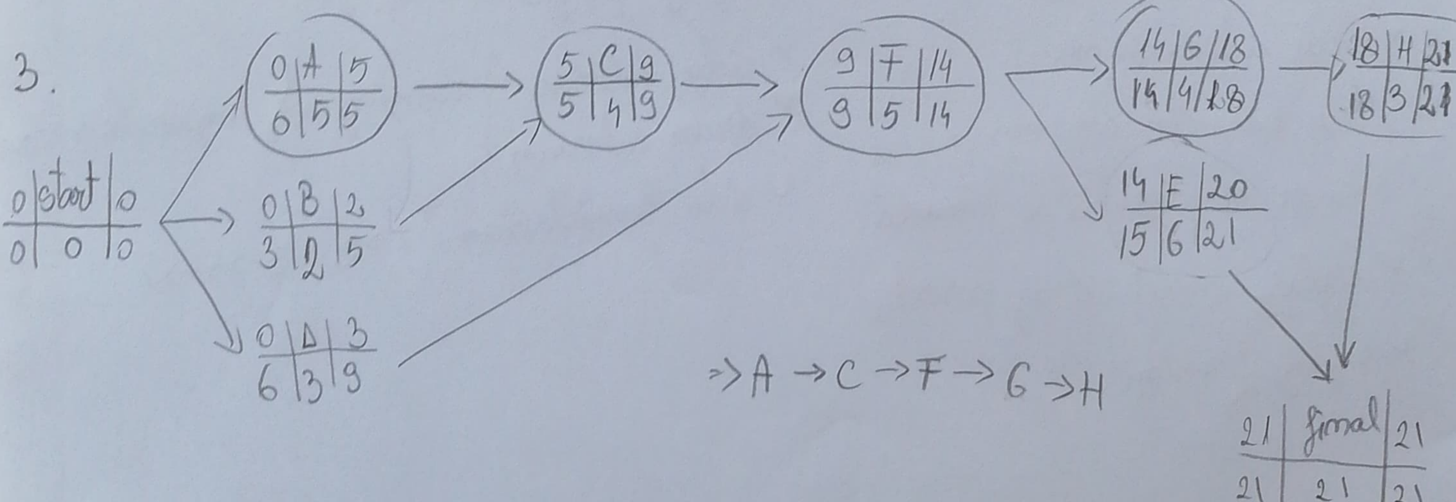
-> drumul critic e cel care are ES=LS și EF=LF

2.



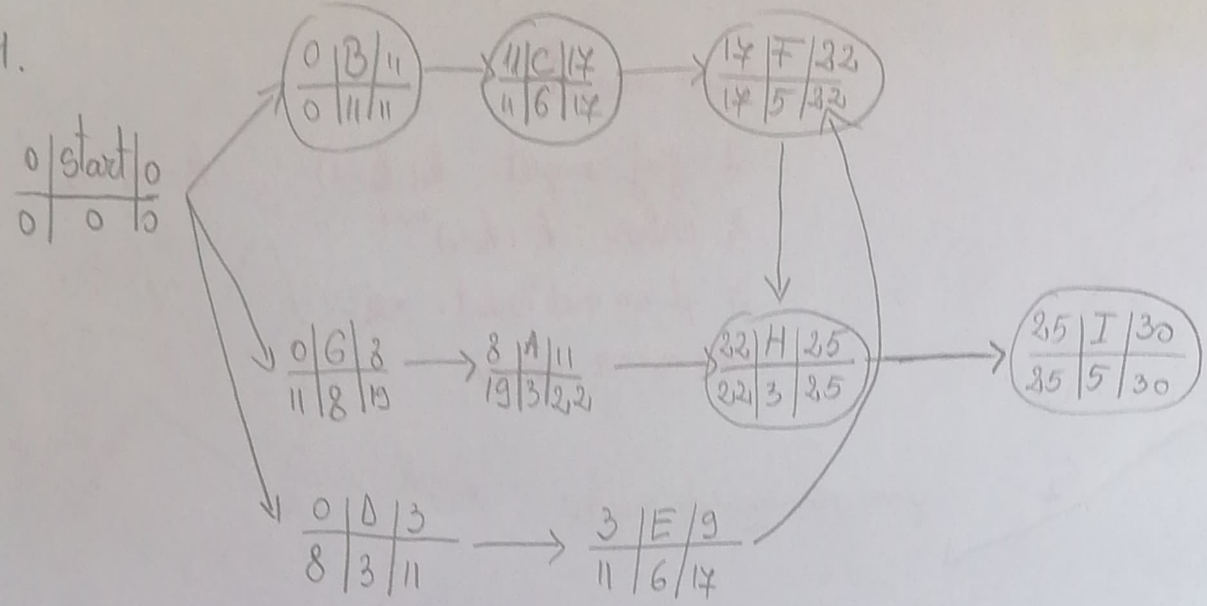
=> A → B → D → E → G → H → I

3.



