Tema 2, Algoritmica Grafurilor

Bujoreanu Vlad, Paduraru Andra - Elena
Grupa A4, Anul 2

Problema 1.

Solutie. Fie G = (V, E), C - cuplaj perfect in G.
Algoritm:

```
build_tree(G, C)
{
    // A si B vor fi multimi stabile de cardinal maxim in T
    A = {};
    B = {};
    E' = {};
    viz[] = {0};
    queue q;
    a = random(1, |V|);
    q.push(a);
    viz[a] = 1;
    A = A ∪ {a};
    b = perechea lui a din cuplajul perfect C;
    B = B ∪ {b};
    E' = E' ∪ {ab};
    while (q.size() > 0)
    {
        x = q.front(); q.pop();
        foreach t = xy, x ∈ E(G) && viz[y] == 0
        {
                ∃ t' = yz a.i. t' ∈ C
                viz[y] = viz[z] = 1;
                q.push(y); q.push(z);
                if (x ∈ A)
                {
                        B = B ∪ {y};
                        A = A ∪ {z};
                }
                else
                {
```

```
                    A = A ∪ {y};
                    B = B ∪ {z};
                }
                E' = E' ∪ {xy, yz};
            }
            V' = A ∪ B;
        }
        return T = (V', E');
}
```

Corectitudine:
Algoritmul simuleaza o parcurgere in adancime (BFS) de complexitate $\mathcal{O}(|V|+|E|)$.
Fiecare nod este evaluat o singura data datorita vectorului boolean viz.
Pentru fiecare nod putem gasi o pereche in cuplaj, deoarece cuplajul este perfect.
Fiecare muchie are un capat in A si altul in B, astfel ca multimile A si B formeaza bipartitia arborelui T rezultat.
G admite arbore partial $\rightarrow$ este conex $\rightarrow$ fiecare nod este parcurs cel putin o data $\rightarrow$ A si B au cardinal maxim.

$\square$

Problema 2.

Solutie. :

" $\Longrightarrow$ ":
Graful G admite un cuplaj de ordin p $\Longrightarrow$ |G|-2p noduri nu fac parte din cuplaj. Graful G-S poate contine maxim |G|-2p care nu fac parte din cuplajul de ordin p, deci $\forall$ q(G-S), cate un nod cu exceptia a |G|-2p vor avea o extremitate a muchiei in S. Intrucat extremitatile sunt diferite, $\Longrightarrow$ q(G-S) - (|G|-2p) $\leq$ |S| $\Longrightarrow$ q(G-S) $\leq$ |S| + |G| - 2p, $\forall$ S $\subseteq$ G.

" $\Longleftarrow$ ":
Presupunem $\forall$G' cu |G'|$\leq$|G|, daca $\exists$p' a.i. q(G'-S') $\leq$ |S'| + |G'| - 2p', $\forall$ S' $\subseteq$ G' $\Longrightarrow$ cardinalul maxim al unui cuplaj e p'
Demonstratie prin inductie:
P(1), P(2): adevarat
P(G): q(G-S) $\leq$ |S| + |G| - 2p, $\forall$ S $\subseteq$ G
Consideram S_0 a.i. q(G-S_0) = |S_0| + |G| - 2p
Ipoteza inductiva este respectata $\forall$ componenta conexa impara din G-S_0.
q(G-S_0) = |S_0| + |G| - 2p $\Longrightarrow$ |S_0| = q(G-S_0) - |G| + 2p. Avem |G| + 2p noduri ramase neacoperite de cuplaj, deoarece fiecare componenta conexa impara are exact un nod neacoperit, iar acesta poate fi imperecheat cu un nod din S_0 $\Longrightarrow$ avem cuplaj de ordin p.
p este maximal cu aceasta proprietate din implicatia directa $\Longrightarrow$ cuplajul gasit (de ordin p) este maximal.

$\square$

Problema 3.

