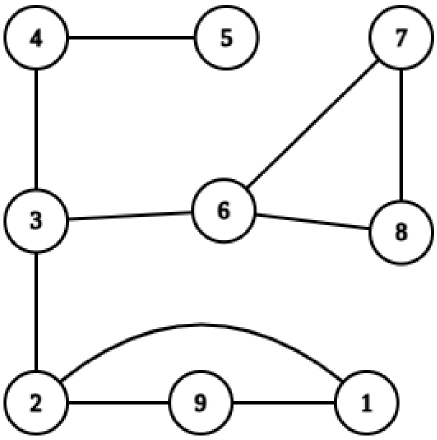


Pentru graful din imaginea din stânga:

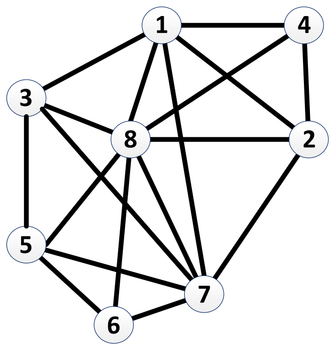
1) Care sunt nodurile critice ?

2) Care sunt muchiile critice ?

3) Exemplificați cum funcționează df(3) până când sunt vizitate 7 vârfuri, ilustrând si arborele df asociat; vecinii unui vârf se consideră în ordine lexicografică

4) Puneți ponderi pe muchii astfel încât costul unui arbore parțial de cost minim în graful obținut să fie 42.

5) Care este distanta de editare între cuvintele “examen” si “restanta” ? Justificați

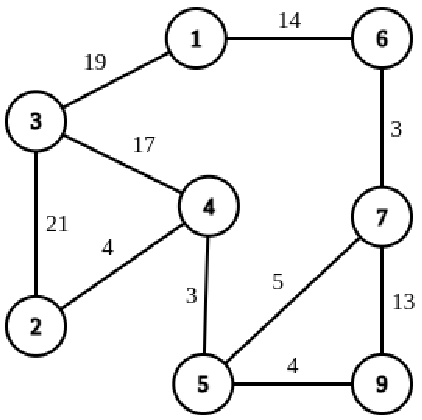


6) Descrieți algoritmul de 6-colorare a vârfurilor unui graf neorientat conex planar și **exemplificați** acest algoritm pentru graful alăturat. Justificați și de ce acest graf este planar.

Barem 0.5 fiecare problema 1)-6)

7) Meșterel vrea să asambleze o megamasina și a citit cu atenție instrucțiunile. A identificat cele n acțiuni care trebuie sa le facă și perechi (i,j) de acțiuni care depind direct una de cealaltă (acțiunea j se poate face după ce activitatea i s-a terminat). Meșterel vrea sa faca activitatea p care este activitatea sa preferată. Pentru acest lucru el trebuie sa faca toate activitățile de care p depinde direct sau indirect.
Ajutați-l pe Mesterel găsind toate activitățile pe care trebuie sa le facă și o ordine în care le poate face. (De restul activităților se vor ocupa prietenii săi).
Descrieți cum puteți rezolva aceasta problemă și complexitatea soluției. Dacă exista mai multe soluții/implementării puneți accent pe discuția despre când ar trebui sa folosim o soluție si când alta.
Barem: **1,5p** (0,75 soluție corectă + 0,75 discuții complexitate + complexitate optimă)

Pentru graful din stânga (vecinii se consideră în ordine lexicografică):



1) Exemplificați Dijkstra din 4, opriti-va după ce ați găsit distanța către 6

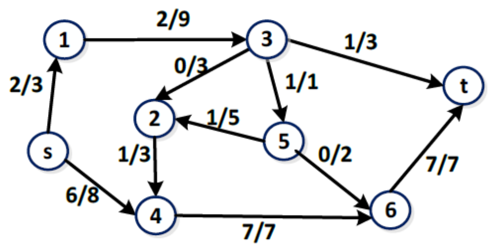
2) Cum funcționează algoritmul lui Kruskal ? Exemplificați alegerea primelor 6 muchii.

3) Este graful bipartit ? Dacă nu eliminați un număr minim de muchii astfel încât el sa devina bipartit. Care este numărul maxim de muchii ale unui graf bipartit cu 9 vârfuri? Justificați.

