Zadaća 4

iz predmeta Diskretna matematika

Prezime i ime: Hamzić Huso

Broj indeksa: 18305

Grupa: DM2 [Pon 15.00]

Odgovorni asistent: Šeila Bećirović

Elektrotehnički fakultet Sarajevo

1. Data su tri neusmjerena grafa:
G_1 = {{x1, x2, x3, x4, x5, x6, x7, x8}, {{x1, x3}, {x1, x4}, {x1, x5}, {x2, x4}, {x2, x6}, {x2, x7}, {x2, x8}, {x3, x5}, {x3, x7}, {x3, x8}, {x4, x7}, {x5, x6}, {x5, x8}, {x7, x8}}}
G_2= {{x1, x2, x3, x4, x5, x6, x7, x8}, {{x1, x2}, {x1, x4}, {x1, x6}, {x1, x8}, {x2, x3}, {x2, x7}, {x2, x8}, {x3, x5}, {x3, x7}, {x3, x8}, {x4, x5}, {x4, x8}, {x5, x7}, {x6, x7}}}
G_3= {{x1, x2, x3, x4, x5, x6, x7, x8}, {{x1, x3}, {x1, x4}, {x1, x6}, {x1, x8}, {x2, x3}, {x2, x4}, {x2, x7}, {x2, x8}, {x3, x4}, {x3, x7}, {x4, x5}, {x5, x7}, {x5, x8}, {x6, x7}}}

Za ove grafove potrebno je uraditi sljedeće:
-Predstavite ih pomoću matrica susjedstva i pomoću listi susjedstva.
-Utvrdite ima li među ovim grafovima nekih koji su međusobno izomorfni. Ukoliko neka dva jesu izomorfna (ako takvih parova ima), prikažite kako glasi izomorfizam između njih. Ukoliko neka dva nisu izomorfna (ako takvih parova ima), argumentirano objasnite zašto nisu.
-Utvrdite ima li među ovim grafovima planarnih grafova. Za one koji su planarni (ako ih ima), nacrtajte ih tako da im se grane ne presjecaju. Za one koji nisu planarni (ako ih ima), argumentirano objasnite zašto nisu.
-Pronađite hromatske brojeve za ova tri grafa. Odgovor mora biti argumentiran.

Najprije predstavimo zasebne grafove preko matrica i lista susjedstva:

	x1	x2	x3	x4	x5	x6	x7	x8
x1			1	1	1			
x2				1		1	1	1
x3	1				1		1	1
x4	1	1					1	
x5	1		1			1		1
x6		1			1			
x7		1	1	1				1
x8		1	1		1		1	

$$G_1 = (\{x_3, x_4, x_5\}, \{x_4, x_6, x_7, x_8\}, \{x_1, x_5, x_7, x_8\},$$
$$\{x_1, x_2, x_7\}, \{x_1, x_3, x_6, x_8\}, \{x_2, x_5\}, \{x_2, x_3, x_4, x_8\}, \{x_2, x_3, x_5, x_7\})$$

	x1	x2	x3	x4	x5	x6	x7	x8
x1		1		1		1		1
x2	1		1				1	1
x3		1			1		1	1
x4	1				1			1
x5			1	1			1	
x6	1						1	
x7		1	1		1	1		
x8	1	1	1	1				

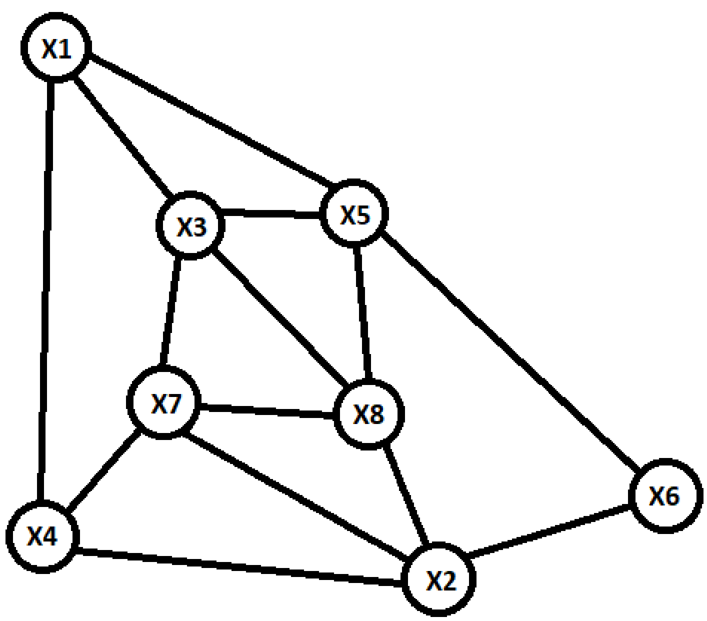
$$G_2 = (\{x_2, x_4, x_6, x_8\}, \{x_1, x_3, x_7, x_8\}, \{x_2, x_5, x_7, x_8\},$$
$$\{x_1, x_5, x_8\}, \{x_3, x_4, x_7\}, \{x_1, x_7\}, \{x_2, x_3, x_5, x_6\}, \{x_1, x_2, x_3, x_4\})$$

	x1	x2	x3	x4	x5	x6	x7	x8
x1			1	1		1		1
x2			1	1			1	1
x3	1	1		1			1	
x4	1	1	1		1			
x5				1			1	1
x6	1						1	
x7		1	1		1	1		
x8	1	1			1			

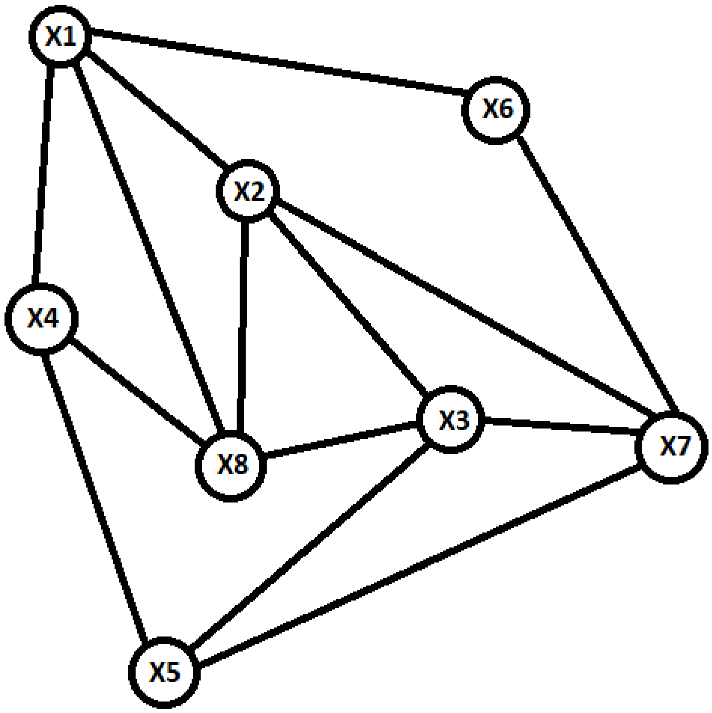
$$G_3 = (\{x_3, x_4, x_6, x_8\}, \{x_3, x_4, x_7, x_8\}, \{x_1, x_2, x_4, x_7\},$$
$$\{x_1, x_2, x_3, x_5\}, \{x_4, x_7, x_8\}, \{x_1, x_7\}, \{x_2, x_3, x_5, x_6\}, \{x_1, x_2, x_5\})$$

