Tema 1

Mocanu Ada-Astrid
&
Ciripescu Teodor

Grupa A7

8 Noiembrie 2019

Problema 1

a) Ne dorim sa aratam ca $\forall u \in V(G)$ si $\forall v \in V(G), \exists$ un drum orientat de la u la v daca si numai daca $\forall$ ar fi $e \in E(G)$, $G \setminus \{e\}$ ramane conex.

($\Rightarrow$) Pp. ca $\exists e \in E(G)$ a.i. $G \setminus \{e\}$ nu e conex.

Atunci, $\exists u, v \in V(G)$ a.i. in $G \setminus \{e\}$ nu $\exists$ drum de la u la v si de la v la u.

Dar din ipoteza, $\forall u, v \in V(G), \exists$ un drum orientat de la u la v $\Rightarrow$ Contradictie.

Asadar, $\forall u, v \in V(G)$ a.i. $\exists$ un drum orientat de la u la v IMPLICA $\forall e \in E(G)$, $G \setminus \{e\}$ ramane conex.

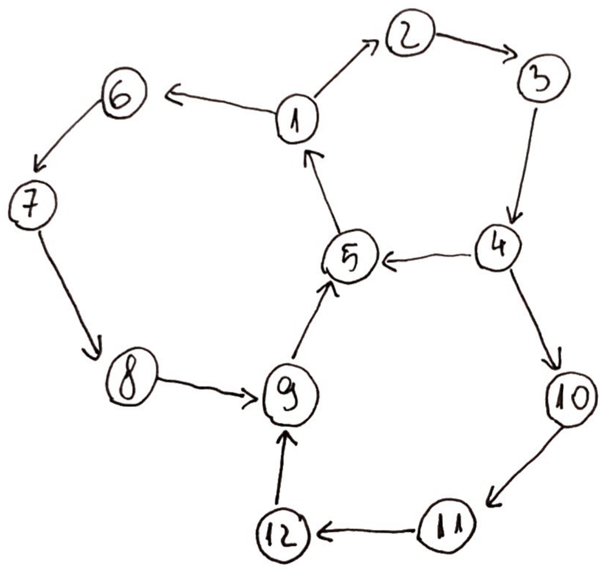
($\Leftarrow$) (este implicata de corectitudinea algoritmului de la subpunctul (b))

b)

```
procedure COMPUTE
    nr_circuit ← 1
    for each nod_curent ∈ V(G) do
        if nu am parcurs toate nodurile then
            parcurgere(nod_curent, nr_circuit)    ▷ incepem ciclul nr k din nodul u(daca ∃ drumul)
            nr_circuit ← nr_circuit + 1           ▷ trecem la ciclul/circuitul urmator
procedure PARCURGERE(nod_curent, nr_ciclu)
    vizitare[nod_curent] ← nr_circuit
    for each nod ∈ V(G) do
        if muchia nod_curent − nod are alocat un sens then
            pastram sensul
            continuam parcurgerea din nod
        else
            if ∃ muchia nod_curent − nod then
                ii dam sens de parcurs de la nod_curent spre nod
                continuam parcurgere din nod
    ne oprim cand ajungem din nodul din care am pornit
```

Grafurile care respecta proprietatile cerute sunt formate din mai multe cicluri alaturate, intre care pot exista coarde. (analogie cu ”floricelele”)
Prin parcurgerea fiecarui ciclu, dand sens fiecarei muchii si formand un circuit, vom ajunge sa vizitam fiecare nod pana la incheierea programului.
(Astfel incat pentru fiecare ciclu ”vecin” cu un circuit deja format, vom incepe acel nou circuit urmand sensul circuitului sau ”vecin”, unde vecin se

refera la faptul ca au muchii comune).



Problema 2

a) $(\Rightarrow) X \subseteq V$ este uv - separatoare minimala si demonstram ca orice nod din X are vecini in ambele componenete.

Pp. Reducere la Absurd
$X \subseteq V$ este uv - separatoare minimala a.i. $\exists x \in V(X)$, care nu are vecini in ambele componente.
$\Rightarrow$
Caz 1. x nu are vecini in niciuna din componenete $\Rightarrow u$ poate fi eliminat din multimea $X \Rightarrow X \setminus \{x\}$ este uv - separatoare si are cardinal $< X \Rightarrow X$ nu e minimal.
Caz 2. x are vecini intr-o singura componenta $C_1 \Rightarrow$ putem muta x din X a.i. $x \in C_1 \Rightarrow \exists Y = X \setminus \{x\}$ care este uv - separatoare si are cardinal $< X \Rightarrow X$ nu este minimal.

$\Rightarrow$Ambele False $\Rightarrow \forall x \in X$, x are vecini in ambele componente rezultate prin eliminarea tuturor nodurilor din X.

$(\Leftarrow)$ Daca $\forall$ nod din X are vecini in ambele componente, atunci $X \subseteq V$ nu este uv - separatoare minimala (adica $X \subseteq V$ este doar uv - separatoare).