4) Există lanț eulerian în graf? Dacă nu adăugați număr minim de muchii astfel incat graful format sa admită lanț eulerian, descriind și strategia după care ați adăugat muchiile. Indicați un lanț eulerian în graful obținut. Enunțați o condiție necesară și suficientă ca un graf neorientat să aibă un lanț eulerian.

0.5p fiecare problema 1)-4)

5) Definiți noțiunile de flux, tăietură, tăietură minimă și lanț nesaturat/drum de creștere. Ilustrați pașii algoritmului Ford-Fulkerson pentru rețeaua din figura următoare (pe un arcul e sunt trecute valorile f(e)/c(e) reprezentând flux/capacitate), pornind de la fluxul indicat și alegând la fiecare pas un s-t lanț f-nesaturat de lungime minimă (algoritmul Edmonds-Karp)**.** Indicați o tăietură (s-t tăietură) minimă în rețea (se vor indica vârfurile din bipartiție, arcele directe, arcele inverse) și determinați capacitatea acestei tăieturi. Mai există și o altă s-t tăietură minimă în această rețea? Justificați răspunsurile **(1p)**

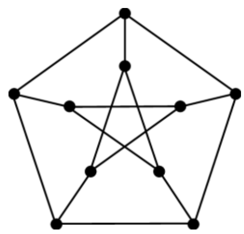


6) a) Fie G un graf planar conex cu n>3 noduri și m muchii care conține cicluri și fie g lungimea minimă a unui ciclu din G. Arătați că

$m \cdot (g-2) \le g \cdot (n-2)$.

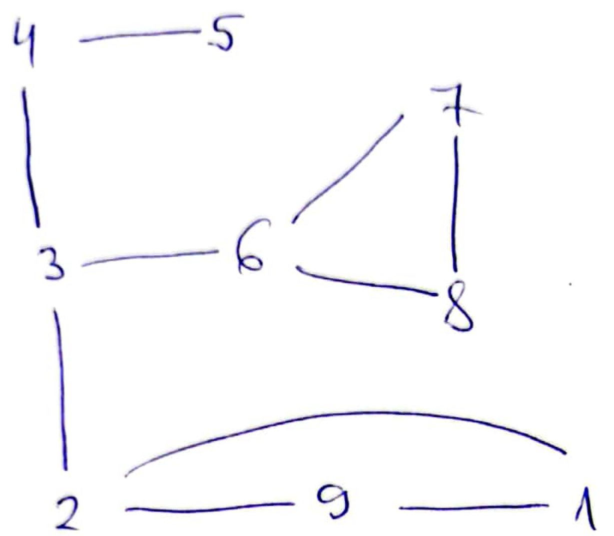
b) Arătați că graful lui Petersen (din figura alăturată) nu este planar.

(1,5p)

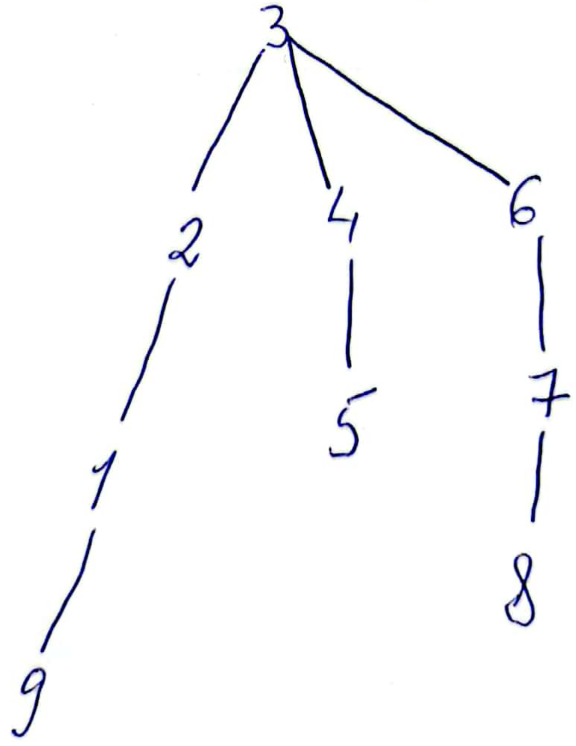


Dani Bacin

Partea 1 - Examen



1. noduri critice: 4, 3, 6, 2. dacă prin eliminarea nodului, se strică conexitatea
2. muchii critice: (4,5), (3,4), (3,6), (3,2) deoarece dacă le eliminăm, atunci graful are 2 comp conexe.
3. df(3).



