

## Tema 2

Student name: *Chirila Gabriela-Valentina 2A4, Ungurean Ana-Maria 2A4*

---

Course: *Algoritmica Grafurilor* – Professor: *Lect dr. Olariu Florentin - Emanuel*  
Due date: *18 Noiembrie 2022*

### Problema 1

**1. (a).** Fie  $\mathcal{U}$  o partiție de cardinal  $p$  a lui  $V$  formată din mai multe submulțimi nevide și disjuncte,  $U_1, U_2 \dots U_p$ , astfel încât reuniunea lor va fi  $V$ . Presupunem prin reducere la absurd că  $p=1$  toate nodurile aparțin aceleiași partiții cu proprietatea că între oricare două noduri ale acestei partiții nu avem drum rezultă că graful respectiv nu poate fi conex. Pentru  $p = 2$  vom avea două partiții cu aceeași proprietate, însă graful poate fi conex. Știind că  $G = (V, E)$  este un graf conex, ajungem la concluzia că valoarea minimă pe care o poate avea  $p$  este 2.

Dat fiind  $T$  un arbore parțial al lui  $G = (V, E)$  obținut prin eliminarea unor muchii din graful  $G$ , fiind aciclic și având  $n$  noduri și  $n-1$  muchii. Presupunem prin reducere la absurd că  $T$  conține  $p-2$   $\mathcal{U}$ -muchii. Dar pentru cazul în care  $p = 2$ , acest lucru este imposibil deoarece nu putem avea arbore cu 0  $\mathcal{U}$ -muchii. Așadar,  $T$  conține cel puțin  $p-1$   $\mathcal{U}$ -muchii, în cazul în care  $p = 2$  vom obține un arbore cu două noduri și o muchie aceea fiind o  $\mathcal{U}$ -muchie.

**1. (b).** Dacă  $G$  are  $s$  arbori parțiali disjuncți pe muchii înseamnă că muchiile pe care le găsim într-un arbore nu le vom găsi în alt arbore. Ținând cont de demonstrația de la punctul a, știm că un arbore parțial are cel puțin  $p-1$   $\mathcal{U}$ -muchii, rezultă că în graful  $G$  există cel puțin  $s(p-1)$   $\mathcal{U}$ -muchii.

### Problema 2

**2. (a).** Știm că o muchie backward are orientarea de la strămoși la fii. Inițial, digraful  $T$  este aciclic, fiind arbore. Fie  $\forall$  muchie backward  $vu$  presupunând că am terminat realizarea orientării tuturor muchiilor backward din graful inițial, această muchie, conform definiției, are orientarea de la părinți la fii  $\Rightarrow v$  este strămoș a lui  $u$ . Observăm că se formează exact un ciclu deoarece de la  $u$  avem drum către  $v$  (drum format doar din arcele din arbore) dat fiind faptul că în arbore orientarea este de la descendenți la strămoși (un descendent nu poate avea mai mulți strămoși, ci exact unul).  $\Rightarrow \forall$  muchie backward  $vu$ , digraful  $T + vu$  conține exact un circuit direct  $C_{vu}$ .

2. (b). Pseudocodul este :

```

for (  $i = \overline{1, n}$  )
{
     $visited[i] \leftarrow 0$  //considerăm toate nodurile nevizitate
     $circuit[i] \leftarrow \emptyset$  //circuit[i]- reprezinta toate circuitele din nodul i
}
for (  $i = \overline{1, n}$  )
{
     $nodv \leftarrow vertex[i]$  //în vertex avem nodurile în ordine crescătoare dfs.
     $visited[nodv] \leftarrow 1$  //marcam nodul ca fiind vizitat
    while(  $\epsilon \neq \emptyset$  ) //în coada  $\epsilon$  avem muchiile backward
    {
         $\epsilon[nodv].pop()$ 
         $nodu \leftarrow \epsilon[nodv].front()$ 
         $\epsilon[nodv].pop()$ 
         $circuit[nodv].push(nodu)$ 
        if (  $visited.[nodu] == 0$  )
        {
             $circuit[nodv].push(nodu)$ 
            while (  $visited[parent[nodu]] == 0$  ) //adăugăm în coada circuit noduri până la primul
            nod vizitat
            {
                 $circuit[nodv].push(parent[nodu])$ 
                 $nodu \leftarrow parent[nodu]$ 
            }
        }
    }
}