$\Rightarrow \exists Y \subseteq X$ submultime a lui X.
$\Rightarrow \exists x \in X$, $x \notin Y$ ai $xu \in E(G), xv \in E(G)$ unde $u \in C_1, v \in C_2$

$\Rightarrow$ cum $x \notin Y \Rightarrow x$ poate apartine uneia dintre cele 2 componente, spre exemplu, $x \in C_1$.
Dar asta inseamna ca prin eliminarea lui Y nu putem obtine 2 componente conexe (pentru ca xv actioneaza ca punte) $\Rightarrow$ Contradictie.
$\Rightarrow$ Daca $\forall$ nod din X are vecini in ambele componente, atunci X este uv - separatoare minimala.

b) -

Problema 3

a) Vom alege o implementare cu matrice de adiacenta, deoarece observam ca implementarea cu liste simplu inlantuite genereaza o complexitate egala cu $O(n^2) + O(n * m)$. Pe de alta parte, utilizand o matrice de adiacenta, obtinem o complexitate de $O(n^2)$. De mentionat ca utilizam un vector d, pentru a memorarea gradelor nodurilor. Crearea matricei are complexitate $O(n^2)$. Parcurgerea nodurilor se face in $O(n)$, iar verificarea gradului nodului in $O(1)$. Eliminarea nodului din graf se face in $O(1)$, insa este necesara si refacerea constanta a vectorului de grade, in $O(n)$, fapt ce este repetat pentru fiecare nod $\Rightarrow O(n) * O(n) = O(n^2)$. Asadar, complexitatea timp a algoritmului este $O(n^2)$.

$G' \leftarrow G$ $\qquad \triangleright\ O(n^2)$

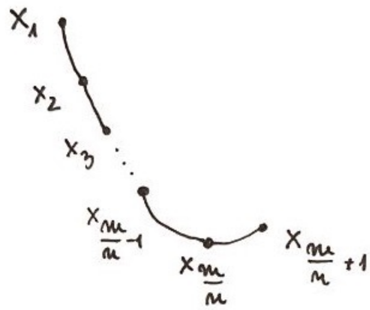
while $\exists u \in V(G')$ a.i. $d_{G'}(u) < m/n$ **do** $\qquad \triangleright\ O(n) + O(1)$(parcurgere + interogare vector grade)

  $G' \leftarrow G' - u$ $\qquad \triangleright\ O(1) + O(n)$(eliminare + refacere vector grade)

return G'

b) Dorim sa observam evolutia raportului muchii-noduri in graful G'.
Pp ca la Pasul nr. 1 eliminam un nod si implicit k muchii. Astfel, vom avem $n_1 = n - 1$ si $m_1 = m - k$.
Comparand raportul m/n cu m_1/n_1, obtinem ca acesta este echivalent cu compararea lui $m * n - m$ cu $m * n - k * n$
$\Rightarrow$ avem de comparat $k*n$ cu m. Dar $k < m/n$ deoarece k este gradul nodului pe care il eliminam $\Rightarrow m/n < m_1/n_1$
Aplicam INDUCTIE
Presupunem ca $m_{p-1}/n_{p-1} < m_p/n_p$.
Fie $m_{p+1} = m_p - y$, unde y este gradul unui nod eliminat si $n_{p+1} = n_p - 1$.
Vom compara m_p/n_p cu m_{p+1}/n_{p+1}. Fapt care se reduce la compararea lui $y * n_p$ cu $m_p \Rightarrow y$ trebuie comparat cu m_p/n_p
Cum $m_p/n_p > m/n$, dar $m/n > y \Rightarrow y < m_p/n_p \Rightarrow m_p/n_p < m_{p+1}/n_{p+1}$.
Fie k pasul final din executarea algoritmului.
Pp ca graful rezultat are 0 muchii$\Rightarrow m_k = 0 \Rightarrow m_k/n_k = 0$. Dar $m_k/n_k > m/n \Rightarrow 0 > n/m$
$\Rightarrow$Contradictie $\Rightarrow$ Graful nu poate fi niciodata nul.

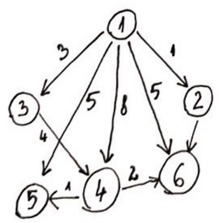
c) $\forall v, d(v) \geq \frac{m}{n}$
Din punctul b) $\Rightarrow \forall$ graf rezultat prin taierea tuturor nodurilor cu grad$< \frac{m}{n}$ este nenul.
Cu alte cuvinte, $\forall v, d(v) \geq \frac{m}{n} \Rightarrow \forall v \in V(G')$ are macar $\frac{m}{n}$ vecini $\Rightarrow \exists$ cel putin $\frac{m}{n}+1$ noduri in G'.
Un nod x_1 are $\frac{m}{n}$ vecini dintre care unul este x_2 (generand astfel un drum de lungime 1).
$x_2 \in V(G') \Rightarrow d(x_2) \geq \frac{m}{n} \Rightarrow x_2$ are si el $\frac{m}{n}$ vecini $\Rightarrow$ are macar $\frac{m}{n}-1$ vecini diferiti de x_1; unul dintre ei fiind x_3, iar x_3 are macar $\frac{m}{n}-2$ vecini, diferiti de x_1, x_2 (generand drum de lungime 3).

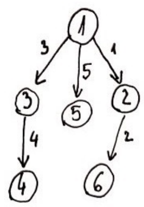


Astfel, ajungem la $x_{\frac{m}{n}-1}$ care are macar 2 vecini diferiti, de $x_1, x_2, ..., x_{\frac{m}{n}-2}$, unul fiind $x_{\frac{m}{n}}$ (gen. drum de lungime $\frac{m}{n}-1$).
Asadar, $x_{\frac{m}{n}}$ are macar 1 vecin diferit de $x_1, x_2, ..., x_{\frac{m}{n}-1}$, notat $x_{\frac{m}{n}+1}$ si el genereaza un drum de lungime $\frac{m}{n}$.
$\Rightarrow$ Graful contine un drum de lungime cel putin $\frac{m}{n}$.