~~adân~~ parcurgerea în adâncime se realizează recursiv.
~~pri~~ și apelează pt vecini.

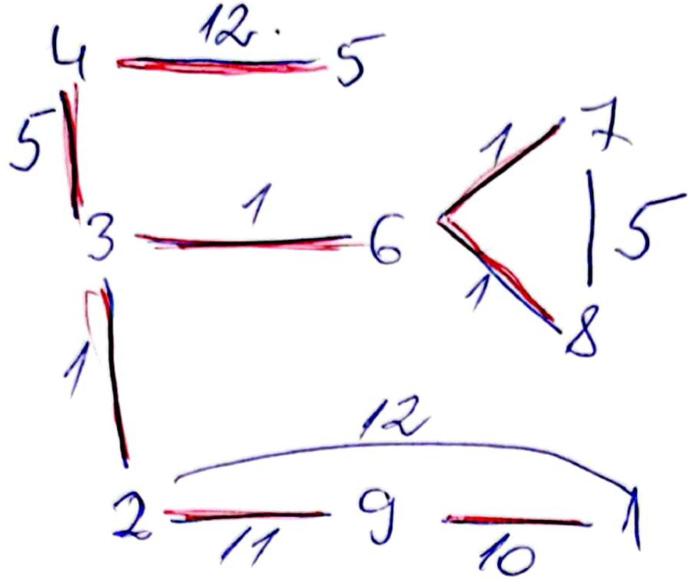
$df(3) \rightleftarrows df(2) \rightleftarrows df(1) \rightleftarrows df(9)$

$df(3) \rightleftarrows df(4) \rightleftarrows df(5)$

$df(3) \rightleftarrows df(6) \rightleftarrows df(7) \rightleftarrows df(8)$

2

4)



$10+11+1+1+1+1+5+12 =$

$= 21+4+5+12 = 30+12 = 42.$

putem pune oricât pe muchiile critice pentru că ele vor fi în orice APCM.

Am pus ca nou APCM-ul rezultat folosind Alg. lui Prim din 5.

3

5) distanta Levenstein: (o problemă clasică de prog. dinamică)

Formula este pentru 2 nume a și b, iar i și j sunt pentru substringurile sale.

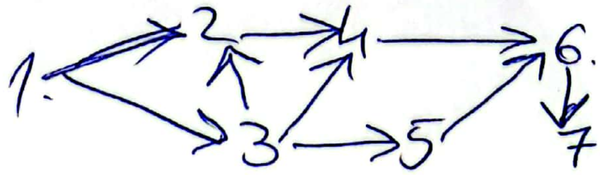
$$d_{ij} = \begin{cases} j, & i = 0. \\ i, & j = 0. \\ d_{i-1\,j-1}, & a[i] == b[j]. \\ 1 + \min \begin{cases} d_{i-1\,j} & \text{//~~stergere~~ clona.} \\ d_{i\,j-1} & \text{//~~stergere~~ inserare în a.} \\ d_{i-1\,j-1} & \text{//înlocuire.} \end{cases} \end{cases}$$

Folosind un alg. recursiv împreună cu un ~~este 6~~. map, am descoperit că dist levenstein dintre „examen" și „restanta" este 6.

---

7. (aici este la fel cu problema pe care v-am trimis-o pentru interviu).

Aici putem traduce într-o problemă de grafuri orientate. Practic condiția necesară este ca activitățile de care depinde p să nu aibă o dependință ciclică. (DAG).



4

dacă $p=3$ atunci vom apela o funcție care merge recursiv din p, în vecinii lui pana gasește un nod care are gradul extern 0, în cazul asta e 7. Afisam 7 și eliminam din graf.

Practic, e o sortare topologică unde eliminam nodurile care nu fac parte din arborele de parcurgere a lui p.

Această soluție funcționează doar dacă nu există dependințe ciclice în graful ramas.