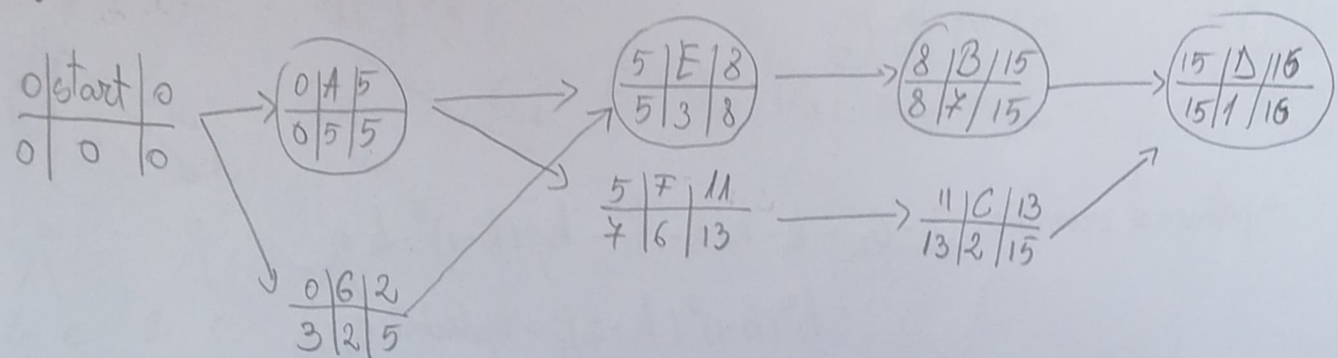
=> A → C → F → G → H

4.



=> B -> C -> F -> H -> I

5.



=> A -> E -> B -> D

-> pun start dacă sunt mai multe noduri cu care se începe și final dacă sunt mai multe noduri de finish

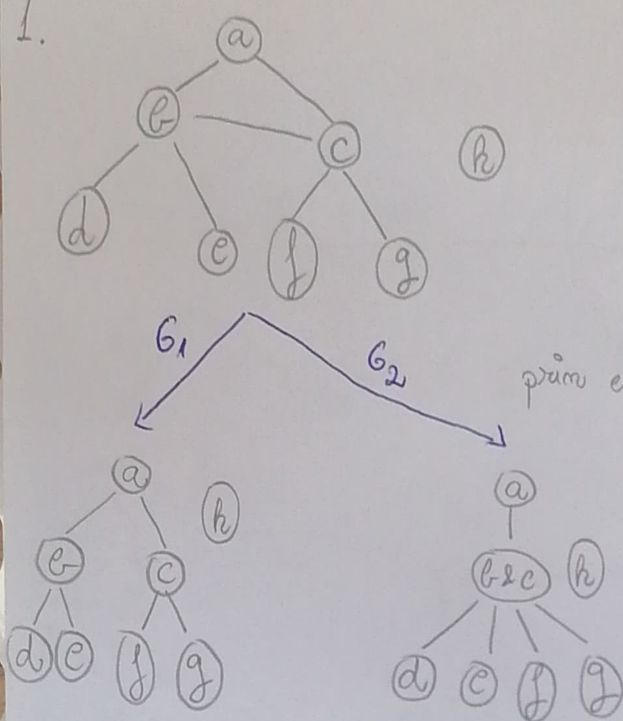
POLINOM CROMATIC

cazuri cunoscute:

1. graf complet: $k(k-1)\cdot\ldots\cdot(k-n+1)$
2. arbori: $k(k-1)^{n-1}$
3. pt. un nod izolat: $\cdot k$

unde n = nr. noduri

1.



prin eliminare de muchii

$G_1 - G_2$

CUNOSCUT CUNOSCUT

$\Rightarrow$ polinomul cromatic: $G_1 - G_2 = k\cdot k(k-1)^6 - k\cdot(k-1)^5\cdot k =$

$= k^2(k-1)^5(k-2)$ $\Rightarrow$ rădăcini 0, 1, 2

$\Rightarrow$ nr. cromatic este 3

graful poate fi colorat în 3 culori

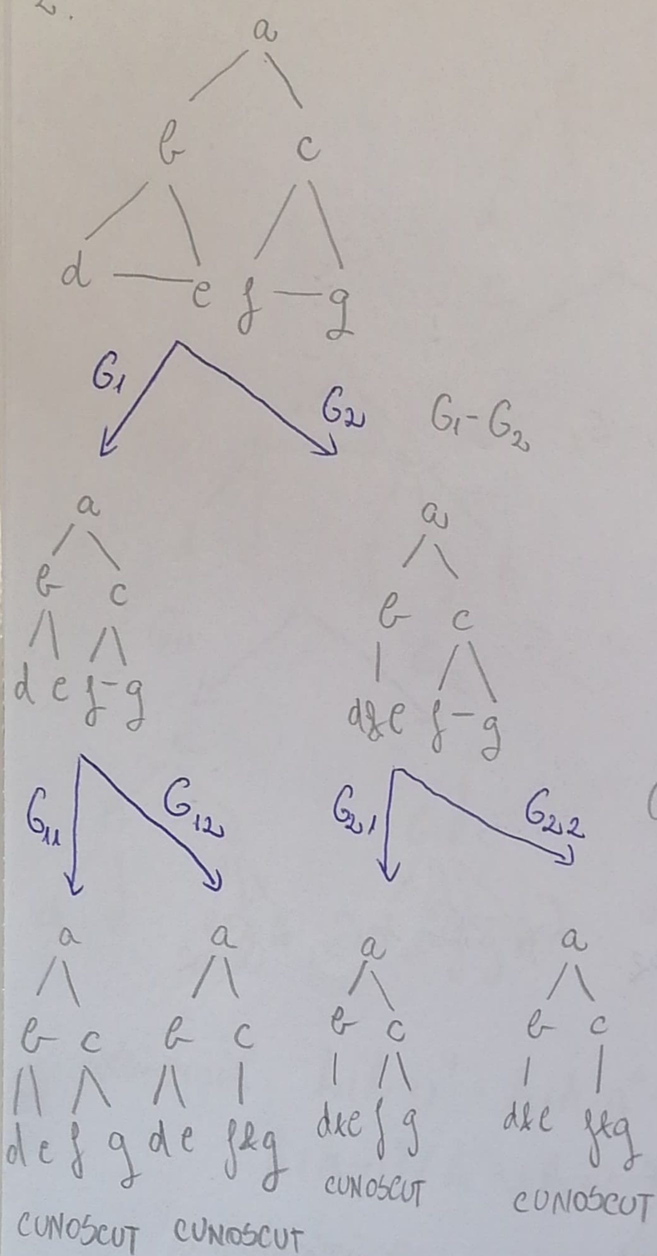
$\Rightarrow$ nr. de colorări posibile se obține înlocuind nr. cromatic în polinom