Problema 4

a) $x_0 \in V$ din care toate celelalte noduri sunt accesibile $\Rightarrow G$ conex
Avem si o functie de cost $a : E \rightarrow \mathbb{R}+ \Rightarrow \nexists$ cicluri negative
$\Rightarrow$ putem aplica algoritmul lui Dijkstra, astfel generand un arbore cu drumuri minime(de la un varf la toate celelalte)





In exemplul alaturat am pornit din nodul 1

b) Fie a o matrice de dimensiune $n \times n$, care reprezinta matricea cost drumuri.

$$a[i][j] = \begin{cases} 0, & \text{daca i=j (nu exista cost de la un nod la el insusi)} \\ cost(arc), & \text{daca } \exists \text{ arc de la i la j} \\ \infty, & \text{altfel} \end{cases}$$

Fie x_0 nodul din care vom porni pentru aflarea drumului de la x_0 la oricare alt nod. $drum[i]$ este vectorul in care obtinem drumul de la x_0 la i de cost minim.

```
procedure COMPUTE
    for each u ∈ V(G) a.i. u ≠ x0 do
        drum[u] ← a[x0][u]
selectat[v] ← 1
k ← 1
    while k ≤ n − 1 do
        min ← ∞
        for each u ∈ V(G) do
            if drum[u] < min && (!selectat[u]) then
                min ← drum[u]
                pozmin ← u
        selectat[pozmin] ← 1
        for each u ∈ V(G) do
            if !selectat[u] then
                if drum[u] > drum[pozmin] + a[pozmin][u] then
                    drum[u] ← drum[pozmin] + a[pozmin][u]
            k ← k + 1
```

Tema 2

Mocanu Ada-Astrid
&
Ciripescu Teodor

Grupa A7

22 Noiembrie 2019

Problema 1

a) ADEVARAT

Pentru orice muchie specifica de cost minim,$\exists$ un APM care o contine, deoarece sortarea muchiilor poate fi adaptata asa fel incat sa fie aleasa muchia dorita cu prioritate, fara sa genereze ciclu (de exemplu muchia dorita sa fie prima muchie aleasa).

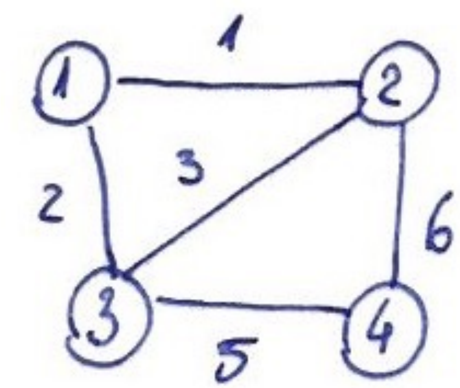
b)FALS

Contraexemplu:

Fie circuitul $C = \{(2, 3, 4)\}$

Muchia de cost minim $= (2, 3)$

Ea nu poate fi adaugata in APM deoarece $\exists$ muchiile $(1, 2)$ si $(1, 3)$, cu cost mai mic decat $(2, 3)$, care vor fi adaugate inainte in APM, deci adaugarea lui $(2, 3)$ ar genera circuit.



c)ADEVARAT

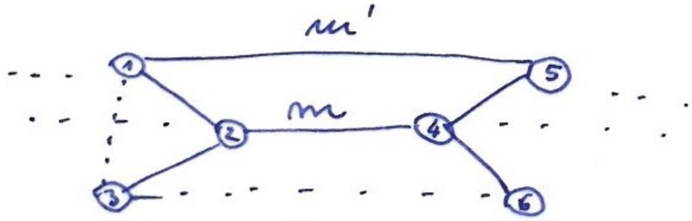
Fie T_G arborele partial de cost minim al lui G si m muchie din APM astfel incat $m \in$ unei anumite taieturi a lui G, M.

Cum $m \in$ taieturii $\Rightarrow$ eliminarea lui m deconecteaza APM-ul, obtinandu-se astfel, 2 componente conexe.

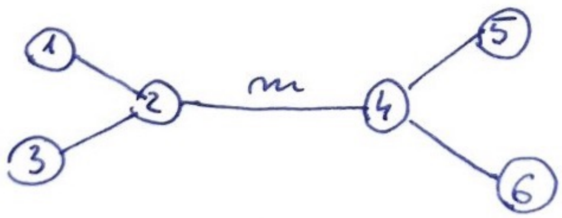
In G, eliminarea muchiilor din M genereaza aceleasi 2 componente conexe. Daca m nu ar fi muchia cu costul cel mai mic din taietura, atunci $\exists m' \in M$ astfel incat $c(m') < c(m) \Rightarrow$ la crearea APM-ului, am fi putut uni cele 2 componente conexe folosind m' in loc de m.

Dar, asta ar insemna ca am obtine un arbore partial cu cost $<$ APM$\Rightarrow$

pentru $\forall m$ din APM, $\exists$ o taietura in care m este muchie de cost minim.