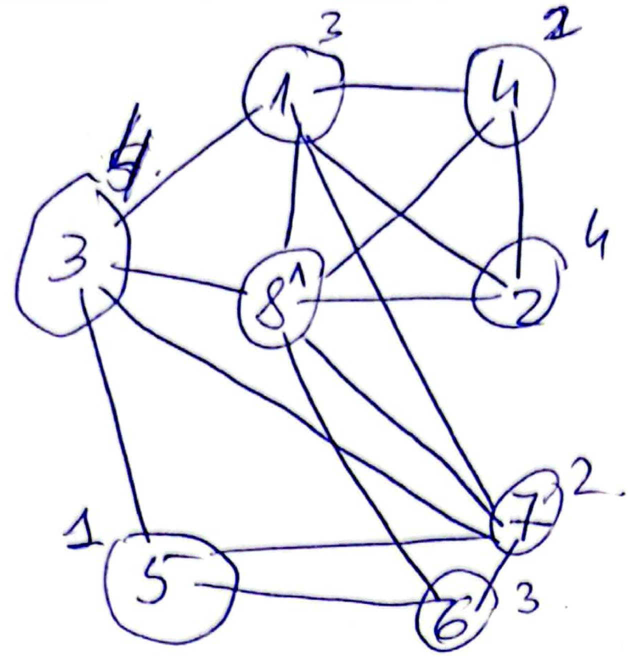
Complexitatea este $O(n+m)$ pt sort topologică

Pt eliminarea unui nodului avem maxim.

$O(n+m)$.

deci complx. totală este $O(n+m)$ pentru alg. descris.

6.1



alegem. cate un vf cu grad < 5 și

colorăm (G - x)

apoi colorăm pe x cu o culoare rămasă.

x = 3 → colorăm (G - 3) → col. (G - {2,3})

ș.a.m.d.

ordinea:

8 7 6 5 1 4 2 3

6.

Dani Baciu

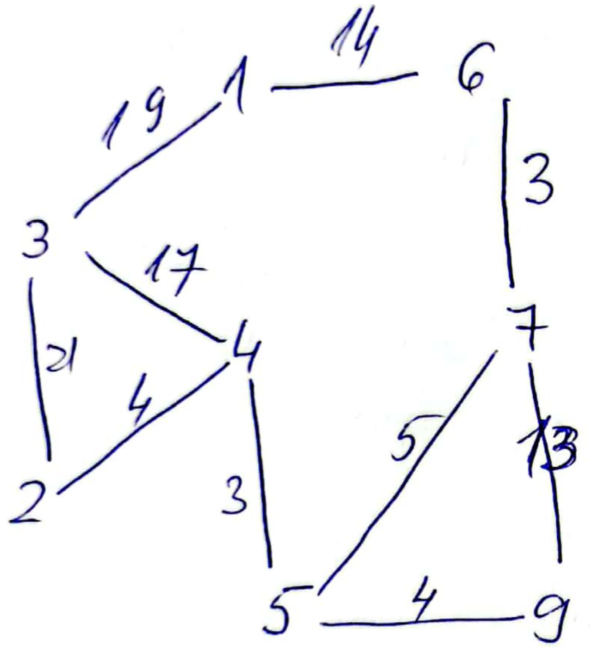
Q: 4,

1. Dijkstra (4).

alegem X un vf din coada.
pentru fiecare vecin verificam relatia:
dc $(d[x] + w(x,y) < d[y])$ atunci:

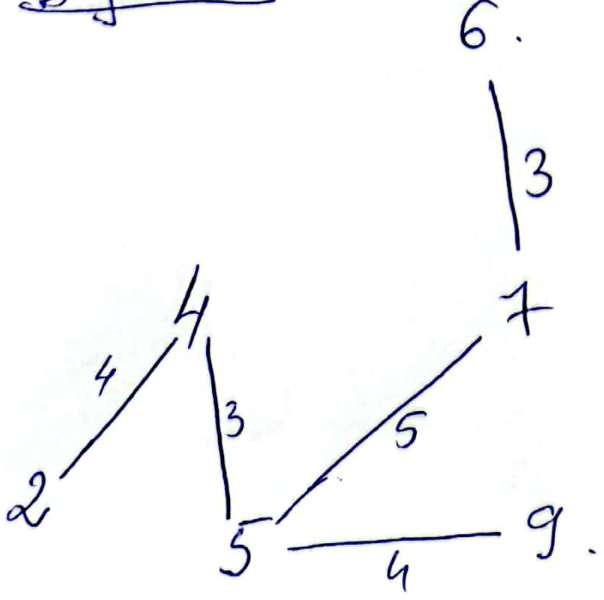
$d[y] = d[x] + w(x,y)$.

tata $[y] = x$.