b) Provjerimo izomorfnost grafova:
Broj grana je isti u svim grafovima, što vrijedi i za broj grana i stepene pojedinačnih čvorova pa je potrebno brojati konture grafa. Najprije nacrtajmo sve grafove:

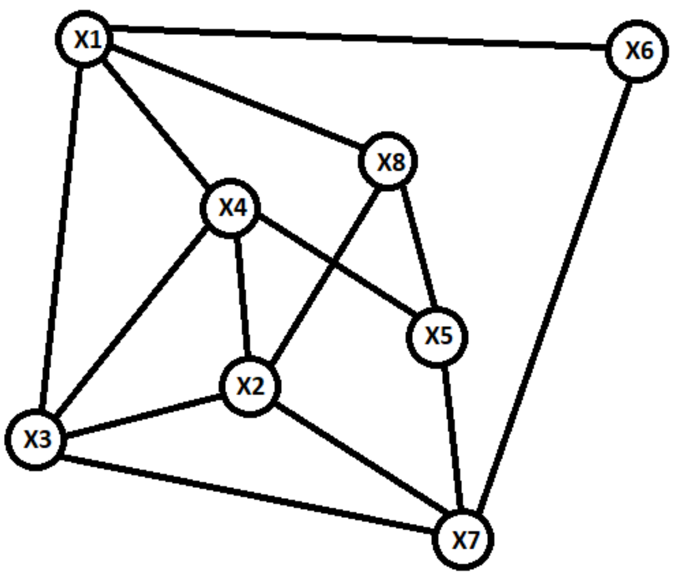
Graf G_1:



Graf G_2:



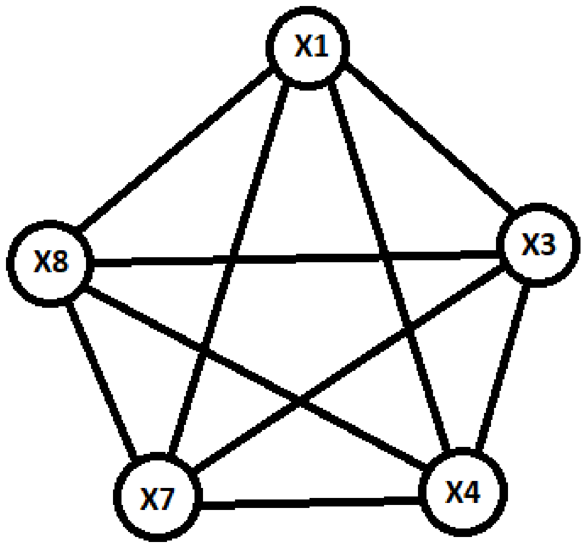
Graf G_3:



Prvi i drugi graf sadrže po 5 kontura dužine 3, dok graf 3 sadrži samo 3 takve konture pa samim time ne može biti izomorfan ni sa jednim od preostala dva grafa.
Ostaje još da provjerimo da li se čvorovi u grafovima 1 i 2 mogu preimenovati tako da se dobije onaj drugi graf.

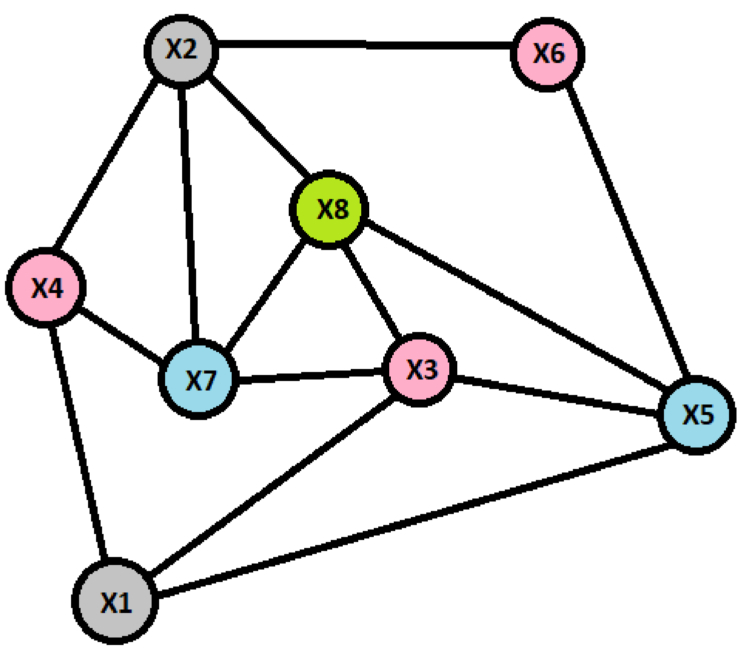
To je zaista i moguće, preimenovanjem čvorova G_1 na način:
f: $x_2 \to x_1; x_6 \to x_6; x_4 \to x_4; x_8 \to x_2; x_7 \to x_8, x_3 \to x_3; x_1 \to x_5; x_5 \to x_7$
zaista dobijemo graf ekvivalentan grafu G_2, pa su G_1 i G_2 međusobno izomorfni grafovi.

c) Već smo pokazali da su grafovi G_1 i G_2 planarni sa skice, potrebno je još dokazati da je graf G_3 neplanaran graf. Vrijedi potreban uslov planarnosti jer je 14 $\leq (3 \cdot 8 - 6)$, odnosno $14 \leq 18$. Međutim, možemo pokazati da je G_3 svodiv na K_5. Ukoliko utopimo čvorove $x_2 \to x_8$, $x_5 \to x_7$ i $x_6 \to x_1$ po Wagnerovoj teoremi, zaista dobijamo K-5 graf i graf G_3 sigurno nije planaran.