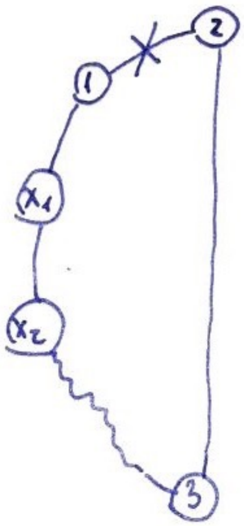
G



T_G

Problema 2

a)
$E(T^*_H) = E(T^*) \cap E(H)$
T_H* este conex (ipoteza) si $E(T^*_H) \in E(T^*) \Rightarrow T^*_H$ conex si aciclic $\Rightarrow T^*_H$ arbore.
Presupunem ca T^*_H nu ar fi arborele partial de cost minim al lui H.
$\Rightarrow \exists$ in H o muchie (x, y) cu cost mai mic decat drumul de la x la y in $T^* \cap H$.
Daca incercam adaugarea lui (x, y) la $T^*_H \Rightarrow$ trebuie scoasa o muchie din T^*_H si inlocuita cu $(x, y) \Rightarrow$ am crea un alt drum minim care sa uneasca x cu y si nodurile dintre ele.



(Adaugam muchia $(2, 3)$, cream circuit, scadem muchia $(2, 1) \Rightarrow$ ramane conex)
Cu alte cuvinte, cum muchia $(x, y) \in H \Rightarrow (x, y) \in G \Rightarrow$ exista in G o muchie cu cost $<$ decat un drum de la x la y care se afla in APM-ul grafului.$\Rightarrow$ contradictie $\Rightarrow$ presupunere falsa $\Rightarrow T^*_H$ este un arbore partial de cost c minim al lui $H \Rightarrow H$ este c-extensibil.
NOTA: toate nodurile din H apartin lui $H \cap T^*$

b) Contractarea tuturor muchiilor din H se reduce la existenta unei singure muchii reprezentanta a intregului subgraf H.
Muchii multiple se vor gasi intre $G - H$ si H contractat.
Fie T_{G_H} arborele partial de cost minim al lui $G_H \Rightarrow$.

Prin eliminarea muchiei care reprezinta contractarea grafului H vom obtine 2 componente conexe, ambele arbori partiali de cost minim.
Astfel, prin inlocuirea muchiei care reprezinta contractarea grafului H cu arborele partial de cost minim al lui H, vom obtine tot un arbore, fiind un graf conex si fara cicluri care contine muchiile cu costurile cele mai mici din T_{G_H} si muchiile cu costurile cele mai mici din $T_H^* \Rightarrow$ un APM

Problema 3

3.

$i \to ii$

In graful G vom elimina pe rand cate un nod din multimile S, respectiv T si vom urmari comportamentul acestora si al nodurilor ramase.

$G - \{x, y\}$ este cuplaj perfect $\forall x \in S, \forall y \in T \Rightarrow G - \{x, y\}$, unde $G - \{x, y\}$ bipartit, continand un cuplaj perfect, inseamna ca satureaza toate nodurile $\Rightarrow$ nu raman noduri izolate nici in multimea $S - \{x\}$, nici in multimea $T - \{y\} \Rightarrow |S - \{x\}| = |T - \{y\}| \Rightarrow$ readaugand x si y fiecare in multimea lui, $|S| = |T| =$numar noduri$/2 = n/2$ notat cu k.

Fie $S = \{x, x_1, x_2, ..., x_{k-1}\}$ si $T = \{y, y_1, y_2, ..., y_{k-1}\}$.

Prin eliminarea lui x si y ramanem cu un cuplaj perfect.

Prin notatie, alegem ca cuplajul sa se faca

prin existenta muchiilor (x_1, y_1),(x_2, y_2),...,(x_{n-1}, y_{n-1}).

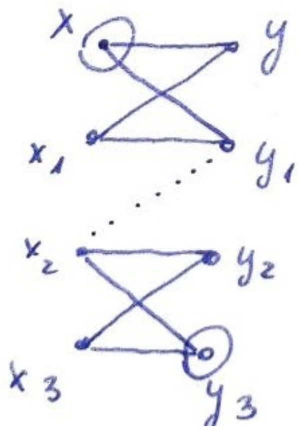
Pe de alta parte, prin eliminarea lui x si y_1 trebuie sa obtinem de asemenea un cuplaj perfect$\Rightarrow$ x_1 si y trebuie si ele saturate, tinand cont de faptul ca, cunoastem o modalitate de a satura celelalte noduri, din pasul precedent.

Ne dam seama ca oricum vom elimina perechea x si y_i unde $i = 1...k-1$, vom fi nevoiti sa il saturam pe $y \Rightarrow$ va exista macar un nod diferit de x pentru care exista o muchie care apartine unui cuplaj perfect.

Utilizand aceeasi metoda, vom obtine ca $\forall y_i$ este legat de 2 noduri distincte.

De asemenea $\forall x_i$ este legat de 2 noduri distincte.

Vom incerca sa cream cel mai mai nefavorabil caz (cel in care graful nu ar fi conex), de unde vom avea niste structuri, ca niste "clepsidre", care vor arata astfel:



Alaturand mai multe astfel de structuri, obtinem mai multe componente conexe in interiorul carora oricum am elimina 2 noduri $x \in S$ si $y \in T$, graful $G - \{x, y\}$ sa contina un cuplaj perfect, fara insa sa fie conex.
Insa, oricum am elimina x din S si y din T care sunt din componente "clepsidra" diferite, vom obtine macar 2 noduri care nu vor putea fi saturate, unul in S si unul in T.
$\Rightarrow$ pentru a putea mentine cuplaj perfect, oricum am elimina 2 noduri, x, y ca mai sus, trebuie sa existe macar o muchie care sa conecteze cele 2 componente conexe. Analog, se vor conecta astfel toate compontentele, graful G rezultand a fi conex, chiar si in cel mai nefavorabil caz.