$\Rightarrow$ prin eliminare de muchii $\Rightarrow G_1 - G_2$

prin adăugare de muchii $\Rightarrow G_1 + G_2$

$\Rightarrow$ nr. colorări posibile: $3^2\cdot 2^5\cdot 1 = 9\cdot 32 = 270 + 18 = 288$

2.



$G_1 - G_2$

$G_{11} - G_{12} - (G_{21} - G_{22}) = G_{11} - G_{12} - G_{21} + G_{22}$

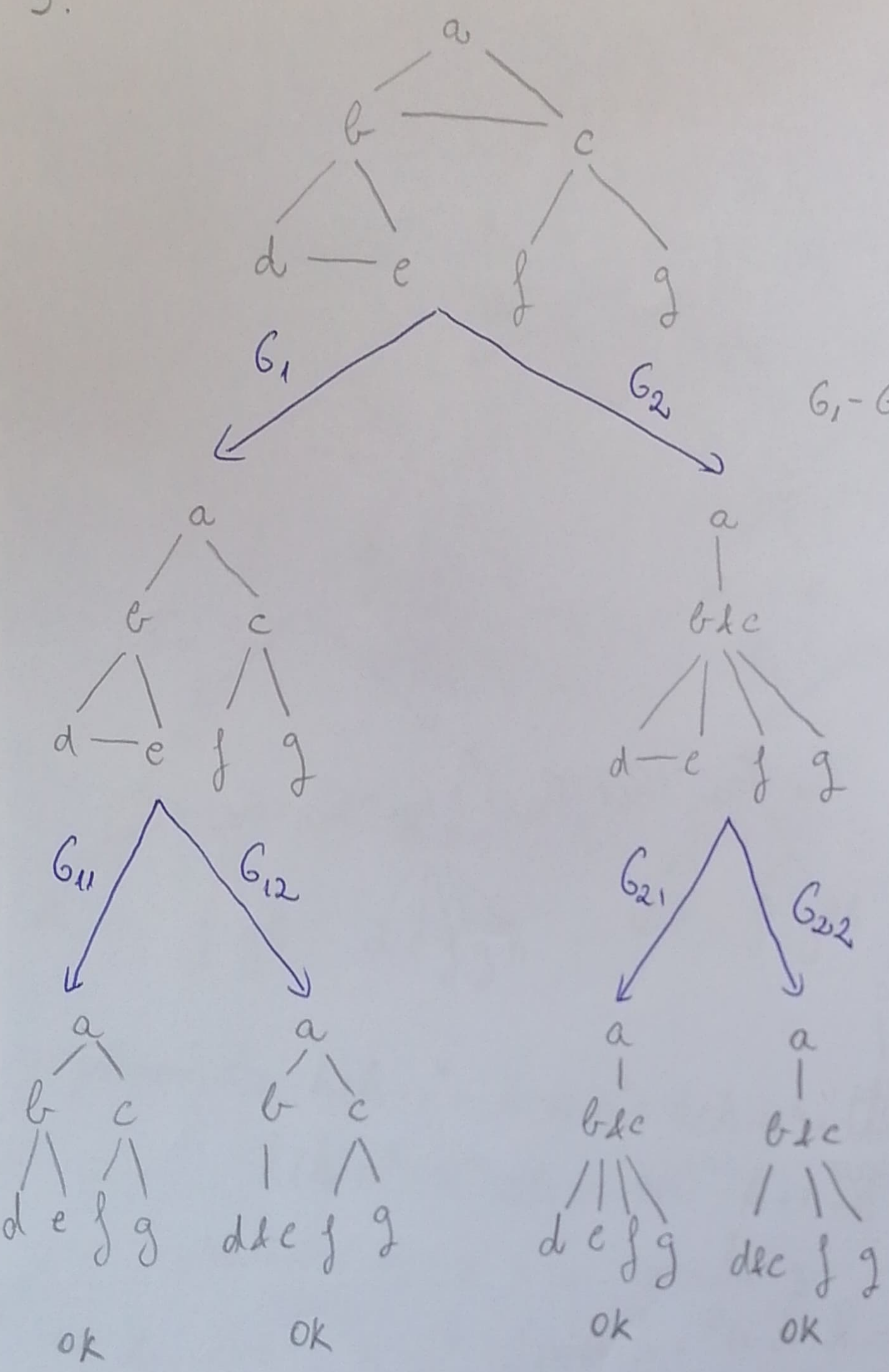
$\Rightarrow$ polinomul $k(k-1)^6 - k(k-1)^5 - k(k-1)^5 + k(k-1)^4 =$

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

$= k(k-1)^4(k^2 - 4k + 4) \Rightarrow$ rădăcini $0, 1, 2 \Rightarrow$ nr. cromatic 3

nr. de colorări posibile: $3 \cdot 2^4 \cdot (9 - 12 + 4) = 48 \cdot 1 = 48$

3.