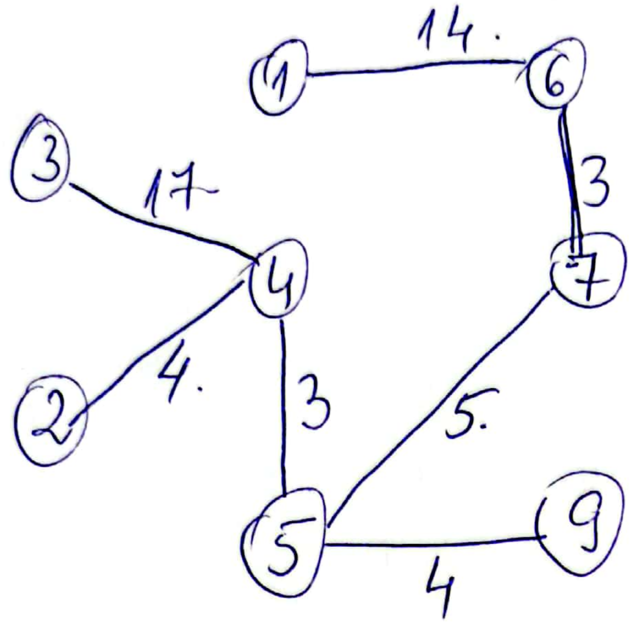
Dijkstra ia de fiecare dată nodul cu eticheta minimă din vectorul de distanțe

Dijkstra:



$dist(4,6) =$
$= 3+5+3 = 11.$

2) Kruskal: la fiecare pas alegem o muchie de cost minim ce nu închide un ciclu (din comp conexe disjuncte) și o adăugăm la arborele creat.



Acesta este APCM al rezultat după alg lui Kruskal.

Pas 1: alegem (6,7).

2: alegem (4,5).

3: alegem (5,9)

4: alegem (2,4).

5: alegem (5,7)

6: alegem (1,6).

7: alegem (3,4)

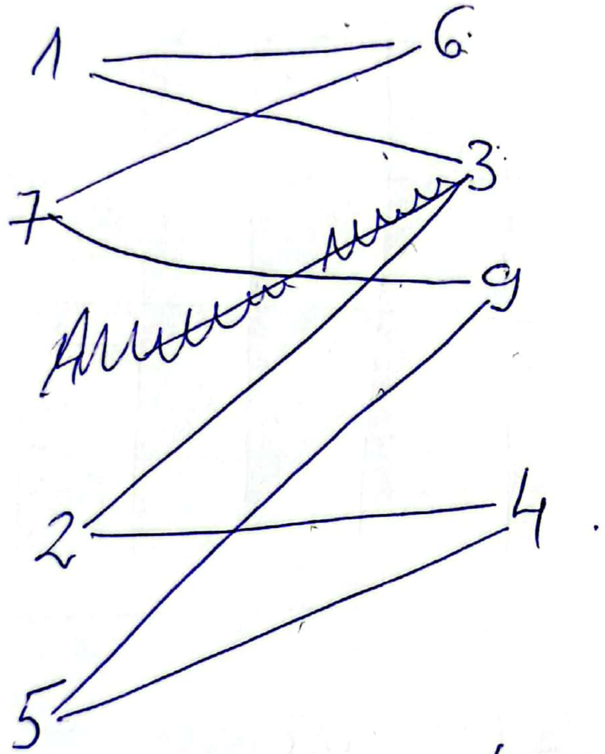
STOP (final algoritm)

3. Graful NU este bipartit deoarece are cicluri impare. Pentru a-l face bipartit trebuie să eliminăm ciclurile impare.

Nr minim de muchii care trb elim este 2 și acestea pot fi ~~oricare~~ unele din ciclurile (3, 4, 2) și (7, 9, 5) (una din fiecare ciclu)

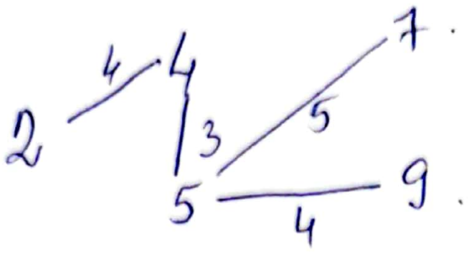
Ex: eliminăm (3,4) și (5,7).