Fie x si y, 2 noduri intre care exista muchie.
Stim ca $G - \{x, y\}$ are cuplaj perfect. $\Rightarrow$ daca am adauga inapoi muchia (x, y) in graful G, am avea un cuplaj perfect. Analog, eliminam capetele oricarei muchii din G si readaugand-o mereu vom obtine ca G are un cuplaj perfect.
$\Rightarrow \forall$ muchie din G apartine unui cuplaj perfect.

$ii \rightarrow iii$
G conex si $\forall e \in E(G) \in$ unui cuplaj perfect $\Rightarrow$
$|S| = |T|$ si $\emptyset \neq A \subsetneq S$ a.i. $|N_G(A)| > |A|$
$\forall e \in E(G) \in$ cuplaj perfect $\Rightarrow$ $(*)$ toate nodurile sunt saturate. $(**)$ avem m muchii intr-un cuplaj perfect.$\Rightarrow$ saturam $2m$ noduri.
$(*)$ si $(**) \Rightarrow 2m = n \Rightarrow m = n/2 \Rightarrow |S| = |T| = n/2$

Graful G contine un cuplaj perfect $\Rightarrow$ conform teoremei lui Hall ca $|N_G(A)| \geq |A| \; \forall A \subsetneq S$
Pp ca $\exists A \subsetneq S, |N_G(A)| = |A|$
Toate nodurile din A sunt legate doar de nodurile din $N_G(A)$.
Dar, cum G conex $\Rightarrow$ unul din nodurile din $N_G(A)$ trebuie sa fie legat de un nod din $S \setminus A$
$\Rightarrow \exists y \in N_G(A), \exists x \in S \setminus A$ pentru care $m = xy \in E(G)$
Cum $\forall$ muchie din $E(G) \in$ unui cuplaj perfect $\Rightarrow m \in$ unui cuplaj perfect $\Rightarrow$ Vom avea in A un numar de k elemente, iar in $N_G(A), k-1$ elemente disponibile pentru a realiza cuplajul.
(In contextul in care nu $\exists$ nicio muchie de la nodurile din A catre nodurile din $T \setminus N_G(A)$, iar existenta altor muchii de la nodurile din $T \setminus N_G(A)$ la nodurile din $S \setminus A$ se reduce la acelasi lucru ca apartenenta lui m la un cuplaj perfect).
Nu se poate realiza un cuplaj perfect (ramane un nod izolat)$\Rightarrow$ contradictie $\Rightarrow$
$|N_G(A)| > |A| \Rightarrow$ c.c.t.d.

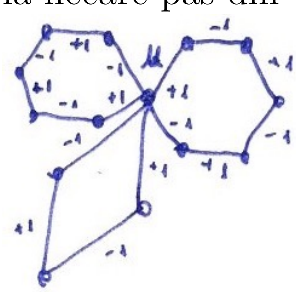
$iii \to i$
$|N_G(A)| > |A|$ pentru $A \subsetneq S$ si $|N_G(A)| = |A|$ pentru $A = S \Rightarrow |N_G(A)| \geq |A|, \forall A \subseteq S$
G graf bipartit $\Rightarrow$ din Teorema lui Hall ca G are un cuplaj perfect.

Stim ca $|N_G(A)| > |A|$ pentru $\forall A \subset S$, unde $|S| = n/2 = k \Rightarrow$ luam acei A pentru care $|A| = k-1$ (luam cazul in care eliminam x din S) $\Rightarrow |N_G(A)| > k-1 \Rightarrow |N_G(A)| = k \Rightarrow$ oricum am elimina pe rand un $y \in N_G(A), |N_G(A)| = |A| = k-1$.Aceasta proprietate se propaga la orice submultime A a lui S, cu mentiunea ca $|N_G(A)| \geq |A|$ (caz in care eliminam mai mult de un nod din S si mai mult de un nod din T)
$\Rightarrow$ oricum elimin 2 noduri, $x \in S$ si $y \in T$, $G - \{x, y\}$ are cuplaj perfect.

Problema 4

a)
Fie C un circuit din G.
M_1 si M_2 cuplaje astfel incat $C = M_1 \cup M_2$ si $a(M_1) \geq a(M_2)$.
Fie $a(M_2) = x$ si $a(M_1) = x + k \Rightarrow$ dupa modificarile efectuate in while, $a(M_2) = x - m_2, a(M_1) = x + k + m_1$, unde m_1, m_2 sunt numarul de muchii din M_1, respectiv M_2, iar $m_1 = m_2$ (G graf bipartit $\Rightarrow C$ lungime para $\Rightarrow$ 2 cuplaje au acelasi numar de muchii)
$\Rightarrow [a(M_2)]^2 = x^2$ si $[a(M_1)]^2 = (x+k)^2$ iar dupa modificarile efectuate in while $[a(M_2)]^2 = (x - m_2)^2, [a(M_1)]^2 = (x + k + m_1)^2 \Rightarrow$ diferenta dintre $\left[\sum_{e \in C} a(e^2)\right]$ la pasul i si pasul urmator va fi: $x^2 - 2xm_2 + m_2^2 + 2xk + k^2 + m_1^2 + 2m_1x + 2m_1k - 2x^2 - k^2 - 2xk = m_2^2 + m_1^2 + 2m_1k > 0$ fiindca k, m_1, m_2 sunt numere pozitive(am redus anumiti termeni tinand cont de faptul ca $m_1 = m_2$).
$\Rightarrow$ la fiecare circuit avut, $\left[\sum_{e \in C} a(e^2)\right]$ va creste cu macar 1 $\Rightarrow$ in total $\left[\sum_{e \in E^+} a(e^2)\right]$ va creste cu cel putin $|C|$, unde $|C|$ este numarul circuitelor existente in G.

b)
G p-regulat $\Rightarrow \forall u \in G(V)$ este legat de p muchii $\Rightarrow$
initial $\left[\sum_{uv \in E^*} a(uv) = p\right]$
La fiecare pas din while, pentru $\forall u$ se va parcurge fiecare circuit incident cu el. Doua muchii incidente cu u care fac parte din acelasi circuit se vor afla in cuplaje diferite $\Rightarrow$ costul uneia dintre muchii va creste cu 1, in timp ce costul celeilalte va scadea cu 1, ramanand astfel, intotdeauna cu un cost constant la fiecare pas din while. Astfel, costul ramane intotdeauna p.