Solutie. :

a.

" $\implies$ "

Prin eliminarea unei muchii de pe fiecare drum de la u la v, obtinem cel putin doua componente conexe, dintre care intr-una se gaseste u, iar in cealalta se gaseste v.

e $\in$ E(G) - muchie aleasa arbitrar.

Daca $\exists T^*$ APM pentru G a.i. e $\in$ E(T^*) $\implies$ c(T^*) = $min_{T \in T_0}$(c(T))

Printr-o taietura S, se genereaza in graf doua componente conexe G_u si G_v, cu T_u^* si T_v^*, APM ai celor doua componente conexe.

$$\left.\begin{array}{c} c(T^*) = c(T_u^*) + c(T_v^*) + min(S) \\ e \in E(T^*) \end{array}\right\} \Rightarrow e = min(S)$$

$\exists T^* APM, e \in E(T^*) \implies e = min(S), S\ taietura\ in\ G$

" $\impliedby$ "

S - taietura arbitrara in G, e - muchie de cost minim in S, T^* APM a.i. e $\notin$ E(T^*). T^* conex $\implies$ $\exists$ e^* $\in$ E(T^*) a.i. $e^* \in E(S)$.

Arborele T' = (V, E'), E' = E(T^*) $\setminus\{e^*\}$ $\cup$ {e}.

$$\left.\begin{array}{c} c(T') = c(T^*) - c(e^*) + c(e) \\ c(e) = min(S) \leq c(e^*) \end{array}\right\} \Rightarrow c(T') \leq c(T^*)$$

T^* este APM $\implies$ c(T') = c(T^*), deci T' este un alt APM care il contine pe $\implies$ e = min(S), S taietura in G $\implies$ $\exists T^*$ APM, a.i. e $\in$ E(T^*) $\iff$ $\exists S$ taietura in G, a.i. e are cost minim in S.

b.

" $\implies$ "

$\exists$ S taietura in G, e = uv muchie de cost minim in S. Daca $\nexists$ C circuit in G a.i. e $\in$ C $\implies$ e nu are cost maxim intr-un circuit.

Daca $\exists$ $C_0, ..., C_k$ a.i. e $\in$ C (oricare circuit de la u la v), atunci $\exists e_1 \in C_1, ..., e_k \in C_k$ a.i. $\{e_1, ..., e_k\}$ $\subset$ S.

e = min(S) $\implies$ c(e) <c(e_1), ..., c(e) <c(e_k) $\implies$ $\exists$ cel putin o muchie cu cost mai mare decat e $\implies$ e nu este muchie de cost maxim pe niciun circuit.

" $\impliedby$ "

Consideram e = uv, daca e nu este muchie de cost minim pe nicio taietura $\implies$ $\exists$ cel putin inca un drum de la u la v. Fie $D_1, D_2, ..., D_k$ toate drumurile de la u la v. Multimea tuturor muchiilor din aceste drumuri genereaza o taietura in G. Intrucat e nu este de cost minim, c(e) este mai mare decat costurile tuturor celorlalte muchii din $D_1, D_2, ..., D_k$. Fiecare din aceste drumuri reunesc toate circuitele care il includ pe e, iar c(e) fiind mai mare decat a tuturor celorlalte muchii de pe circuite $\implies$ c(e) are cost maxim in cel putin un circuit. Daca e nu este de cost maxim pe vreun circuit $\implies$ e este de cost minim intr-o taietura.

$\nexists T^*$ APM a.i. e $\in$ E(T^*) $\iff$ e este de cost maxim intr-un circuit

c.
$\forall$ e - muchie aleasa arbitrar, aceasta este fie de cost minim intr-o taietura, fie de cost maxim intr-un circuit, iar asta presupune o colorare in verde sau albastru, dar e este ales arbitrar $\implies$ nu se opreste cata vreme mai exista muchii rosii in graf care pot fi colorate fie in albastru, fie in verde.

d.
La inceput toate muchiile vor fi colorate cu rosu. Alegand mereu o muchie colorata cu rosu, aceasta fie va fi colorata in albastru, fie in verde (rezulta din punctul anterior), deci numarul muchiilor colorate cu rosu va scadea la fiecare iteratie a buclei while. Cum multimea muchiilor este finita, algoritmul se va opri.