Pornim din 1:



Nu putem elimina (3,4) și (5,7) ptc. există un ciclu impar (1,6,7,9, 5,4,3,1)

Continuare 1:



Nod 8 nu există

		1	2	3	4	5	6	7	8	9
d/tata	1:	00/0.	4/4	17/4.	0/0	3/4	00/0	00/0	~~00/0~~	00/0
	2:		-		—	—		8/5		7/5.
	3:		—							
	4:									—
	5.						11/7.	—		
							—			

Continuare 3:

Nr max de vârfuri este atunci când avem 4 noduri într-o parte și 5 în cealaltă

$m = 4 \cdot 5$ (de la fiecare nod ducem muchii către toate nodurile din partea cealaltă)

$\sum d_i = 2m = \cancel{4 \cdot 5 \cdot 2} = 40$

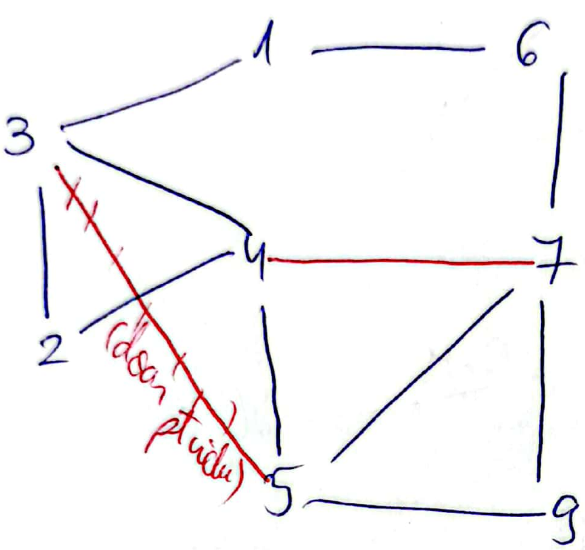
$\Rightarrow m = 20.$

4. Un graf are un lanț eulerian doar dacă e conex, fără vf izolate și toate gradele nodurilor sunt pare.

G are gradele nodurilor: $\{ 2_1, 2_2, 3_3, 3_4, 3_5, 2_6, 3_7, 2_9$

Avem ~~3~~ 4 vf de gradul 3 => trb să adăugăm 2 muchii pt a avea un ciclu eulerian.

Lanț eulerian:



3 1 6 7 5 9 7 4 3 2 4

$\boxed{\text{unde adăugăm doar o muchie } (7,4)}$

ca să avem lanț => max. 2 vf cu grad impar care au muchie directă (eliminăm muchia dintre ele în cazul ciclu)

Avem ciclul: 6 7 5 9 7 4 3 2 4 5 3

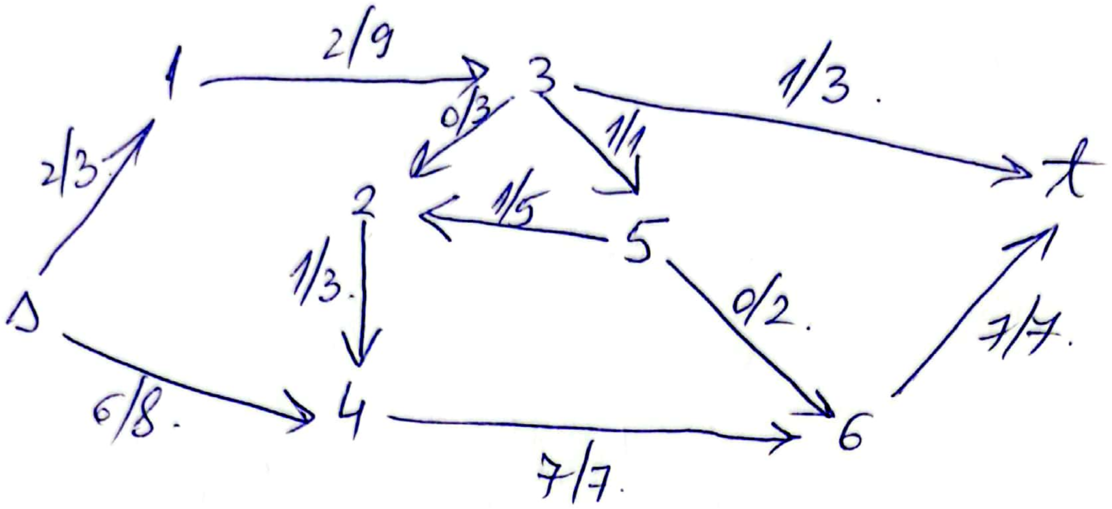
---

5. flux = cantitate de informație ce poate fi transmisă din sursă în destinație a.î. în fiecare nod intermediar cant care intră e egală cu cea care iese.