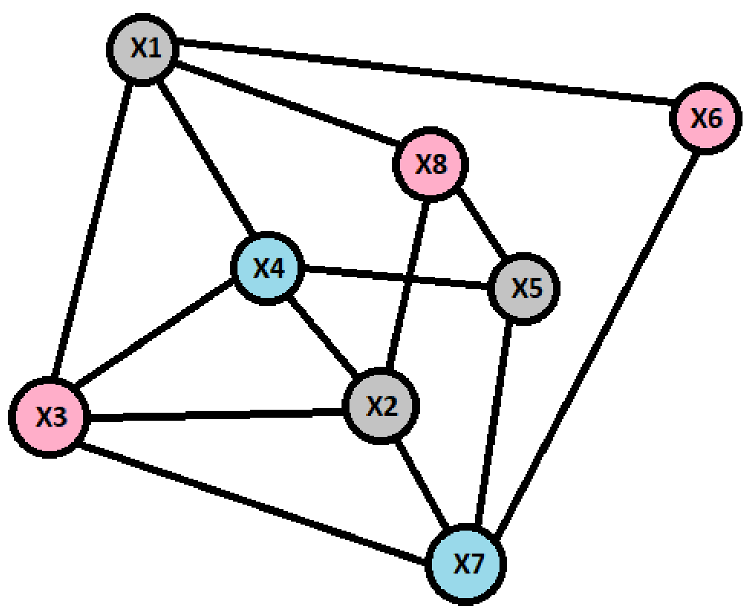
Illuminati confirmed

d) Obojimo grafove G_1 (odnosno i ekvivalentni G_2) pohlepnim algoritmom:



Najprije bojimo čvor x_1 u polaznu sivu boju, nakon toga kako x_2 nema obojenog nijednog susjeda onda njega možemo također obojiti u početnu sivu boju, slijedi čvor x_3 koji ima jednog sivog susjeda pa njega bojimo sljedećom bojom po redu a to je crvena, prelazimo na čvor x_4 za kojeg vrijedi isto kao za x_3 kako ima sivog susjeda bojimo ga u crvenu. Sljedeći čvor je čvor x_5 koji sada ima i sivog i crvenog susjeda pa ga bojimo u sljedeću boju a to je plava, nakon njega na redu je čvor x_6 koji ima plavog i sivog susjeda pa ga možemo obojiti u crvenu te na kraju ne manje bitan čvor x_8 ima i sivog i plavog i crvenog susjeda pa ga moramo obojiti u boju koju do sada nismo vidjeli odnosno Nihadu Alibegoviću dragu, zelenu. Očigledno je hromatski broj grafova G_1 i G_2 jednak 4, imajući u vidu da tako mora biti (manji ili jednak 4) jer su grafovi planarni.

Graf G_3



Prvo bojimo x_1 u početnu sivu boju i nakon toga prelazimo na x_2 koji nema obojenog nijednog susjeda pa i njega možemo obojiti u sivu boju, slijedi čvor x_3 koji ima sivog susjeda pa ga bojimu u sljedeću po redu boju a to je crvena. Na redu je x_4 koji ima i sivog i crvenog susjeda pa ga bojimo u plavu boju. Dolazi x_5 koji nema obojenog niti jednog susjeda pa ga možemo obojiti u početnu sivu boju. Sljedeći je x_6 koji ima sivog susjeda pa ga bojimo u crvenu boju, nakon čega čvor x_7 bojimo u plavu boju jer ima i sivog i crvenog susjeda. Te na kraju čvor x_8 bojimo u crvenu boju jer ima sive susjede.
Očito je hromatski broj grafa 3.

2. Potrebno je povezati 12 lokacija L1 – L12 u računarsku mrežu. Zbog tehnoloških ograničenja, kablove nije moguće razvesti između proizvoljne dvije lokacije. Sljedeći spisak opisuje sve moguće načine kablovskog povezivanja lokacija, pri čemu trojka oblika (Li, Lj, dij) označava

da je moguće spojiti lokacije Li i Lj, i to kablom dužine dij (u metrima):

(L1, L4, 670) (L1, L9, 1290) (L1, L12, 1030) (L2, L5, 1210) (L2, L6, 270) (L2, L9, 660) (L2, L10, 450) (L2, L11, 750) (L3, L4, 990) (L3, L6, 790) (L3, L7, 790) (L3, L9, 1020) (L4, L8, 570) (L5, L6, 1170) (L5, L8, 410) (L5, L12, 850) (L6, L7, 970) (L6, L9, 870) (L6, L10, 930) (L7, L11, 300) (L8, L9, 770) (L8, L12, 1300) (L9, L11, 1190) (L9, L12, 450) (L10, L11, 1100)

Dizajnirajte računarsku mrežu u skladu sa navedenim specifikacijama tako da ukupan utrošak kablova bude minimalan i obavezno naznačite koliko iznosi taj utrošak. Dizajn obavite:
a)Primjenom Kruskalovog algoritma sa bojenjem čvorova;
b)Primjenom optimalnog Kruskalovog algoritma;
c)Primjenom optimiziranog (kvadratnog) Primovog algoritma.
U sva tri slučaja, nemojte crtati odgovarajući graf, nego sve neophodne radnje obavljajte "naslijepo", koristeći samo raspoložive podatke, eventualno uz bilježenje izvjesnih pomoćnih informacija.

a) Najprije ćemo primijeniti Kruskalov algoritam sa bojenjem čvorova:

Grana	Težina	Uzeti	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12
/	/	/	1	2	3	4	5	6	7	8	9	10	11	12
2-6	270	da		1				1						
7-11	300	da							2				2	
5-8	410	da					**3**			**3**				
2-10	450	da										1		
9-12	450	da									4			4
4-8	570	da				**3**								
2-9	660	da									1			1
1-4	670	da	**3**											
2-11	750	da							1				1	
8-9	770	da		**3**				**3**	**3**		**3**	**3**	**3**	**3**
3-7	790	da			**3**									
3-6	790	ne												
5-12	850	ne												
6-9	870	ne												
6-10	930	ne												
6-7	970	ne												
3-4	990	ne												
3-9	1020	ne												
1-12	1030	ne												
10-11	1100	ne												
5-6	1170	ne												
9-11	1190	ne												
2-5	1210	ne												
1-9	1290	ne												
8-12	1300	ne												

Optimalni način povezivanja je koristeći kablove (2,6), (7,11), (5,8), (2,10), (9,12), (4,8), (2,9), (1,4), (2,11), (8,9), (3,7).