$G_1 - G_2$

$G_{11} - G_{12} - G_{21} + G_{22}$

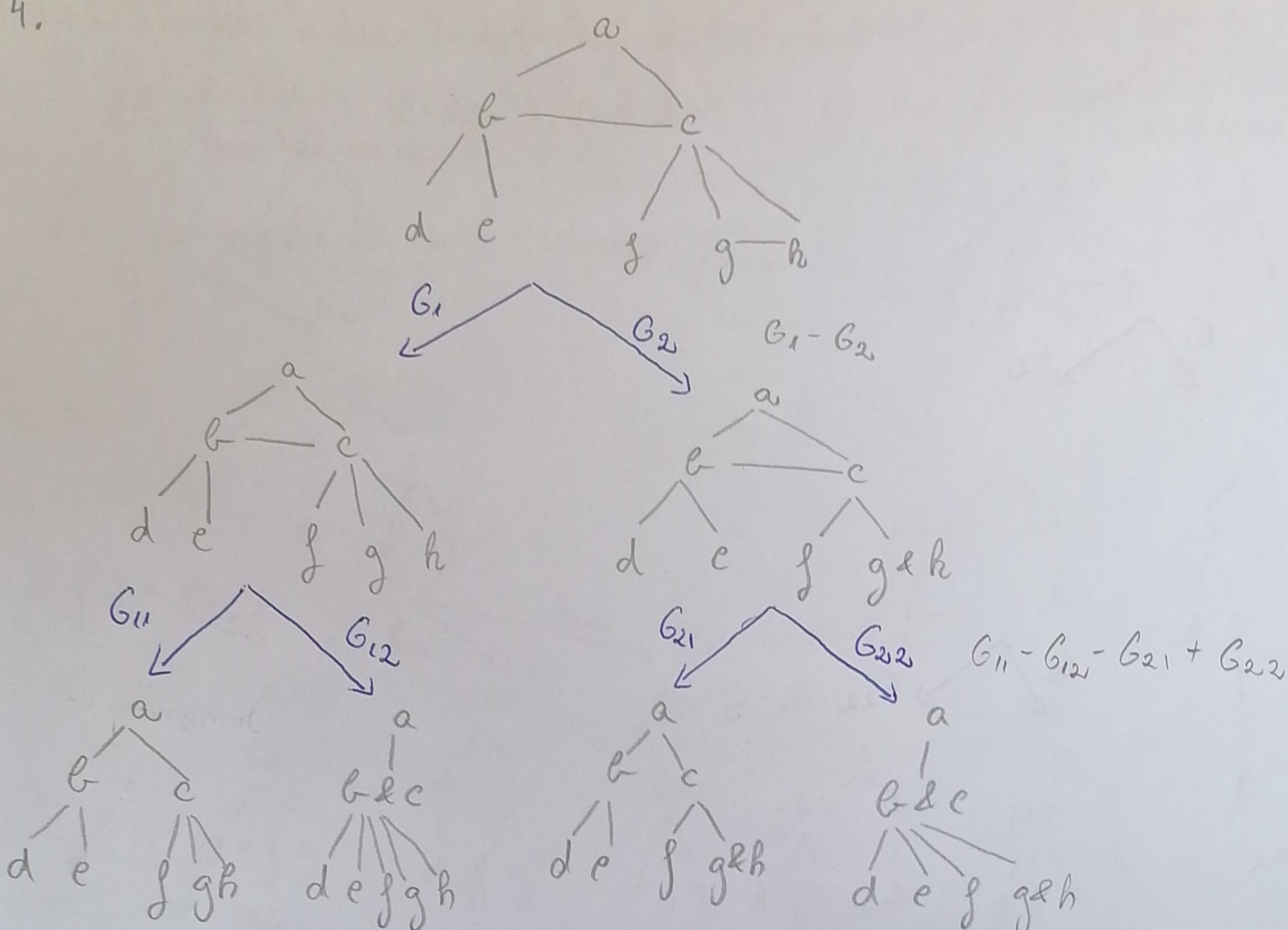
$$k(k-1)^6 - k(k-1)^5 - k(k-1)^5 + k(k-1)^4 =$$

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

$$= k(k-1)^4 (k^2 - 4k + 4)$$ => rădăcini 0, 1, 2 => nr. cromatic 3

nr. de colorări 48

4.



$G_1 - G_2$

$G_{11} - G_{12} - G_{21} + G_{22}$

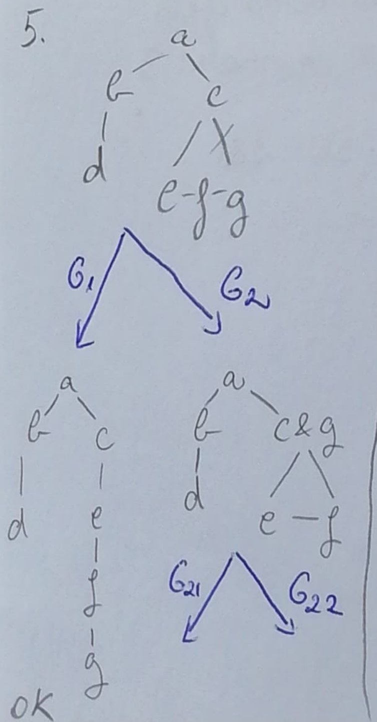
$\Rightarrow$ polinomul: $k(k-1)^7 - k(k-1)^6 - k(k-1)^6 + k(k-1)^5 =$