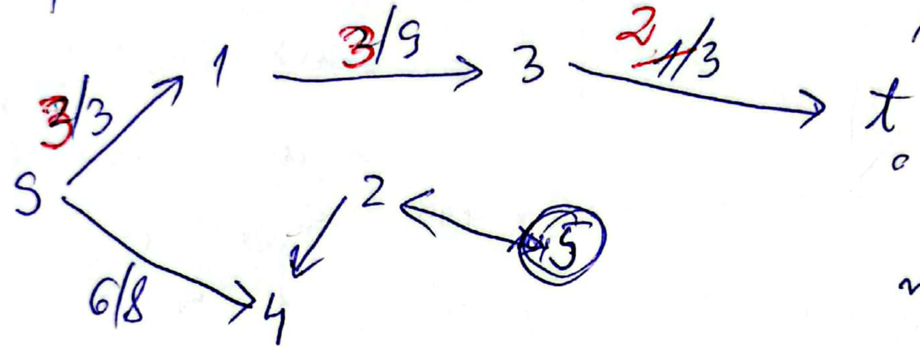
Tăietură = o tăietură în rețea e o bipartiție a nodurilor $G = X \cup Y$, $X \cap Y = \emptyset$ unde sursa $\in X$ și dest $\in Y$.

tăietura minimă = o tăietură a î suma muchiilor de la X către Y e minimă

lanț nesaturat = lanț pe care mai pot trimite flux

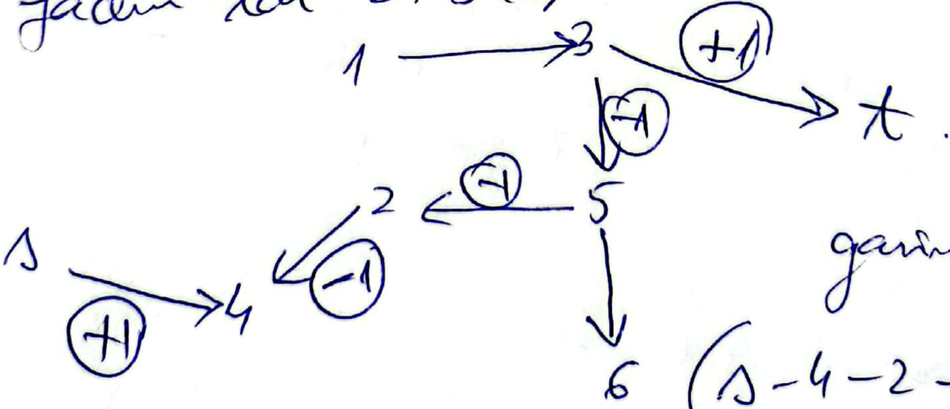


1.) facem BFS din S.



unde am găsit lanțul s-1-3-t. pe care mai pot trimite o unitate.

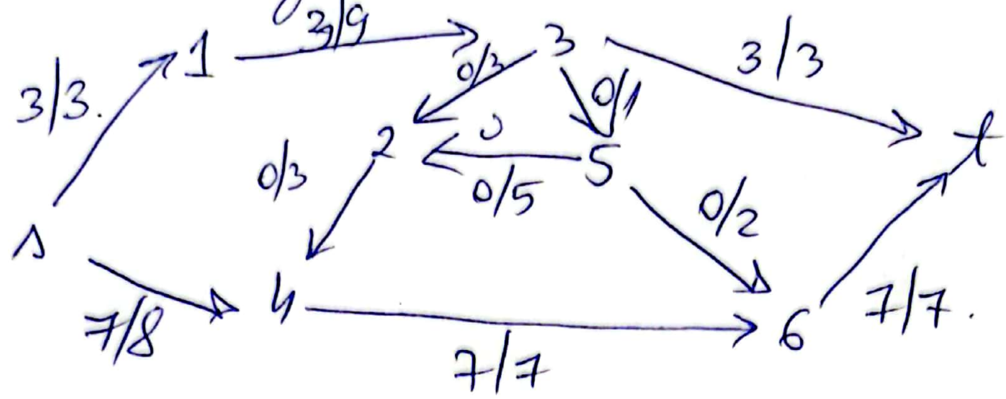
2) facem iar BFS(s).



(s-4-2-5-3t).

găsim lanțul (s-4-2-5-3-t) unde mai trimit o unitate.

Acum avem graful rezultat:



Min-cut: $K = \{\{s,4\}, \{1,2,3,5,6,t\}\}$
care are costul 10.