Ukupan utrošak kablova iznosi:

$$270 + 300 + 410 + 450 + 450 + 570 + 660 + 670 + 750 + 770 + 790 = \boxed{6090}$$

b) Sada riješimo problem primjenom optimalnog Kruskalovog algoritma:

Grana	Težina	Uzeti	c1/1	c2/1	c3/1	c4/1	c5/1	c6/1	c7/1	c8/1	c9/1	c10/1	c11/1	c12/1
/	/	/	1	2	3	4	5	6	7	8	9	10	11	12
2-6	270	da		c2/2				c2/1						
7-11	300	da							c7/2				c7/1	
5-8	410	da					c5/2			c5/1				
2-10	450	da		c2/3								c2/1		
9-12	450	da									c9/2			c9/1
4-8	570	da				c5/1	c5/3							
2-9	660	da		c2/5							c2/2			
1-4	670	da	c5/1				c5/4							
2-11	750	da		c2/7					c2/2					
8-9	770	da		c2/11			c2/4							
3-7	790	da		c2/12	c2/1									
3-6	790	ne												
5-12	850	ne												
6-9	870	ne												
6-10	930	ne												
6-7	970	ne												
3-4	990	ne												
3-9	1020	ne												
1-12	1030	ne												
10-11	1100	ne												
5-6	1170	ne												
9-11	1190	ne												
2-5	1210	ne												
1-9	1290	ne												
8-12	1300	ne												

Utrošak kablova u ovom slučaju je jednak slučaju pod a) jer se koristi isti put obilaska.

c) Naposlijetku, riješimo zadatak primjenom kvadratnog Primovog algoritma. Na početku uzimamo x_1 kao referentni čvor:

Ref. čvor	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11	c12
	0											
x1				**670/x1**					1290/x1			1030/x1
x4			990/x4					**570/x4**	1290/x1			1030/x1
x8			990/x4		**410/x8**				770/x8			1030/x1
x5		1210/x5	990/x4			1170/x5			**770/x8**			850/x5
x9		660/x9	1020/x9			870/x9					1190/x9	**450/x9**
x12		**660/x9**	1020/x9			870/x9					1190/x9	
x2			1020/x9			**270/x2**				450/x2	750/x2	
x6			790/x6				970/x6			**450/x2**	750/x2	
x10			790/x6				970/x6				**750/x2**	
x11			790/x6				**300/x11**					
x7			**790/x6**									

Dobijamo isto minimalno povezujuće stablo kao i kod Kruskalovog algoritma. Međutim to ne mora biti slučaj jer MPS ne mora biti jedinstveno kada su težine barem dvije grane jednake. Zbog redoslijeda uzimanja smo dobili jednako stablo. Ne bismo napravili grešku ni da smo te dvije grane uzeli suprotnim redoslijedom obzirom da je suma težina i dalje ista kao u a) i b).

3. Turistička agencija “Pljačkaš tours” ima poslovnice u 8 gradova: Zavkul, Quwuti, Amcazo, Sosyab, Uyradu, Joxocap, Topufun i Rekazga. U sljedećoj tablici su date cijene direktnih avionskih letova između pojedinih gradova izražene u škafiškafnjacima (crtica znači da direktan let ne postoji):

	Zavkul	Quwuti	Amcazo	Sosyab	Uyradu	Joxocap	Topufun	Rekazga
Zavkul	0	-	470	370	340	1270	360	430
Quwuti	-	0	370	300	700	430	-	260
Amcazo	470	370	0	400	960	-	1100	230
Sosyab	370	300	400	0	-	1120	420	440
Uyradu	340	700	960	-	0	230	230	680
Joxocap	1270	430	-	1120	230	0	690	340
Topufun	360	-	1100	420	230	690	0	300
Rekazga	430	260	230	440	680	340	300	0

Međutim, poznato je da direktni letovi nisu uvijek i najjeftiniji način aviotransporta između gradova, nego je nekada povoljnije koristiti presjedanje (pogotovo ako se na taj način mogu koristiti usluge low-cost kompanija. Na primjer, iz Topufuna jeftinije je u Joxocap putovati sa presjedanjem u Uyraduu nego direktnim letom (sa presjedanjem plaćamo $230 + 230 = 460$ škafiškafnjaka, dok direktan let košta 690 škafiškafnjaka). Zbog toga, turistička agencija želi da sastavi tablicu koja sadrži informacije koliko iznose najjeftinije cijene aviotransporta između svakog para gradova u kojima agencija ima poslovnice (uz dopuštanje presjedanja) kao i u kojim gradovima treba eventualno vršiti presjedanja za svaki od tih transporta. Pomozite agenciji “Pljačkaš tours” da sastavi ove tablice. Postupak obavite uz pomoć Dijkstrinog algoritma, ali bez crtanja grafova, nego samo vršeći manipulacije nad zadanom tablicom, uz eventualno bilježenje pomoćnih dopunskih informacija.

Prije primjene Dijkstrinog algoritma označimo gradove pogodnim slovima:
A-Zavkul, B-Quwuti, C-Amcazo, D-Sosyab, E-Uyradu, F-Joxocap, G-Topofun, H-Rekazga

A-Zavkul

	A	B	C	D	E	F	G	H
A(0)	0	-	470/A	370/A	**340/A**	1270/A	360/A	430/A
E(340)		1040/E	470/A	370/A		570/E	**360/A**	430/A
G(360)		1040/E	470/A	**370/A**		570/E		430/A
D(370)		670/D	470/A			570/E		**430/A**
H(430)		670/D	**470/A**			570/E		
C(470)		670/D				**570/E**		
F(570)		**670/D**						

B-Quwuti

	A	B	C	D	E	F	G	H
B(0)	-	0	370/B	300/B	700/B	430/B	-	**260/B**
H(260)	690/H		370/B	**300/B**	700/B	430/B	560/H	
D(300)	670/D		**370/B**		700/B	430/B	560/H	
C(370)	670/D				700/B	**430/B**	560/H	
F(430)	670/D				660/F		**560/H**	
G(560)	670/D				**660/F**			
E(660)	**670/D**							

C-Amcazo

	A	B	C	D	E	F	G	H
C(0)	470/C	370/C	0	400/C	960/C	-	1100/C	**230/C**
H(230)	470/C	**370/C**		400/C	910/H	570/H	530/H	
B(370)	470/C			**400/C**	910/H	570/H	530/H	
D(400)	**470/C**				910/H	570/H	530/H	
A(470)					810/A	570/H	**530/H**	
G(530)					760/G	**570/H**		
F(570)					**760/G**			

D-Sosyab

	A	B	C	D	E	F	G	H
D(0)	370/D	**300/D**	400/D	0	-	1120/D	420/D	440/D
B(300)	**370/D**		400/D		1000/B	730/B	420/D	440/D
A(370)			**400/D**		710/A	730/B	420/D	440/D
C(400)					710/A	730/B	**420/D**	440/D
G(420)					650/G	730/B		**440/D**
H(440)					650/G	**730/B**		
F(730)					**650/G**			