```

După executarea pseudocodului de mai sus, în coada circuit pentru fiecare nod din  $G$  vom avea circuitele din nodul respectiv, iar toate drumurile din  $G$  se află în parcurgerea circuitelor.

2. (c). În timpul determinării circuitului  $C_{vu}$  ce are loc în bucla **while** (  $visited[parent[nodu]] == 0$  ), vom adăuga în coada circuit nodurile din secvența while, aceste noduri formează un drum sau un circuit începând cu nodul  $v$  deoarece prima muchie pe care o adăugăm este o muchie backward de forma  $vz$ , unde  $z$  este strămoșul lui  $v$ ,  $\Rightarrow$  nodurile pe care le adăugăm ulterior vor forma un drum sau un circuit.

2. (d). Prima legătură este un circuit deoarece presupunând că pornim din nodul  $v$ , pe parcursul determinării circuitului  $C_{vu}$  nu vom întâlni un nod vizitat până nu ajungem din nou la  $v$ , dat fiind faptul că  $v$  este primul nod vizitat. Astfel, se va determina un circuit ca fiind prima legătură.

2. (e). Un graf  $G$  este  $p$ -conex dacă  $G$  nu poate fi deconectat prin ștergerea a mai puțin de  $p$  noduri.  $\delta(G)$  este gradul minim al nodurilor din  $G$ .

Presupunem prin reducere la absurd că  $\delta(G) = 1$ , prin eliminarea unui singur nod, cel care are gradul 1, am putea deconecta graful  $\Rightarrow G$  este 1-conex  $\Rightarrow G$  este 2-conex, dacă  $\delta(G) \geq 2$ .

Descompunerea dată conține doar o singură legătură care este circuit, aceasta fiind prima legătură găsită, fapt demonstrat la punctul d) deoarece nu vom găsi o altă legătură care să fie circuit, având în vedere că vom găsi noduri vizitate până să ajungem în nodul din care am plecat.

### Problema 3

**3. (a).** Algoritmul lui Kruskal identifică arborele parțial de cost minim dintr-un graf. Acest algoritm constă în ordonarea muchiilor crescător după cost și analizarea lor în această ordine. Dacă la analizare, extremitățile muchiei fac parte din același subarbore, muchia se ignoră, iar dacă extremitățile muchiei fac parte din arbori diferiți, aceștia se vor reuni, iar muchia va face parte din arborele parțial de cost minim.

1. Știm că  $e_1 = u_1v_1$  și  $e_2 = u_2v_2$ , unde  $v_1, v_2 \notin V_s \Rightarrow e_1, e_2 \notin T_s$  încă.

2. Dat fiind faptul că Algoritmul lui Kruskal analizează muchiile în ordinea crescătoare după cost, iar costurile sunt diferite pentru fiecare muchie deoarece funcția  $c$  este injectivă  $\Rightarrow e_1, e_2$  încă nu au fost analizate deoarece au costul mai mare decât costul muchiilor analizate deja.

Din 1. + 2.  $\Rightarrow c(e_1) > c(e)$  și  $c(e_2) > c(e)$ , unde  $e$  este o muchie de cost maxim din drumul de la  $u_1$  la  $u_2$  în  $T_s$ ,  $u_1, u_2 \in V_s$ .

