

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**TAIKOMOSIOS INFORMATIKOS KATEDRA**

**DISKREČIOSIOS STRUKTŪROS (P170B008)**

**KURSINIS DARBAS**

**Užduoties nr. A6**

Atliko:

IFF-1/8 gr. studentas

Matas Palujanskas

Priėmė:

Doc. Martynas Patašius

KAUNAS

2022

## Turinys

1.	Užduotis (A6) .....	3
2.	Užduoties analizė.....	3
3.	Programos algoritmo aprašymas .....	4
4.	Programos tekstas .....	4
5.	Testavimo pavyzdžiai.....	6
	<b>PIRMAS TESTAS .....</b>	<b>6</b>
	<b>ANTRAS TESTAS .....</b>	<b>7</b>
	<b>TREČIAS TESTAS .....</b>	<b>8</b>
6.	Išvados .....	9
7.	Literatūros sąrašas .....	9

## 1. Užduotis (A6)

Nudažyti grafo viršūnes pagal principą „pirma spalva, po to viršūnė“.

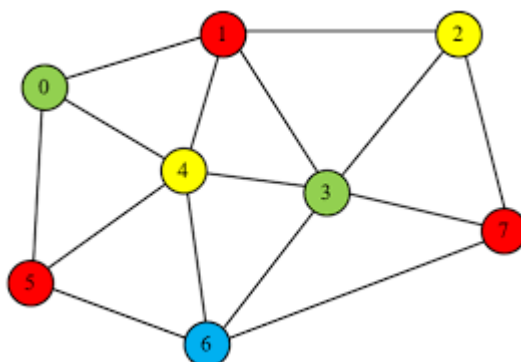
## 2. Užduoties analizė

„Pirma spalva, po to viršūnė“ ir „pirma viršūnė, po to spalva“ yra euristikos idėjos, kuriomis remiantis dažniausia naudojama chromatinio skaičiaus apskaičiavimo uždaviniui spręsti. Kadangi šis uždavinys yra NP pilnasis ir neturi pakankamai efektyvių algoritmų jam išspręsti. Todėl naudojamos šios euristikos. Chromatinio skaičiaus uždavinys – tai diskrečiojo optimizavimo uždavinys, kuriam taikomi ribojimai, jog kiekviena viršūnė turi būti nudažoma viena spalva bei, bet kokios gretimos viršūnės turi būti nudažytos skirtinga spalva. Šioje ataskaitoje bus nagrinėjamas „pirma spalva, po to viršūnė“ algoritmas. Taigi, visų pirma grafo viršūnės yra išrikiuojamos laipsnių mažėjimo tvarka. Tuomet yra keliama sąlyga: kol yra nenudažytų viršūnių – tol dažyti. Pradedama pirmąja nenudažyta viršūne sekoje, sp spalva nuosekliai dažoma viena po kitos. Taip vykdomas algoritmas, kol nudažomos visos grafo viršūnės.

**Uždavinys.** Yra duota gretimumo matrica briaunM. Iš jos yra sudaromas grafas  $G = (virsSar, briaunAib)$ . virsKiek – grafo viršūnių skaičius, briaunKiek – grafo briaunų skaičius, virsSar – viršūnių aibės, o BriaunAib – briaunų aibės. Reikia nudažyti bei pavaizduoti grafą, taip pat išvedamas ir grafo chromatinis skaičius (t.y. kiek spalvų buvo naudota dažant grafą).

### Metodo idėja.

Visų pirmiausia visos grafo viršūnės yra rikiuojamos pagal jų laipsnius, mažėjimo tvarka. Tuomet imama pirma sekos nenudažyta viršūnė ir dažoma sp spalva, toliau visos sekančios viršūnės yra dažomos sp spalva, jei jos nėra gretimos sp spalvos nudažytai viršūnei, jei yra, tuomet yra keičiama spalva bei taip dažoma, kol nelieka nei vienos nenudažytos viršūnės.



1 pav. Grafo dažymas pagal principą: „pirma spalva, o po to viršūnė“

Pirmąja spalva dažome 0-ąją viršūnę, po to pirmąja spalva galime dažyti 3-ąją viršūnę. Gauname:

0,	1,	2,	3,	4,	5,	6,	7
I			I				

Imame antrąją spalvą ir dažome pirmąją nenudažytą sekos viršūnę – 1-ąją, bei 5-ąją ir 7-ąją viršūnes. Gauname:

0,	1,	2,	3,	4,	5,	6,	7
I	II		I		II		II

Trečiaja spalva pirmiausia dažome 2-ąją viršūnę ir 4-ąją viršūnę. Gauname:

0,	1,	2,	3,	4,	5,	6,	7
I	II	III	I	III	II		II

Galiausiai 4-ąją spalva dažome paskutinę likusią viršūnę – 6. Gauname:

0,	1,	2,	3,	4,	5,	6,	7
I	II	III	I	III	II	IV	II

Dažymo algoritmas yra baigtas. Iš gautų rezultatų galime pastebėti, jog algoritmui prireikė 4 spalvų nudažyti visas viršūnes, tad chromatinis skaičius bus 4.