Tema 3

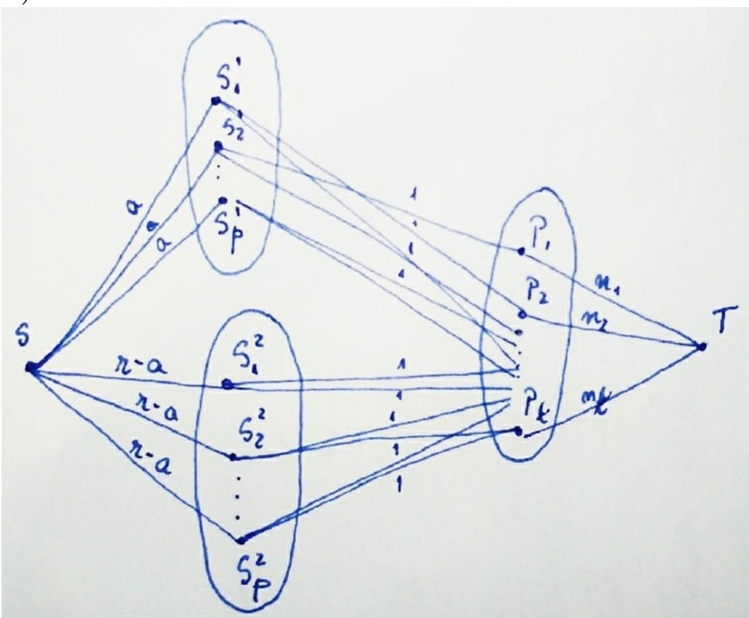
Mocanu Ada-Astrid
&
Ciripescu Teodor

Grupa A7

10 Ianuarie 2020

Problema 2

a)



$V(G) = \{S_1^1, S_2^1, ..., S_p^1, S_1^2, S_2^2, ..., S_p^2\} \bigcap \{P_1, P_2, ..., P_k\} \bigcap \{S, T\}$
$E(G) = \{S_i^1 P_j$, daca profesorul P_j este specializat in licenta studentului $S_i\} \bigcap \{S_i^2 P_j$, daca profesorul P_j nu este specializat in licenta studentului $S_i\} \bigcap \{SS_i^1, i = \overline{1,p}\} \bigcap \{SS_i^2, i = \overline{1,p}\} \bigcap \{P_j T, j = \overline{1,k}\}$

Fie reteaua $R = (G, S, T, c)$ a.i. $c(SS_i^1) = a, \forall i = \overline{1,p}$, $c(SS_i^2) = r - a, \forall i = \overline{1,p}$ $c(P_j T) = n_j, \forall j = \overline{1,k}$
$(S_i^1 P_j) = 1$, $c(S_i^2 P_j) = 1$

b)

Exista o asignare a profesorilor catre studentii ce dau licenta cu proprietatile cerute $\Leftrightarrow$ in $R = (G, S, T, c)$ valoarea maxima a unui flux este: $p * r$
$(\Rightarrow)$

$$x(S_i^1 P_j) = \begin{cases} 1, & \text{daca } \ P_j \text{ este specializat in licenta lui } S_i \\ 0, & \text{altfel} \end{cases}$$

$$x(S_i^2 P_j) = \begin{cases} 1, & \text{daca } \ P_j \text{ este specializat in licenta lui } S_i \\ 0, & \text{altfel} \end{cases}$$

$x(SS_i^1) = a, \forall i$
$x(SS_i^2) = r - a, \forall i$
$x(P_jT) \leq n_j$, unde n_j este nr maxim de echipe din care poate face parte profesorul P_j.
In fiecare nod S_i^1 intra flux de valoare a(de la S) si iese flux de valoare a din ipoteza ($\exists$ a profesori specializati)
In fiecare nod S_i^2 intra flux de valoare $r - a$ si iese flux de valoare $r - a$ din ipoteza ($\exists$ $r - a$ profesori nespecializati)
In fiecare nod P_j intra flux de valoare q_j si iese flux de valoare $q_j \leq n_j$
$\Rightarrow$ Fluxul creat respecta conditia de conservare si e maximal deoarece din nodul S nu poate pleca mai mult.
($\Leftarrow$)
Fie un flux de valoare maxima $p \cdot r$, asociat retelei modelate anterior. El rezolva problema data deoarece putem considera ca nodurile P_j sunt asociate profesorilor, nodurile S_i^1, S_i^2 sunt asociate unui student S_i, muchiile de la multimea S_1 la P corespund modalitatii de alegere a celor a profesori specializati, iar cele de la S^2 la P corespund modalitatii de alegere a celor $r - a$ profesori nespecializati.

c)
$n = 2 + 2p + k$
$m = 2p + k + (x_1 + x_2 + ... + x_p) + (k - x_1 + k - x_2 + ... + k - x_p) = 2p + k + (x_1 + ... + x_p) + kp - (x_1 + ... + x_p) = 2p + k + kp$