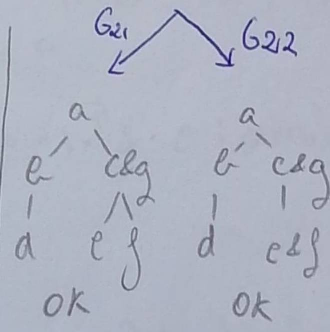
$= k(k-1)^5(k^2-2k+1-k+1-k+1+1) = k(k-1)^5(k^2-4k+4)$

$\Rightarrow$ rădăcini $0, 1, 2 \Rightarrow$ nr. cromatic 3

nr. de colorări: $3 \cdot 2^5 \cdot (9-12+4) = 3 \cdot 32 \cdot 1 = 96$ colorări

5.





$G_1 - G_2 = G_1 - (G_{21} - G_{22}) =$

$= G_1 - G_{21} + G_{22}$

$\Rightarrow$ polinomul $k(k-1)^6 - k(k-1)^5 + k(k-1)^4 =$

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

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

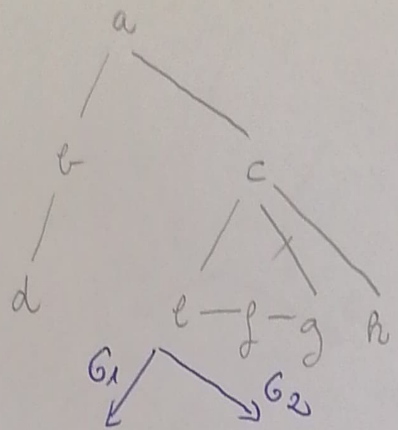
↳ nu are răd reale

$\Rightarrow$ rădăcini $0, 1 \Rightarrow 2$ nr. cromatic

$\Rightarrow$ nr. colorări $= 2 \cdot 1^4 \cdot (4-6+3) =$

$= 2$

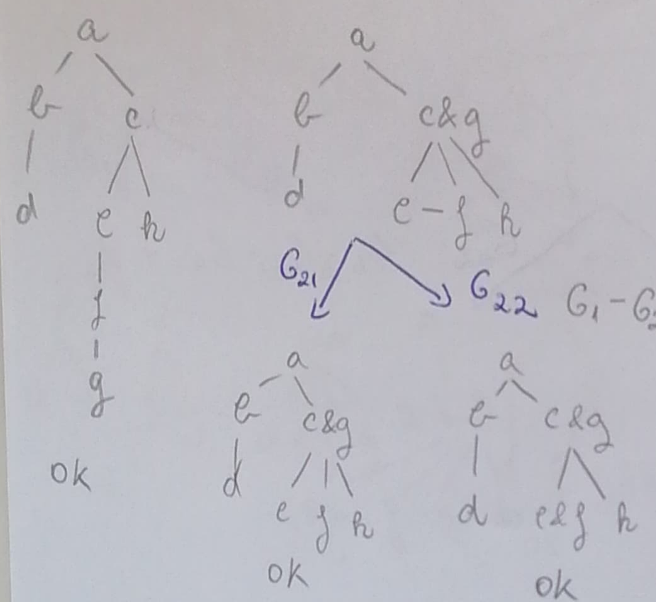
6.



=> polinomul: $k(k-1)^7 - k(k-1)^6 + k(k-1)^5 = k(k-1)^5(k^2-2k+1-k+1+1)$

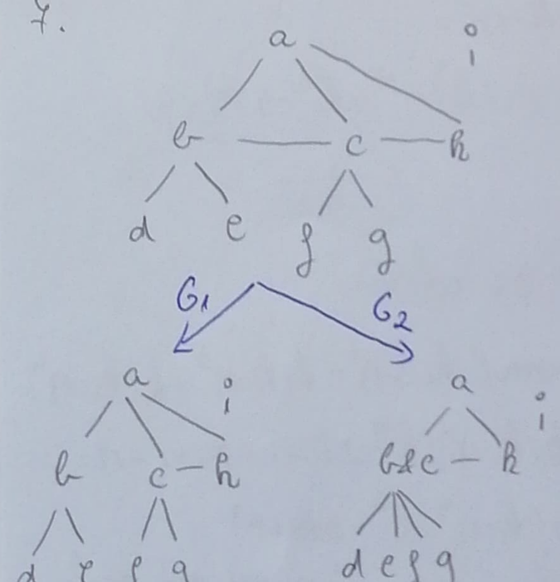
$= k(k-1)^5(k^2-3k+3)$ => rădăcini 0,1
↳ nu are răd. reale

=> nr. cromatic 2 => nr. de colorări: 2



$G_1 - G_{21} + G_{22}$

7.