E-Uyradu

	A	B	C	D	E	F	G	H
E(0)	340/E	700/E	960/E	-	0	**230/E**	230/E	680/E
F(230)	340/E	660/F	960/E	1350/F			**230/E**	570/F
G(230)	**340/E**	660/E	960/E	650/G				530/G
A(340)		660/E	810/A	650/G				**530/G**
H(530)		660/E	760/H	**650/G**				
D(650)		**660/E**	760/H					
B(660)			**760/H**					

F-Joxocap

	A	B	C	D	E	F	G	H
F(0)	1270/F	430/F	-	1120/F	**230/F**	0	690/F	340/F
E(230)	570/E	430/F	1190/E	1120/F			460/E	**340/F**
H(340)	570/E	**430/F**	570/H	780/H			460/E	
B(430)	570/E		570/H	730/B			**460/E**	
G(460)	**570/E**		570/H	730/B				
A(570)			**570/H**	730/B				
C(570)				**730/B**				

G-Topofun

	A	B	C	D	E	F	G	H
G(0)	360/G	-	1100/G	420/G	**230/G**	690/G	0	300/G
E(230)	360/G	930/E	1100/G	420/G		460/E		**300/G**
H(300)	**360/G**	560/H	530/H	420/G		460/E		
A(360)		560/H	530/H	**420/G**		460/E		
D(420)		560/H	530/H			**460/E**		
F(460)		560/H	**530/H**					
C(530)		**560/H**						

H-Rekazga

	A	B	C	D	E	F	G	H
H(0)	430/H	260/H	**230/H**	440/H	680/H	340/H	300/H	0
C(230)	430/H	**260/H**		440/H	680/H	340/H	300/H	
B(260)	430/H			440/H	680/H	340/H	**300/H**	
G(300)	430/H			440/H	530/G	**340/H**		
F(340)	**430/H**			440/H	530/G			
A(430)				**440/G**	530/G			
D(440)					**530/G**			

Tablica najkraćih puteva između pojedinih gradova:

	A-Zavkul	B-Quwuti	C-Amcazo	D-Sosyab	E-Uyradu	F-Joxocap	G-Topufun	H-Rekazga
A-Zavkul	0	670/D	470/A	370/A	340/A	570/E	360/A	430/A
B-Quwuti	670/D	0	370/B	300/B	660/B	430/B	560/B	260/B
C-Amcazo	470/C	370/C	0	400/C	760/G	570/H	530/H	230/C
D-Sosyab	370/D	300/D	400/D	0	650/G	730/B	420/D	440/D
E-Uyradu	340/E	660/E	760/H	650/G	0	230/E	230/E	530/G
F-Joxocap	570/E	430/F	570/H	730/B	230/F	0	460/E	340/F
G-Topufun	360/G	560/H	530/H	420/G	230/G	460/E	0	300/G
H-Rekazga	430/H	260/H	230/H	440/G	530/G	340/H	300/H	0

4. Dat je usmjereni težinski graf G = {{A, B, C, D, E, F, G, H, I, J}, {(A, F, 64), (A, I, -52), (B, E, 16), (C, A, 44), (C, B, 60), (C, F, 52), (D, B, 52), (E, G, 48), (E, J, 36), (F, D, -84), (F, E, 48), (G, C, -24), (G, D, 32), (G, I, 40), (H, C, -68), (H, F, 20), (I, E, 56), (J, G, -76)}} Koristeći Bellman-Fordov algoritam, dokažite da u ovom grafu postoji kontura sa negativnom sumom težina u konturi.
Nakon toga pronađite makar jednu takvu konturu. Postupak obavite "naslijepo", bez crtanja grafa, koristeći samo raspoložive informacije (eventualno uz bilježenje raznih pomoćnih informacija).

Za dokazivanje da u grafu postoji barem jedna kontura sa negativnom sumom težina moramo dobiti da je λ_A negativno u nekoj od iteracija algoritma:

x_j	A	B	C	D	E	F	G	H	I	J	λ_i
x_i	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	
A						64			-52		∞
B					∞						∞
C	∞	∞				∞					∞
D		∞									∞,-20
E							∞			∞	∞,112,4
F				-20	112						∞,64
G			∞	∞					∞		∞
H			∞			∞					∞
I					4						∞,-52
J							∞				∞

x_j	A	B	C	D	E	F	G	H	I	J	λ_i
x_i	0	∞	∞	**-20**	**4**	**64**	∞	∞	**-52**	∞	
A											∞
B											∞,32
C											∞,28
D		32									-20
E							52			40	4
F				-20	112						64
G			28	84					92		∞,52,-36
H											∞
I					4						-52
J							-36				∞,40

x_j	A	B	C	D	E	F	G	H	I	J	λ_i
x_i	0	32	28	**-20**	**4**	**64**	**-36**	besk	**-52**	40	
A											
B					48						32
C	72	88				80					28,-60
D											
E											
F											
G			**-60**	-4					4		-36
H											
I											
J							-36				40

x_j	A	B	C	D	E	F	G	H	I	J	λ_i
x_i	0	32	-60	**-20**	**4**	**64**	**-36**	besk	**-52**	40	
A											
B											
C	**-16**										-60
D											
E											
F											
G											
H											
I											
J											

Pošto smo u četvrtoj iteraciji dobili da je λ_A negativno, time je dokazano da postoji barem jedna kontura sa negativnom sumom težina. Jedna takva kontura je ***A-I-E-J-G-C-A*** *čija je suma:*

-52 + 56 + 36 - 76 - 24 + 44 = -16.

5. Dat je usmjereni težinski graf

   G= {{x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11}, {(x2, x4, 29), (x2, x8, 11), (x3, x7, 24), (x3, x10, 5), (x4, x5, 6), (x4, x6, 24), (x5, x1, 7), (x5, x6, 15), (x6, x1, 6), (x6, x11, 13), (x7, x2, 16), (x7, x4, 22), (x8, x6, 38), (x8, x11, 17), (x9, x3, 21), (x9, x7, 30), (x9, x10, 5), (x10, x2, 5), (x10, x8, 34), (x11, x1, 11)}}
   -Pokažite da u ovom grafu ima tačno jedan izvor (čvor ulaznog stepena 0) i tačno jedan ponor (čvor izlaznog stepena 0), te da se radi o acikličkom grafu;
   -Izvršite topološko sortiranje čvorova ovog grafa obavljajući DFS pretragu počev od izvora grafa;
   -Primjenom Dijkstrinog algoritma, pronađite najkraći put od izvora do ponora grafa i navedite koliko iznosi dužina tog puta;
   -Primjenom Bellman-Fordovog algoritma, pronađite najkraći put od izvora do ponora grafa i navedite koliko iznosi dužina tog puta;
   -Primjenom Bellman-Fordovog algoritma, pronađite najduži put od izvora do ponora grafa i navedite koliko iznosi dužina tog puta.
   Postupak provedite "naslijepo", bez crtanja grafa, koristeći samo raspoložive informacije

(eventualno uz bilježenje raznih pomoćnih informacija).