1. Analizam algoritmul Ford & Fulkerson
Complexitate in general:$O(n \cdot m \cdot U)$
Consideram ca majorant
$U = r$ (pp $r > n_i, \forall i = \overline{1, k}$)
$n \cdot m \cdot U = (2 + 2p + k)(2p + k + kp)r = (4p + 2k + 2kp + 4p^2 + 4pk + k^2 + 2p^2k + k^2p)r = (4p + 2k + 6pk + 4p^2 + k^2 + 2p^2k + k^2p)r$
$\Rightarrow$ Complexitatea devine:$O(p^2kr)$, pentru $p \geq k$ sau $O(k^2pr)$, pentru $k > p$

2. Analizam algoritmul Edmonds-Karp
Complexitate generala: $O(m^2n)$
$m^2n = (2p+k+kp)^2(2+2p+k) = (4p^2+k^2+k^2p^2+4pk+4kp^2+2k^2p)(2+2p+k)$
$\Rightarrow$ Complexitatea devine: $O(k^3p^2), k \geq p$ sau $O(k^2p^3), k < p$
$\Rightarrow$ cum $r < k \Rightarrow$ mult mai ineficient decat F&F

3. Analizam algoritmul Ahuja-Orlin
Complexitate generala: $O(nm + n^2 \log U)$
$nm + n^2 \log U = (2p + k + kp)(2 + 2p + k) + (2 + 2p + k)^2 \log r = 4p + 4p^2 + 2pk + 2k + 2pk + k^2 + 2kp + 2kp^2 + k^2p + (4 + 4p^2 + k^2 + 4p + 2k + 2pk) \log r$
pentru $p \geq k \Rightarrow$ complexitatea devine: $O(2p^2k + 4p^2 \log r)$
pentru $k > p \Rightarrow$ complexitatea devine: $O(k^2p + k^2 \log r)$
In concluzie, preferam algoritmul F&F.

Problema 3

3.
a)
Consideram circuitul corespunzator lui X_i
Orice nod alegem sa introducem intr-o prima multime, el acopera 2 muchii $\Rightarrow$
Nu este necesar sa introducem in aceeasi multime niciunul dintre vecinii sai directi (Cate una dintre muchiile pe care acestia le-ar putea acoperi, au fost acoperite deja de nodul curent introdus)$\Rightarrow$ Vom lua noduri din 2 in 2 si le punem in prima multime (spre exemplu, prima multime contine nodurile $\{a_{i,1}, a_{i,2}, ..., a_{i,k_i}\}$)
$\Rightarrow$ cea de-a 2-a multime le va contine pe restul, adica pe $\{f_{i,1}, f_{i,2}, ..., f_{i,k_i}\}$ (se procedeaza ca mai sus, alegand insa ca prim nod, un nod f_i)
Ambele multimi au cardinal k_i si au proprietatea ca sunt minime.

b1)
U acopera muchiile lui $G \Rightarrow U$ acopera muchiile lui $H_j, j = \overline{1,m}$
Pentru a avea acces la muchiile lui H_j, trebuie introdus cel putin un nod din H_j. Insa daca introducem doar un nod, nu putem acoperi decat 2 muchii din cele 3 de acoperit.
Astfel, daca introducem 2 noduri este deja suficient pentru a putea acoperi cele 3 muchii de acoperit (nefiind necesar sa le introducem pe toate 3 neaparat)
$\Rightarrow$In multimea U vom avea minim 2 noduri din W_j (multimea nodurilor lui H_j)
$\Rightarrow \mid U \cap V_j \mid \geq 2, \forall j = \overline{1,m}$

b2)
$\mid U \cap (\bigcup_{i=1}^{n} V_i) \mid = \mid U \cap V_1 \mid + \mid U \cap V_2 \mid + ... + \mid U \cap V_n \mid$
cum $\mid U \cap V_1 \mid \geq k_1$ (de la punctul a), $\mid U \cap V_2 \mid \geq k_2$, ... ,$\mid U \cap V_n \mid \geq k_n$
$\Rightarrow \mid U \cap (\bigcup_{i=1}^{n} V_i) \mid \geq k_1 + k_2 + ... + k_n$

$k_1 =$ nr aparitiilor lui x_1 in C
$k_2 =$ nr aparitiilor lui x_2 in C
...
$k_n =$ nr aparitiilor lui x_n in C
$\Rightarrow k_1 + k_2 + ... + k_n =$ numarul aparitiilor tuturor literalilor in $C =$ numarul literalilor din fiecare clauza $*$ nr de clauze $= 3 * m$
$\Rightarrow \mid U \cap (\bigcup_{i=1}^{n} V_i) \mid \geq 3m$

b3)
$\mid U \cap (\bigcup_{i=1}^{n} V_i) \mid \geq 3m$ (1)
$\mid U \cap W_j \mid \geq 2 \Rightarrow \mid U \cap (\bigcup_{j=1}^{m} W_j) \mid \geq 2m$ (2)
Din (1) si (2) $\Rightarrow \mid U \mid = \mid U \cap (\bigcup_{i=1}^{n} V_i) \mid + \mid U \cap (\bigcup_{j=1}^{m} W_j) \mid \geq 5m$
Dar din ipoteza $\mid U \mid = 5m \Rightarrow$
Suntem obligati ca $\mid U \cap (\bigcup_{i=1}^{n} V_i) \mid = 3m$ si $\mid U \cap (\bigcup_{j=1}^{m} W_j) \mid = 2m$
$\Rightarrow \mid U \cap W_j \mid = 2$