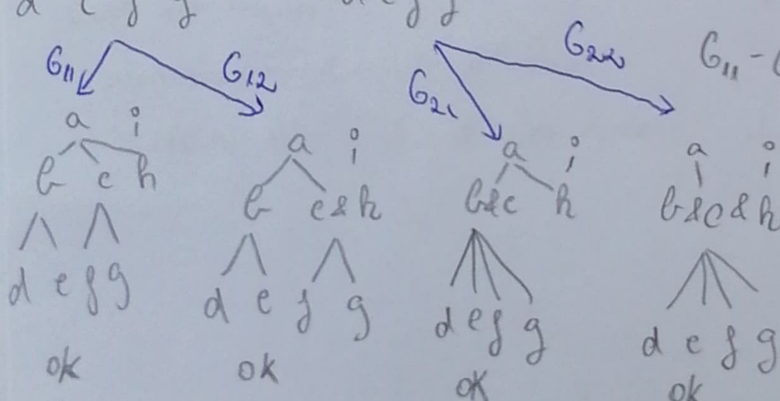
=> polinomul:

$k(k-1)^7 \cdot k - k(k-1)^6 \cdot k - k(k-1)^6 \cdot k + k(k-1)^5 \cdot k =$

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

$= k^2(k-1)^5(k^2-4k+4)$ => rădăcini 0,1,2
=> nr. cromatic 3

nr. colorări: $9 \cdot 2^5 \cdot 1 = 9 \cdot 32 = 288$



$G_{11} - G_{12} - G_{21} + G_{22}$

8.

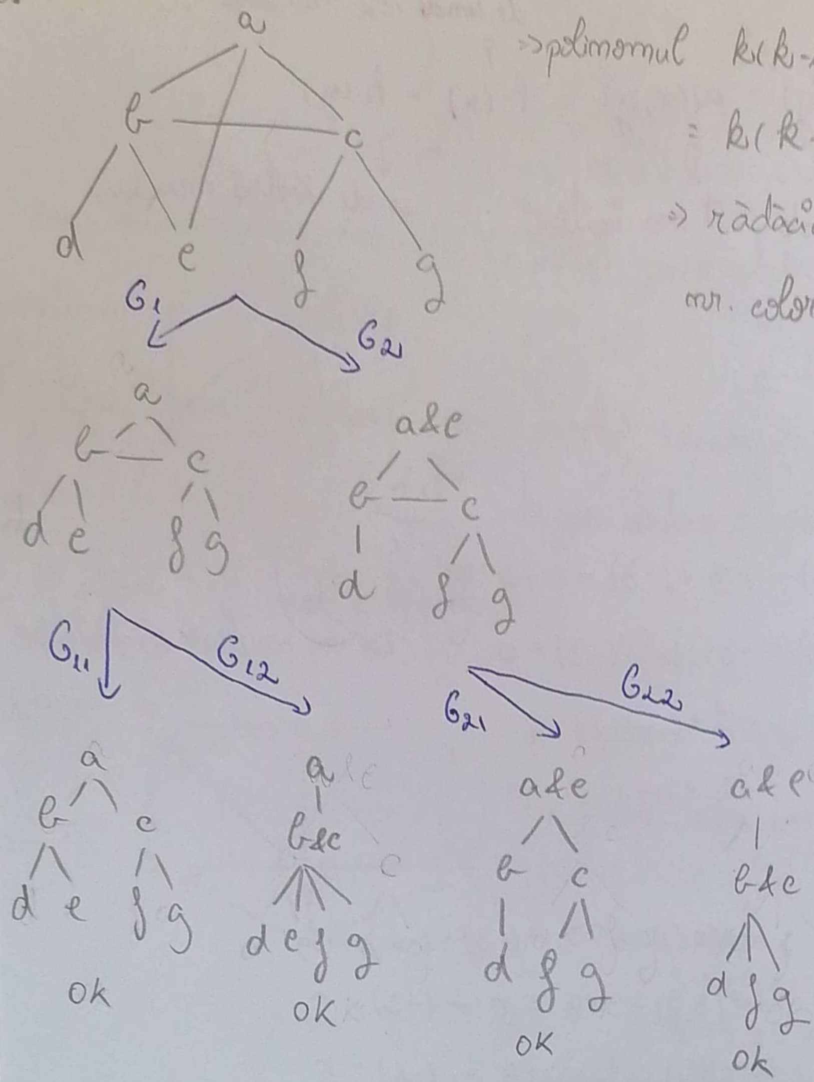
$\Rightarrow$ polinomul $k(k-1)^6 - k(k-1)^5 - k(k-1)^5 + k(k-1)^4 =$

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

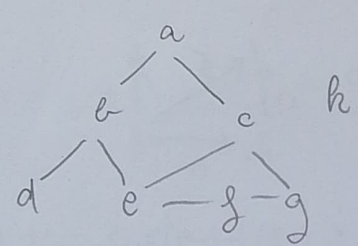
$= k(k-1)^4 (k^2 - 4k + 4)$

$\Rightarrow$ rădăcini $0, 1, 2 \Rightarrow$ nr. crom. 3

nr. colorări :



9.



REPONDERARE

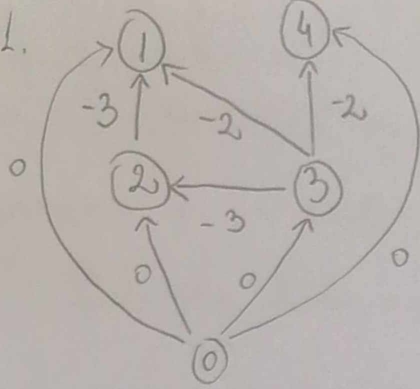
$$\hat{\omega}(x,y) = \omega(x,y) + h(x) - h(y)$$

de unde iese muchia

cât era inițial

unde intră muchia

1.