Za lakše dokazivanje napišimo graf u formi matrične tabele:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
x_1											
x_2				29				11			
x_3							24			5	
x_4					6	24					
x_5	7					15					
x_6	6										13
x_7		16		22							
x_8						38					17
x_9			21				30			5	
x_{10}		5						34			
x_{11}	11										

Očigledno je (iz tabele) izvor čvor x_9, a ponor čvor x_1.

Sada izvršimo topološko sortiranje čvorova putem DFS pretrage. Potrebno je invertirati graf, odnosno zamijeniti kolone i redove, obaviti DFS pretragu od izvora i obrnuti dobijenu sekvencu:

Topološko sortiranje daje sekvence:

1.) $x_1 \to x_5 \to x_4 \to x_7 \to x_3 \to x_9$

2.) $x_1 \to x_5 \to x_4 \to x_2 \to x_{10}$

3.) $x_1 \to x_{11}$

*Sada kad opet invertujemo naš graf dobijamo topološki sortiran naš početni graf. Kako niti u jednom trenutku nismo naišli na prethodno posjećen čvor, samim time je graf **acikličan**.*

Primijenimo Dijkstrin algoritam za pronalazak najkraćeg puta:

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11
x9(0)			21/x9				30/x9		**0**	**5/x9**	
x10(5)		**10/x10**	21/x9				30/x9	39/x10			
x2(10)			**21/x9**	39/x2			30/x9	21/x2			
x3(21)				39/x2			30/x9	**21/x2**			
x8(21)				39/x2		59/x8	**30/x9**				38/x8
x7(30)				39/x2		59/x8					**38/x8**
x11(38)	49/x11			**39/x2**		59/x8					
x4(39)	49/x11				**45/x4**	59/x8					
x5(45)	**49/x11**					59/x8					
x1(49)						**59/x8**					

Najkraći put od izvora do ponora je $x_9 \to x_{10} \to x_2 \to x_8 \to x_{11} \to x_1$, dužina puta je 49.

Sada primijenimo Bellman-Fordov algoritam za traženje najkraćeg puta od izvora do ponora grafa:

Iteracija 1.

x_j	1	2	3	4	5	6	7	8	9	10	11	λ_i
x_i	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	
x_1												∞
x_2				∞				∞				∞,46,10
x_3							45			26		∞,21
x_4					∞	∞						∞,52
x_5	∞					∞						∞
x_6	∞										∞	∞
x_7		46		**52**								∞,30
x_8						∞					∞	∞,39
x_9			**21**				**30**			**5**		0
x_{10}		**10**						**39**				∞,5
x_{11}	∞											∞

Iteracija 2.

x_j	1	2	3	4	5	6	7	8	9	10	11	λ_i
x_i	∞	**10**	**21**	52	∞	∞	**30**	39	0	**5**	∞	
x_1												52,49
x_2				**39**				**21**				10
x_3							45			26		21
x_4					**45**	63						52,39
x_5	52					60						45
x_6	66										73	63,60,59
x_7		46		52								30
x_8						**59**					**38**	39,21
x_9												
x_{10}		10						39				5
x_{11}	**49**											73,38

Iteracija 3.

x_j	1	2	3	4	5	6	7	8	9	10	11	λ_i
x_i	**49**	**10**	**21**	**39**	**45**	**59**	**30**	**21**	**0**	**5**	**38**	
x_1												49
x_2												
x_3												
x_4					45	63						39
x_5	52					60						45
x_6	65										72	59
x_7												
x_8						59					38	21
x_9												
x_{10}												
x_{11}	49											38

Kako nema promjena u potencijalima tu naš algoritam terminira te najkraći put od izvora do ponora je $x_9 \to x_{10} \to x_2 \to x_8 \to x_{11} \to x_1$, dužina puta je 49.

Naposlijetku nađimo i najduži put pomoću Bellman-Fordovog algoritma:

Xj	1	2	3	4	5	6	7	8	9	10	11	λ_i
Xi	∞	∞	∞	∞	∞	∞	∞	∞	0	∞	∞	
x1												∞
x2				∞				∞				∞,-10
x3							∞			∞		∞,-21
x4					∞	∞						∞
x5	∞					∞						∞
x6	∞										∞	∞
x7		∞		∞								∞,-30
x8						∞					∞	∞,-39
x9			-21				-30			-5		0
x10		-10						-39				∞,-5
x11	∞											∞

Druga iteracija:

Xj	1	2	3	4	5	6	7	8	9	10	11	λ_i
Xi	∞	-10	**-21**	∞	∞	∞	-30	-39	0	-5	∞	
x1												∞,-52,-69,-87
x2				-39				-21				-10,-61
x3							**-45**			**-26**		-21
x4					**-45**	-63						∞,-39,-67
x5	-52					-60						∞,-45
x6	-69										**-76**	∞,-63,-77
x7		**-61**		**-67**								-30,-45
x8						**-77**					-56	-39,-60
x9												0
x10		-31						**-60**				-5,-26
x11	**-87**											∞,-76

Treća iteracija:

Xj	1	2	3	4	5	6	7	8	9	10	11	λ_i
Xi	-87	**-61**	**-21**	-67	-45	-77	**-45**	-60	0	**-26**	-76	
x1												-87,-103,-120,-138
x2				**-90**				**-72**				-61
x3							-45			-26		-21
x4					**-96**	**-114**						-67,-90
x5	-103					-111						-45,-96
x6	-120										**-127**	-77,-114
x7		-61		-67								-45
x8						-110					-89	-60,-72
x9												0
x10		-31						-60				-26
x11	**-138**											-76,-127

Četvrta iteracija, ujedno i zadnja jer nema promjena u tabeli

Xj	1	2	3	4	5	6	7	8	9	10	11	λ_i
Xi	-138	**-61**	**-21**	-90	-96	-114	**-45**	-72	0	**-26**	-127	
x1												-138
x2												
x3												
x4					-96	-114						-90
x5	-103					-111						-96
x6	-120										-127	-114
x7												
x8						-110					-89	-72
x9												
x10												
x11	-138											-127

Najduži put je $x_9 \rightarrow x_3 \rightarrow x_7 \rightarrow x_2 \rightarrow x_4 \rightarrow x_6 \rightarrow x_{11} \rightarrow x_1$

dužina je 21+24+16+29+24+13+11 = 138.