La fiecare iteratie a buclei while exista o unica muchie e care sa indeplineasca conditia de cost minim in taietura si cost maxim in circuit, deci vom avea o unica succesiune de muchii care va constitui in final APM-ul H.

$\square$

Problema 4.

Solutie. :

a.
$T_i = T \setminus \{k_1, k_2, ..., k_{i-1}\}$
Eliminand oricare frunza din arbore, acesta ramane in continuare conex si aciclic $\forall$ i $\in$ {1, 2, ..., p-1}
$T_{p-1} = T \setminus \{k_1, k_2, ..., k_{p-2}\}$ = {u,p}, T_{p-1} arbore $\implies$ up $\in$ E(T_{p-1}), u, p, frunze.
k_{p-1} = min (u, p) = u $\implies$ x_{p-1} = p $\forall$ T - construit arbitrar.

b1.
T_k = (V, E)
V(T_k) = $\{i_k, ..., i_{p-1}\} \cup \{x_k, ..., x_{p-1}\}$
E(T_k) = $\{i_k x_k, i_{k+1} x_{k+1}, ..., i_{p-1} x_{p-1}\}$
$\forall k_1, k_2, k_1 \leq k_2$
i_{k_2} = min(V $\setminus \{i_1, ..., i_{k_1}, ..., i_{k_2-1}, x_{k_2}, ..., x_{p-1}\}$ $\implies$ $i_{k_2} \neq i_{k_1}$ $\implies$ $i_k \notin \{i_{k_1}, ..., i_{p-1}\}$
i_k = min(V $\setminus \{i_1, ..., i_{k-1}, x_k, ..., x_{p-1}\}$ $\implies$ $i_k \notin \{x_k, ..., x_{p-1}\}$
$\implies$ i_k face parte doar din muchia $i_k x_k$ $\implies$ i_k este frunza in T_k

b2.
V(T_k) = $\{i_k, i_{k+1}, ..., i_{p-1}\} \cup \{x_k, x_{k+1}, ..., x_{p-1}\}$
i_k = min(V $\setminus \{i_1, ..., i_{k-1}, x_k, ..., x_{p-1}\}$) $\implies$ i_k < u, $\forall$ u $\in$ $\{i_k, ..., i_{p-1}\} \setminus \{x_k, ..., x_{p-1}\}$ ①

x_i - arbitrar, i > k, daca $x_i = x_j$, i $\neq$ j, j > k $\implies$ x_i incident in $i_i x_i$ si in $i_j x_j$, $\implies$ x_i nu este frunza.
$x_i \neq x_j$, $\forall$ j $\neq$ i, j > k $\implies$ $\exists$ j > i a.i. x_i = min (V $\setminus \{i_1, ..., i_{j-1}, x_j, ..., x_{p-1}\}$) $\implies$ x_i incident in $i_i x_i$ si in $i_j x_j$, deci x_i nu este frunza.
x_i - arbitrar $\implies$ $\nexists$ u frunza, a.i. u $\in$ $\{x_k, ..., x_{p-1}\}$ ②

①, ② $\implies$ i_k este cea mai mica frunza din T_k

b3.
T_k = T $\setminus$ $\{i_1, ..., i_{k-1}\}$
pct. b2 $\implies$ i_k cea mai mica frunza din T_k
$i_k x_k \in$ E(T_k)
Daca x_k se afla pe pozitia k in emblema lui T $\forall$ k $\in$ {1, ..., p-1} $\implies$ x este emblema lui T. □