**3. (b).** Algoritmul lui Prim identifică arborele parțial de cost minim dintr-un graf conex ponderat. Acest algoritm constă în alegerea unui nod arbitrar pentru început, iar apoi în mod repetat se alege câte un nod care nu a fost vizitat încă și care are muchie cu unul dintre nodurile deja vizitate, această muchie fiind de cost minim.

1. Știm că  $e_1 = u_1v_1$  și  $e_2 = u_2v_2$ , unde  $v_1, v_2 \notin V_s \Rightarrow e_1, e_2 \notin T_s$  încă.

2. Dat fiind faptul că Algoritmul lui Prim tot alege noduri spre care costul muchiei este minim. După un anumit număr de iterații a buclei while,  $\forall u_1, u_2 \in V_s$ , muchia  $e$  de cost maxim de pe drumul de la  $u_1$  la  $u_2$  va fi mai mică cel puțin decât costul unei muchii care pleacă din aceste două noduri spre un nod care  $\notin V_s$  deoarece în timpul analizei costurilor muchiilor până în acel moment, am ales costul minim a unei muchii care pleacă dintr-un nod din  $V_s$ . Presupunem că  $e = u_3u_4$ ,  $c(e)$  poate fi mai mare decât  $c(e_1)$  sau decât  $c(e_2)$  în cazul în care ultimul nod adăugat în  $V_s$  a fost  $u_1$  sau  $u_2$ , neavând șansa să comparăm și costurile muchiilor care pleacă din ultimul nod adăugat.

Din 1. + 2.  $\Rightarrow c(e_1) > c(e)$  sau  $c(e_2) > c(e)$ , unde  $e$  este o muchie de cost maxim din drumul de la  $u_1$  la  $u_2$  în  $T_s$ ,  $u_1, u_2 \in V_s$ .

### Problema 4

**4. (a).** Presupunem prin reducere la absurd că  $G'$  este un digraf ciclic și are un circuit de forma  $a-b-a$ , unde  $a, b \in V(G')$ . Așadar, pentru a avea acest circuit avem nevoie de muchia  $ab$  și muchia  $ba$ .

Conform definiției acoperirii prietenești, vom avea:

1. muchia  $ab \in E(G') \Leftrightarrow$  există drum direct de la  $a$  la  $b$  în  $G$
2. muchia  $ba \in E(G') \Leftrightarrow$  există drum direct de la  $b$  la  $a$  în  $G$

Din 1. + 2.  $\Rightarrow$  având drum de la  $a$  la  $b$  și de la  $b$  la  $a$  în  $G$  avem un circuit de la  $a$  la  $a \Rightarrow G$  este un digraf ciclic, însă acest rezultat este o contradicție pentru ipoteza noastră.  
 $\Rightarrow$  acoperirea prietenească a unui digraf aciclic este tot un digraf aciclic.

**4. (b).** Știm că o mulțime stabilă este o mulțime formată din noduri neadiacente două câte două.

Demonstrăm că o submulțime de noduri  $S \subseteq V$  este stabilă în  $G' \Leftrightarrow$  este o adversitate în  $G$  astfel:

" $\Rightarrow$ ": Știm că  $S$  este o mulțime stabilă în  $G'$  și arătăm că este o adversitate în  $G$ . Dacă  $S$  este o mulțime stabilă în  $G'$ , atunci nu avem niciun arc de la un nod la alt nod, ambele fiind din  $S \Rightarrow$  nu avem drum direct între două noduri din  $S$  în digraful  $G \Rightarrow S$  este o adversitate în  $G$ .

" $\Leftarrow$ ": Știm că  $S$  este o adversitate în  $G$  și arătăm că este o mulțime stabilă în  $G'$ . Dacă  $S$  este o adversitate în  $G$ , atunci nu avem niciun drum între oricare două noduri, ambele fiind din  $S \Rightarrow$  nu avem arc între două noduri din  $S$  în digraful  $G' \Rightarrow S$  este o mulțime stabilă în  $G'$ .