6. Klijentski računar K vrši istovremeni download neke datoteke sa 3 servera S1, S2 i S3. Put od tih servera do klijentskog računara vrši se putem mreže posrednika (rutera) R1 – R8. Ispod je naveden popis raspoloživih komunikacionih kanala, pri čemu trojka oblika (X, Y, b) označava da je moguća komunikacija od čvorišta X do čvorišta Y maksimalnom brzinom od b Mbita/s:

   (S1, R2, 40) (S2, R2, 110) (S2, R7, 150) (S3, R4, 140) (S3, R7, 150) (R1, K, 30) (R2, R6, 120) (R3, K, 90) (R4, R7, 110) (R4, R8, 30) (R5, K, 80) (R6, R1, 110) (R6, R3, 30) (R7, R1, 30) (R7, R5, 120) (R8, R1, 130) (R8, R5, 120)

   Primjenom Ford-Fulkersonovog algoritma odredite maksimalnu brzinu kojom klijent može izvršiti download posmatrane datoteke kao i kolika će pri tome biti aktuelna brzina prenosa podataka kroz svaki od navedenih raspoloživih komunikacionih kanala. Postupak obavite "naslijepo", bez crtanja grafa, vršeći samo manipulacije sa matricom kapaciteta grana.

Obzirom da imamo tri izvora, napravimo jedan superizvor (SI) i povežimo ga sa ta tri izvora beskonačnim kapacitetima. Imamo:

	SI	S1	S2	S3	R1	R2	R3	R4	R5	R6	R7	R8	K	
SI	0	∞	∞	∞	0	0	0	0	0	0	0	0	0	-/0
S1	0	0	0	0	0	40	0	0	0	0	0	0	0	SI/1
S2	0	0	0	0	0	110	0	0	0	0	**150**	0	0	SI/1
S3	0	0	0	0	0	0	0	140	0	0	150	0	0	SI/1
R1	0	0	0	0	0	0	0	0	0	0	0	0	30	R7/3
R2	0	0	0	0	0	0	0	0	0	120	0	0	0	S1/2 - S2/2
R3	0	0	0	0	0	0	0	0	0	0	0	0	90	R6/4
R4	0	0	0	0	0	0	0	0	0	0	110	30	0	S3/2
R5	0	0	0	0	0	0	0	0	0	0	0	0	**80**	R7/3
R6	0	0	0	0	110	0	30	0	0	0	0	0	0	R2/3
R7	0	0	0	0	30	0	0	0	**120**	0	0	0	0	S2/2 - S3/2
R8	0	0	0	0	130	0	0	0	120	0	0	0	0	R4/3
K	0	0	0	0	0	0	0	0	0	0	0	0	0	R5/4

Povećavajući lanac 1: $S_2 \rightarrow R_7 \rightarrow R_5 \rightarrow K$

$\Delta_{max} = \min\{150, 80, 120\} = 80;$

	SI	S1	S2	S3	R1	R2	R3	R4	R5	R6	R7	R8	K	
SI	0	∞	∞	∞	0	0	0	0	0	0	0	0	0	-/0
S1	0	0	0	0	0	40	0	0	0	0	0	0	0	SI/1
S2	0	0	0	0	0	110	0	0	0	0	70	0	0	SI/1
S3	0	0	0	0	0	0	0	140	0	0	**150**	0	0	SI/1
R1	0	0	0	0	0	0	0	0	0	0	0	0	**30**	R7/3
R2	0	0	0	0	0	0	0	0	0	120	0	0	0	S1/2 - S2/2
R3	0	0	0	0	0	0	0	0	0	0	0	0	90	R6/4
R4	0	0	0	0	0	0	0	0	0	0	110	30	0	S3/2
R5	0	0	0	0	0	0	0	0	0	0	0	0	0	R7/3
R6	0	0	0	0	110	0	30	0	0	0	0	0	0	R2/3
R7	0	0	0	0	**30**	0	0	0	40	0	0	0	0	S2/2 - S3/2
R8	0	0	0	0	130	0	0	0	120	0	0	0	0	R4/3
K	0	0	0	0	0	0	0	0	80	0	0	0	0	R1/4

Povećavajući lanac 2: $S_3 \to R_7 \to R_1 \to K$

$\Delta_{max} = \min\{150, 30, 30\} = 30;$

	SI	S1	S2	S3	R1	R2	R3	R4	R5	R6	R7	R8	K	
SI	0	∞	∞	∞	0	0	0	0	0	0	0	0	0	-/0
S1	0	0	0	0	0	40	0	0	0	0	0	0	0	SI/1
S2	0	0	0	0	0	**110**	0	0	0	0	70	0	0	SI/1
S3	0	0	0	0	0	0	0	140	0	0	120	0	0	SI/1
R1	0	0	0	0	0	0	0	0	0	0	0	0	0	R6/4 - R8/4
R2	0	0	0	0	0	0	0	0	0	**120**	0	0	0	S1/2 - S2/2
R3	0	0	0	0	0	0	0	0	0	0	0	0	**90**	R6/4
R4	0	0	0	0	0	0	0	0	0	0	110	30	0	S3/2
R5	0	0	0	0	0	0	0	0	0	0	0	0	0	R7/3
R6	0	0	0	0	110	0	**30**	0	0	0	0	0	0	R2/3
R7	0	0	0	0	0	0	0	0	40	0	0	0	0	S2/2 - S3/2
R8	0	0	0	0	130	0	0	0	120	0	0	0	0	R4/3
K	0	0	0	0	30	0	0	0	80	0	0	0	0	R3/5

Povećavajući lanac 3: $S_2 \to R_2 \to R_6 \to R_3 \to K$

$\Delta_{max} = \min\{110, 120, 30, 90\} = 30;$

	SI	S1	S2	S3	R1	R2	R3	R4	R5	R6	R7	R8	K
SI	0	∞	∞	∞	0	0	**0**	0	0	0	0	0	0
S1	0	0	0	0	0	40	**0**	0	0	0	0	0	0
S2	0	0	0	0	0	80	**0**	0	0	0	70	0	0
S3	0	0	0	0	0	0	**0**	140	0	0	120	0	0
R1	0	0	0	0	0	0	**0**	0	0	0	0	0	0
R2	0	0	0	0	0	0	**0**	0	0	90	0	0	0
R3	0	0	0	0	0	0	**0**	0	0	0	0	0	**60**
R4	0	0	0	0	0	0	**0**	0	0	0	110	30	0
R5	0	0	0	0	0	0	**0**	0	0	0	0	0	0
R6	0	0	0	0	110	0	**0**	0	0	0	0	0	0
R7	0	0	0	0	0	0	**0**	0	40	0	0	0	0
R8	0	0	0	0	130	0	**0**	0	120	0	0	0	0
K	0	0	0	0	30	0	30	0	80	0	0	0	0

Algoritam terminira jer smo iscrpili sve povezujuće grane, što se može vidjeti iz posljednje tabele jer jedini čvor koji ima veze sa ponorom je R3 ali nemamo više nijednu vezu sa R3 jer su sve iskorištene