b4)
Dupa completarea multimii U algoritmul ne descrie o functie de adevar a.i. $\forall i, t(x_i) = true$ daca si numai daca $U \cap V_i = \{f_{i,1}, f_{i,2}, ..., f_{i,k_i}\}$
Conform algoritmului, intr-o clauza C_j va ramane un nod $w_{j,1}$(spre exemplu) care nu apartine lui U. Daca $w_{j,1}$ corespunde unui nod x_i din clauza C_j, atunci nodul incident la muchia care-l contine pe $w_{j,1}$ este un nod $a_{i,k}$ si $a_{i,k} \in U$
$\Rightarrow t(x_i) = true \Rightarrow t(C_j) = true$
Iar daca $w_{j,1}$ corespunde unui nod $\overline{x_i}$ din clauza C_j, atunci nodul incident la muchia care-l contine pe $w_{j,1}$ este un nod $f_{i,k}$ si $f_{i,k} \in U \Rightarrow t(x_i) = false \Rightarrow t(\overline{x_i}) = true \Rightarrow t(C_j) = true \Rightarrow$ functia satisface toate clauzele din C.

c)

Din modul in care il construim pe $U \Rightarrow$ din fiecare clauza adaugam 2 noduri $\Rightarrow$ in total se adauga 2*nr de clauze $= 2m$ noduri. Apoi, pentru fiecare x_i, daca $t(x_i) = true$, se adauga k_i noduri, iar daca $t(x_i) = false$, se adauga tot k_i noduri $\Rightarrow$ in total $\Rightarrow \sum_{i=1}^{n} k_i = 3m$

$\Rightarrow \mid U \mid = 2m + 3m = 5m$

Ne ramane de demonstrat ca U acopera tot graful G.

i) pentru fiecare clauza $\exists$ 2 noduri care apartin lui $U \Rightarrow$ toate muchiile fiecarei clauze sunt acoperite.

ii) pentru fiecare literal x_i in U se gasesc fie $\{a_{i,1}, a_{i,2}, ..., a_{i,k_i}\}$, fie $\{f_{i,1}, f_{i,2}, ..., f_{i,k_i}\}$ dupa cum $t(x_i) = true$ sau $false$

$\Rightarrow$ cele k_i noduri acopera toate muchiile unui subgraf ce defineste un literal.

iii) In ceea ce priveste muchiile care apartin multimilor A si F, pentru fiecare clauza 2 muchii sunt deja acoperite, prin faptul ca $\exists$ 2 noduri din fiecare clauza care $\in U$.

Pentru cel de-al treilea nod dintr-o clauza, stim ca el este legat de un nod dintr-un literal.

Daca este legat de un nod $a_{i,p}, p = \overline{1, k_i}$, atunci in clauza C_j va aparea x_i $\Rightarrow t(x_i) = true \Rightarrow \{a_{i,1}, a_{i,2}, ..., a_{i,k_i}\} \in U \Rightarrow a_{i,p} \in U \Rightarrow$am acoperit muchia utilizand un nod din U.

Iar daca este legat de un nod $f_{i,p}, p = \overline{1, k_i}$, atunci in clauza C_j va aparea $\overline{x_i} \Rightarrow t(\overline{x_i}) = true \Rightarrow t(x_i) = false \Rightarrow \{f_{i,1}, f_{i,2}, ..., f_{i,k_i}\} \in U \Rightarrow f_{i,p} \in U \Rightarrow$ am acoperit muchia utilizand un nod din U.

Astfel, am reusit acoperirea tuturor muchiilor lui $G \Rightarrow U$ are proprietatea ceruta.

Problema 4

a)
Dandu-se graful G, consideram o $\chi(G)$-colorare(colorare minimala a nodurilor lui G).
Alegem o culoare X.
Niciun nod de culoare X nu poate fi adiacent cu un alt nod de culoare X (asta ar incalca principiul colorarii nodurilor)
$\Rightarrow$ Toate nodurile de culoarea X fac parte dintr-o multime stabila.
Incercam adaugarea de noi noduri la multimea culorii X.
Nu puteam alege niciun nod adiacent cu vreun nod de culoare X $\Rightarrow$ Vom cauta printre nodurile care nu au vecini in multimea culorii X. Cand gasim un astfel de nod, ii schimbam culoarea in X, pastrand proprietatea de multime stabila si minimalitatea colorarii (nu este introdusa nicio culoare noua). Se repeta procedeul pana cand nu se mai pot adauga noduri, obtinand in acest fel o multime stabila maximala pentru cel putin una din clasele de colorare.

b)
Daca x si y au aceeasi culoare $\Rightarrow$
(1) $\chi(G) = \chi(G \mid xy)$
(2)$\chi(G) \leq \chi(G + xy)$ (adaugam o culoare noua cu care il coloram pe unul din ei)
(3)$\chi(G) = \chi(G + xy)$
(4)$\chi(G) \leq \chi(G \mid xy)$ (adaugam o culoare noua cu care sa coloram nodul rezultat)
Din (1),(2) $\Rightarrow \chi(G) = \chi(G \mid xy) \leq \chi(G + xy)$
Din (3),(4) $\Rightarrow \chi(G) = \chi(G + xy) \leq \chi(G \mid xy)$
$\Rightarrow$ in ambele cazuri se alege minimul dintre cele 2
$\Rightarrow \chi(G) = min\{\chi(G + xy), \chi(G \mid xy)\}$