Mai există și alte tăieturi minime decât cele
arcele (3,t) și (6,t).
arce directe: $\{(s,1)\ (4,6)\}$.
arce inverse: $\{(2,4)\}$.

6. a) $\sum d_i = 2m$

$\sum f_i = 2m$ (suma gradelor fețelor)

ciclu de lungime g minim ⇒ gradul unei fețe este minim g.

$f_i \geq g$.

grade minim.

$\Rightarrow \sum f_i \geq g \cdot |F|$

nr. de fețe.

$\Rightarrow 2m \geq g \cdot |F|$.

Euler: $n - m + |F| = 2 \Rightarrow$

$\Rightarrow |F| = 2 + m - n$.

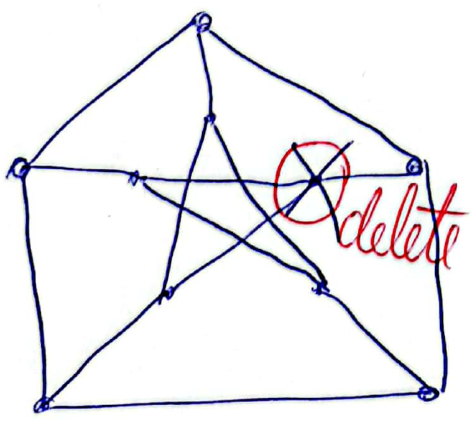
$2m \geq 2g + gm - g \cdot n$.

$m(2-g) \geq g(2-n)$. $\mid \cdot (-1)$.

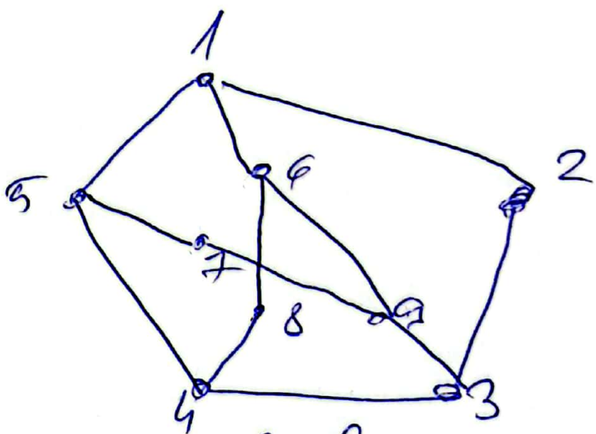
$(g-2)m \leq g(n-2)$.

---

b). Un graf NU este planar dacă are ca și ~~graf parțial pe K_5 sau $K_{3,3}$~~. graf izomorf K_5 și $K_{3,3}$.
Acest graf seamănă cu $K_{3,3}$ așa că îl vom transformăm. Încercăm să elim muchia să îl transformăm un nod din centru.

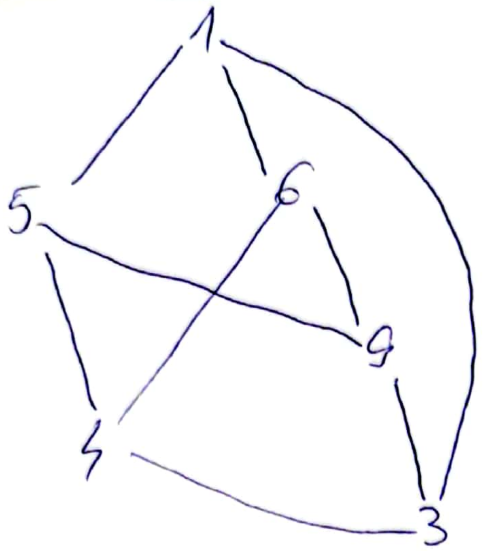


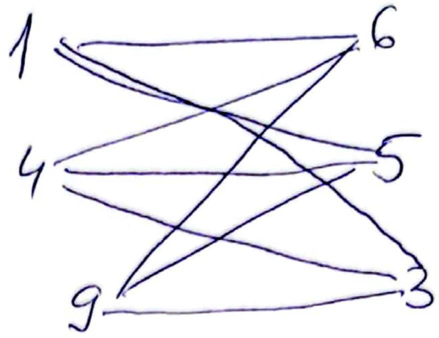
$\rightarrow$



și îl facem bipartit eliminând 4, 8, 2

1





$K_{3,3}$ =>

=> Graful NU este planar =>

=> nu poate fi desenat.

Graful Petersen are 15 muchii și 10 vf.