Sada nađimo maximalni protok

Ukupan maksimalni protok je 80 + 30 + 30 = $\boxed{\textit{140 Mbit/s}}$

7. Na nekoj zabavi, šest atraktivnih djevojaka D1 – D6 upoznalo je pet isto tako atraktivnih momaka M1 – M5. Tom prilikom rodile su se i uzajamne simpatije. Pri tome, djevojke su se pokazale pomalo neodlučne, jer se svakoj od njih sviđa više momaka. U sljedećoj tablici prikazano je kojoj djevojci se sviđaju koji momci:

Djevojke:	Momci:
D1	M1, M5
D2	M1,M3,M5
D3	M2,M4,M5
D4	M3,M5
D5	M1,M3,M5
D6	M1,M3,M4,M5

Što se tiče momaka, oni su se pokazali kao potpuno indiferentni, to jest svakom od njih se sviđa svaka od djevojaka (što se ovdje pokazalo kao dobro, jer se u ovom zadatku oni svakako ništa ne pitaju). Pronađite maksimalan broj parova koji se može formirati tako da ni jedna djevojka ne bude u vezi sa nekim od momaka koji joj se ne sviđa (s obzirom da ima “višak” djevojaka u odnosu na momke, jedna od djevojaka će nažalost morati “izvisiti”). Podrazumijeva se da jedna djevojka može biti u vezi samo sa jednim momkom i obrnuto (tj. poligamija i poliandrija su isključeni). Problem riješite svođenjem ovog problema na problem maksimalnog protoka i primjenom Ford-Fulkersonovog algoritma (bez obzira što se, zbog male dimenzionalnosti, ovaj problem može lako riješiti intuitivno).

Prvi način (Ford-Fulkerson). Označimo izvor sa I a ponor sa P, imamo sljedeće tabele:

	I	D1	D2	D3	D4	D5	D6	M1	M2	M3	M4	M5	P	
I		1	1	1	1	1	1							-/0
D1								1				1		I/1
D2								1		1		1		I/1
D3									1		1	1		I/1
D4										1		1		I/1
D5								1		1		1		I/1
D6								1		1	1	1		I/1
M1													1	D1/2
M2													1	D3/2
M3													1	D2/2
M4													1	D3/2
M5													1	D1/2
P														M1/3

	I	D1	D2	D3	D4	D5	D6	M1	M2	M3	M4	M5	P	
I			1	1	1	1	1							-/0
D1	1											1		M1/3
D2								1		1		1		I/1
D3									**1**		1	1		I/1
D4										1		1		I/1
D5								1		1		1		I/1
D6								1		1	1	1		I/1
M1		1												D2/2
M2													**1**	D3/2
M3													1	D2/2
M4													1	D3/2
M5													1	D2/2
P								1						M2/3

	I	D1	D2	D3	D4	D5	D6	M1	M2	M3	M4	M5	P	
I			1		1	1	1							-/0
D1	1											1		M1/3
D2								1		**1**		1		I/1
D3	1										1	1		I/1
D4										1		1		I/1
D5								1		1		1		I/1
D6								1		1	1	1		I/1
M1		1												D2/2
M2				1										
M3													**1**	D2/2
M4													1	D6/2
M5													1	D2/2
P								1	1					M3/3

	I	D1	D2	D3	D4	D5	D6	M1	M2	M3	M4	M5	P	
I					1	1	**1**							-/0
D1	1											1		
D2	1							1				1		
D3	1										1	1		
D4										1		1		I/1
D5								1		1		1		I/1
D6								1		1	**1**	1		I/1
M1		1												
M2				1										
M3			1											D4/2
M4													**1**	D6/2
M5													1	D4/2
P								1	1	1				M4/3

	I	D1	D2	D3	D4	D5	D6	M1	M2	M3	M4	M5	P	
I					**1**	1								-/0
D1	1											1		M1/3
D2	1							1				1		M3/3
D3	1										1	1		I/1
D4										1		**1**		I/1
D5								1		1		1		I/1
D6	1							1		1		1		I/1
M1		1												D4/2
M2				1										
M3			1											D4/2
M4							1							
M5													**1**	D4/2
P								1	1	1	1			M4/3

	I	D1	D2	D3	D4	D5	D6	M1	M2	M3	M4	M5	P	
I						1								-/0
D1	1											1		M1/3
D2	1							1				1		M3/3
D3	1										1	1		I/1
D4	1									1				I/1
D5								1		1		1		I/1
D6	1							1		1		1		I/1
M1		1												
M2				1										
M3			1											
M4							1							
M5					1									
P								1	1	1	1	1		

Vidimo da je dosegnut maximalan protok koji iznosi 5, odnosno dosegli smo uparivanje od 5 osoba i to :
D1-M1
D2-M3
D3-M2
D4-M5
D6-M4
Izvisila je djevojka D5.

Drugi način: Maksimalno uparivanje možemo odrediti i intuitivno, označimo uparivanja bojama i to (pozitivno uparivanje) zelenom, a nemoguća crvenom bojom:

Iteracija 1:

	M1	M2	M3	M4	M5
D1	1				1
D2	1		1		1
D3		1		1	1
D4			1		1
D5	1		1		1
D6	1		1	1	1

Dobili smo uparivanje D_3 - M_2.

Iteracija 2:

	M1	M2	M3	M4	M5
D1	1				1
D2	1		1		1
D3		1		1	1
D4			1		1
D5	1		1		1
D6	1		1	1	1

Dobili smo uparivanje D_6 - M_4.

Iteracija 3:

	M1	M2	M3	M4	M5
D1	1				1
D2	1		1		1
D3		1		1	1
D4			1		1
D5	1		1		1
D6	1		1	1	1

Dobili smo uparivanje D_1 - M_5.

Iteracija 4:

	M1	M2	M3	M4	M5
D1	1				1
D2	1		1		1
D3		1		1	1
D4			1		1
D5	1		1		1
D6	1		1	1	1

Dobili smo uparivanje D_5 - M_3.

Iteracija 5:

	M1	M2	M3	M4	M5
D1	1				1
D2	1		1		1
D3		1		1	1
D4			1		1
D5	1		1		1
D6	1		1	1	1

Dobili smo uparivanje D_2 - M_1.

Primijetimo da je "izvisila" djevojka D_4 iako ovo naravno nije jedino moguće uparivanje djevojaka kao što znamo iz postavke. Dobijeni rezultat odgovara činjenici da nije dozvoljena poligamija ni poliandrija pa je naše uparivanje:

D_1-M_5
D_2-M_1
D_3-M_2
D_5-M_3
D_6-M_4

Djevojka D_4 je slobodna.