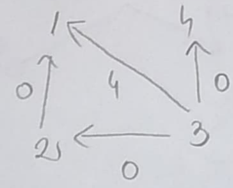
0 -> 1 : -6

0 -> 2 : -3

0 -> 3 : 0

0 -> 4 : -2

=>



$\hat{\omega}(2,1) = -3 + (-3) - (-6) = 0$

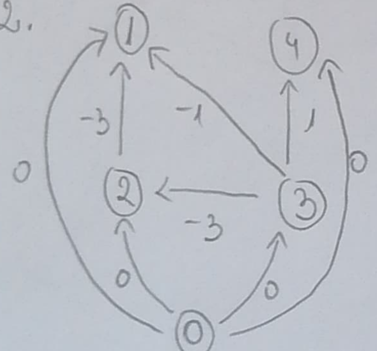
$\hat{\omega}(3,2) = -3 + 0 - (-3) = 0$

$\hat{\omega}(3,1) = -2 + 0 - (-6) = 4$

$\hat{\omega}(3,4) = -2 + 0 - (-2) = 0$

(se iau muchiile existente pe rând)

2.



0 -> 1 : -6

0 -> 2 : -3

0 -> 3 : 0

0 -> 4 : 0

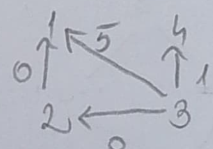
$\hat{\omega}(2,1) = -3 + (-3) - (-6) = 0$

$\hat{\omega}(3,2) = -3 + 0 - (-3) = 0$

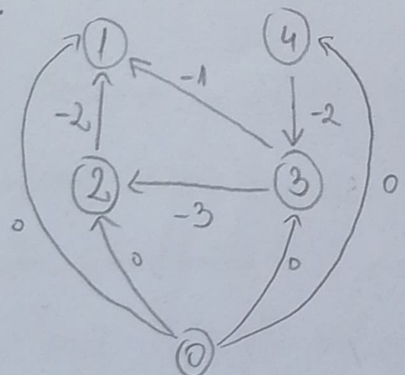
$\hat{\omega}(3,1) = -1 + 0 - (-6) = 5$

$\hat{\omega}(3,4) = 1 + 0 - 0 = 1$

=>



3.



0 -> 1 : -7

0 -> 2 : -5

0 -> 3 : -2

0 -> 4 : 0

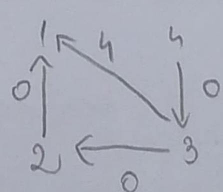
$\hat{\omega}(2,1) = -2 + (-3) - (-5) = 0$

$\hat{\omega}(3,2) = -3 + (-2) - (-5) = 0$

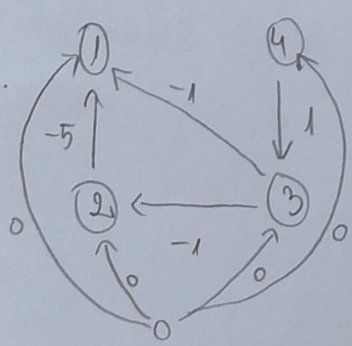
$\hat{\omega}(3,1) = -1 + (-2) - (-7) = 4$

$\hat{\omega}(4,3) = -2 + 0 - (-2) = 0$

=>



4.



0 -> 1 : -6

0 -> 2 : -1

0 -> 3 : 0

0 -> 4 : 0

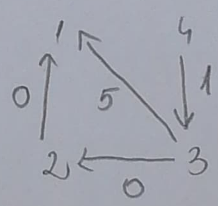
$\hat{\omega}(2,1) = -5 + (-1) - (-6) = 0$

$\hat{\omega}(3,2) = -1 + 0 - (-1) = 0$

$\hat{\omega}(3,1) = -1 + 0 - (-6) = 5$

$\hat{\omega}(4,3) = 1 + 0 - 0 = 1$

=>



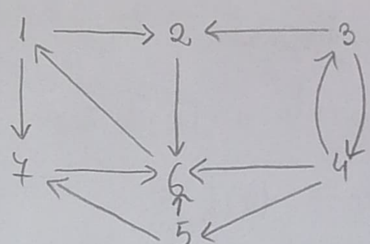
DEMONSTRAȚII

1. Un graf conex minimal cu n noduri are $n-1$ muchii (graf conex minimal = dacă eliminăm o muchie își pierde proprietatea de conexitate). Orice graf conex minimal e un arbore.

Dacă prin eliminarea unei muchii => 2 componente => graf conex minimal. Un graf fără cicluri, cu $n-1$ muchii este un arbore => graf conex minimal.

=> afirmațiile sunt echivalente

12.



a) componente tare conexe: 1,2,6,7   3,4   5

b) drum: nu se pot repeta noduri (path)

$3\to6$: $3\to4\to6$
$3\to2\to6$
$3\to4\to5\to6$

c) lanț (walk): se pot repeta noduri $3\to6$
$3\to4\to3\to2\to6$
$4\to3\to4\to6$

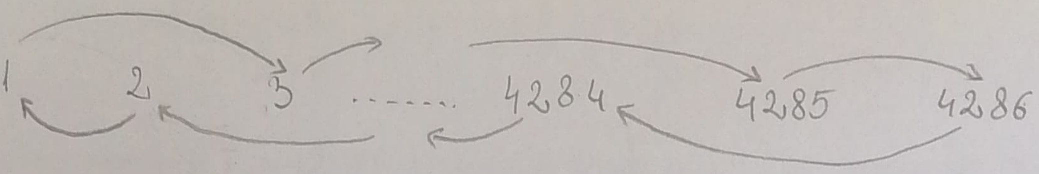
d) cicluri: $2\to6\to1\to2$
$1\to7\to6\to1$

2. Noțiunea de arbore are la bază un graf conex minimal. Într-un graf conex minimal oricare 2 noduri sunt conectate printr-un lanț simplu (nu $\exists$ cicluri, orice muchie am elimina => 2 componente conexe) => afirmațiile sunt echivalente.