**4. (c).** Știm că un drum direct simplu este un drum în care arcele nu se repetă. Demonstrăm că un șir de noduri  $x_1, x_2 \dots x_p$  din  $V$  formează un drum direct simplu în  $G' \Leftrightarrow$  nodurile se regăsesc (în această ordine, dar nu neapărat consecutive) pe un drum direct simplu în  $G$  astfel:

" $\Rightarrow$ ": Știm că  $x_1, x_2 \dots x_p$  din  $V$  formează un drum direct simplu în  $G'$  și arătăm că nodurile se regăsesc (în această ordine, dar nu neapărat consecutive) pe un drum direct simplu în  $G$ .

Fie:

1.  $x_1x_2$  un arc în  $G' \Rightarrow$  există drum direct de la  $x_1$  la  $x_2$  în  $G$

2.  $x_2x_3$  un arc în  $G' \Rightarrow$  există drum direct de la  $x_2$  la  $x_3$  în  $G$

.

p-1.  $x_{p-1}x_p$  un arc în  $G' \Rightarrow$  există drum direct de la  $x_{p-1}$  la  $x_p$  în  $G$

Din 1+2+...+(p-1) și știind că un drum direct de la un nod la alt nod în  $G$  conține cel puțin două noduri  $\Rightarrow$  există drum direct simplu de la  $x_1$  la  $x_p$  în  $G$  de forma  $x_1, x_2 \dots x_p$  (în această ordine, dar nu neapărat consecutive).

" $\Leftarrow$ ": Știm că  $x_1, x_2 \dots x_p$  (în această ordine, dar nu neapărat consecutive) este un drum direct simplu în  $G$  și arătăm că  $x_1, x_2 \dots x_p$  formează un drum direct simplu în  $G'$ .

Fie:  $x_n, x_m, x_t$  o succesiune de noduri consecutive din drumul direct simplu dat  $\Rightarrow$  în  $G'$  vom avea:

1. arcul  $x_nx_m$

2. arcul  $x_mx_t$

3. arcul  $x_nx_t \Rightarrow$  în  $G'$  putem avea un drum direct simplu mai scurt de la  $x_n$  la  $x_t$ .

$\Rightarrow$  în  $G'$  putem avea un drum direct simplu mai scurt de la  $x_1$  la  $x_p$  decât cel din  $G$ .

**4. (d).** Demonstrăm că  $G'$  poate fi partiționat în cel mult  $s$  drumuri disjuncte pe noduri  $\Leftrightarrow G$  poate fi acoperit cu cel mult  $s$  drumuri astfel:

" $\Rightarrow$ ": Știm că  $G'$  poate fi partiționat în cel mult  $s$  drumuri disjuncte pe noduri și arătăm că  $G$  poate fi acoperit cu cel mult  $s$  drumuri. Dacă  $G'$  poate fi partiționat în cel mult  $s$  drumuri disjuncte pe noduri  $\Rightarrow$  (folosindu-ne de punctul  $c$ ) Nodurile din cele cel mult  $s$  drumuri disjuncte din  $G'$  se regăsesc (în această ordine, dar nu neapărat consecutive) în cel mult  $s$  drumuri din  $G \Rightarrow G$  poate fi acoperit cu cel mult  $s$  drumuri.

" $\Leftarrow$ ": Știm că  $G$  poate fi acoperit cu cel mult  $s$  drumuri și arătăm că  $G'$  poate fi partiționat în cel mult  $s$  drumuri disjuncte pe noduri. Dacă  $G$  poate fi acoperit cu cel mult  $s$  drumuri  $\Rightarrow$  în  $G'$  vom avea aceleași și toate drumurile din  $G$  deoarece  $G'$  se formează din  $G$  și eventual se mai adaugă arce între nodurile între care există drum în  $G \Rightarrow G'$  poate fi partiționat în cel mult  $s$  drumuri disjuncte pe noduri.