* dacă nu există cicluri se pot trasa oricâte lanțuri fără a se repeta noduri.

3. orice 2 vârfuri sunt conectate printr-un lanț simplu = arbore
orice muchie ștearsă rezultă în 2 componente conexe = graf conex minimal
arbore = graf conex minimal, afirmațiile sunt echivalente

4.



✓ hamiltonian: da, luăm nodurile impare, apoi cele pare

✗ eulerian: nu, (1, 3 și 4284, 4286 au grad impar)

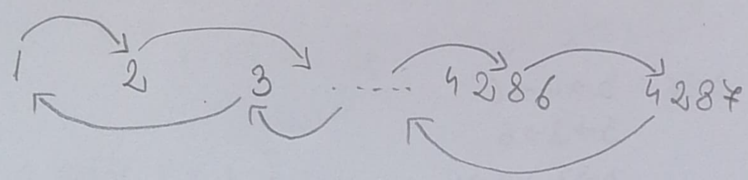
✓ cuplaj perfect: combin 2 câte 2

5. acum doar până la 4287

✗ cuplaj perfect: nr. impar, nu se mai formează perechi

✗ eulerian: avem mai mult de 2 noduri cu grad impar

✓ hamiltonian: da, explicația ca și sus



6.

1 2 3 1109 1110 1111

✗ eulerian

✓ hamiltonian

✗ cuplaj perfect

8. Singurele grafuri aciclice sunt arbori (sau arbori cu noduri izolate). Pp. că complementarul unui arbore nu conține cicluri. Nr. muchii arbore: $n-1$, n noduri.
=> împreună pot avea maxim $2(n-1)$ muchii pt. a fi ambele aciclice
Dar ele au împreună $\frac{n(n-1)}{2}$ muchii $> 2(n-1)$ => unul sigur conține un ciclu
(pt. $n >= 5$)

10. $n \geq 3$ unde m - muchii pt. un graf planar
n - vârfuri
? $m \leq 3n-6$

11. $\delta(G) \leq 5$ pt. orice graf planar
-> dacă $n \leq 6$ => orice vârf are grad ≤ 5 => $\delta(G) \leq 5$
-> dacă $n > 6$ notăm $D = \sum_{v \in V} d(v)$ => $D = 2m \leq 2(3n-6) = 6n-12$
dacă $\delta(G) \geq 6$ atunci $\sum_{v \in V} d(v) \geq \sum_{v \in V} 6 = 6n$ contradicție
=> $\delta(G) \leq 5$ are loc

7. $3k \geq 6$ cu grade $1, 2, 3, 1, 2, 3, \dots, 1, 2, 3$. Dem. că G are un ciclu
-> pp. că ∄ cicluri. avem k vf. grad 1, k vf. grad 2, k vf. grad 3.
-> graful aciclic cu $3k$ vârfuri are $k+1$ vf. grad 1, k vf. grad 2 și $k-1$ vf grad 3
e necesar să facem un ciclu pt. a respecta gradele => pp. falsă

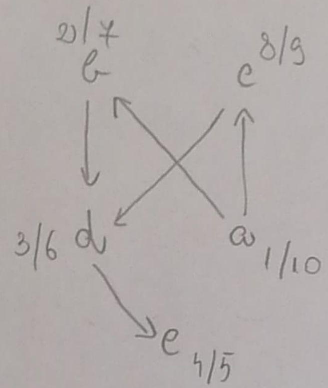
9. Ce se obține dacă înmulțim o matrice de incidență cu transpusa ei?

SORTARE TOPOLOGICĂ

1. Dacă graful conține un ciclu => nu se poate sorta topologic.

ciclu: 1 -> 2 -> 3 -> 1

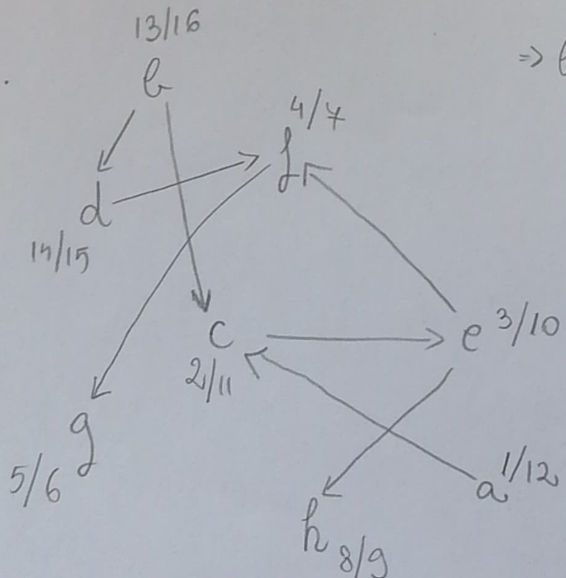
2.



=> a, c, b, d, e

Se face dfs, apoi scriem nodurile în ordine inversă (ordonăm descrescător după al doilea nr - aka momentul când s-au terminat de parcurs).

3.



=> b, d, a, c, e, h, f, g