### 3. Programos algoritmo aprašymas

Programos veikimo principas: bus nurodomas grafas, kuris bus aprašomas briaunų matrica, programa atvaizduos nuspaltintą grafą bei išves grafo chromatinį skaičių. Pradžioje chromatinis skaičius yra lygus 0 ( $sp = 0$ ), taip pat yra paruošiamas briaunų masyvas ( $briaunAib$ ), kuriame viršūnės jau yra surikiuotos jų laipsnių mažėjimo tvarka. Toliau yra keliama sąlyga: kol visos viršūnės nėra nudažytos, tol dažome. Jei *nud* ( $nud = 0$ ) lygus viršūnių skaičiui *virsKiek*, visos viršūnės nudažytos. Toliau yra imama viena viršūnė ir ji dažoma, jei jokia gretima tai viršūnei nėra jau nudažyta ta pačia spalva. Spalvos, kurios jau yra panaudotos yra išsaugomos d masyve. Toliau yra dažomos visos likusios viršūnės kol nebelieka ką dažyti t.y. nebėra ne gretimų spalvų, tuomet pasirenkama kita spalva ir vėl einama per nudažytas viršūnes. Taip vykdoma kol nebelieka nudažytų viršūnių. Galiausiai programa baigia savo darbą.

### 4. Programos tekstas

#### *Virsuniu\_spalvinimas\_Matas\_Palujanskas.m*

```
clc; clear; close all;

% briaunM = [ 1 3 5 7 8 7;
%           2 4 4 8 9 9 ]; %Gretimumo matrica
briaunM = [ 2 3 4 5 8 7 4 6 6 5 5 5;
           3 6 7 3 3 2 8 2 8 2 1 4];
% briaunM = [1 4 7 8 5;
%           2 2 1 6 4];

virsKiek = max(max(briaunM)); %- viršūnių kiekis,
briaunKiek = length(briaunM(1,:)); %- briaunų kiekis,
d(1:virsKiek) = 0; %- viršūnių spalvų masyvas;
virsSar = 1:virsKiek; % Viršūnių sąrašas
briaunAib = []; % į briaunAib struktūrą perrašoma briaunu matrica briaunM
for BrNr = 1:briaunKiek
    briaunAib{BrNr} = briaunM(:, BrNr);
end

% Viršūnių laipsnių apskaičiavimas
for i = 1:2
    for j = 1:briaunKiek
        k = briaunM(i,j);
        d(k) = d(k) + 1;
    end
end
```

```

end

% { BLlst(n, m, b, L, lst) }
lst(1:virsKiek+1) = 0;
for i = 1:virsKiek
    lst(i+1) = lst(i) + d(i);
end
fst = lst + 1;
L(1:2*briaunKiek) = 0;
for j = 1:briaunKiek
    k = briaunM(1,j);
    L(fst(k)) = briaunM(2,j);
    fst(k) = fst(k) + 1;
    k = briaunM(2,j);
    L(fst(k)) = briaunM(1,j);
    fst(k) = fst(k) + 1;
end

tic

% Kintamiesiems ir masyvams pradinių reikšmių priskyrimas
for i = 1:virsKiek
    v(i) = i; %{ Masyve v iš eilės surašomi viršūnių numeriai }
    s(i) = lst(i+1) - lst(i); %{s[i] - i-tosios viršūnės laipsnis }
    d(i) = 0;
end

% Viršūnės rikiuojamos laipsnių mažėjimo tvarka
for k = 1:virsKiek-1
    for i = 1:virsKiek-k
        if s(i) < s(i+1) % Keičiame vietomis s[i] su s[i + 1] ir v[i] su v[i + 1]
            z = s(i);
            s(i) = s(i+1);
            s(i+1) = z;

            z = v(i);
            v(i) = v(i+1);
            v(i+1) = z;
        end
    end
end

%jei d[i] = k, tai reiškia, kad i-oji viršūnė dažoma k-ąja spalva.
sp = 0; % spalvų kiekis
nud = 0; % nudažytų viršūnių kiekis
while nud < virsKiek % vykdoma kol yra nors viena nenudazyta virsune
    sp = sp + 1;
    for i = 1:virsKiek
        u = v(i);
        if d(u) == 0 % Jei virsune nera nudazyta
            % ar is nenudazytu virsuniu yra virsune kuri yra gretima u
            % virsunei nudazyta sp spalva
            j = lst(u) + 1;
            t = false;
            while (j <= lst(u+1)) && ~t
                x = L(j);
                if d(x) == sp
                    t = true;
                else
                    j = j+1;
                end
            end
            if ~t % jei ne t, tai virsune galima dazyti sp spalva
                d(u) = sp;
                nud = nud + 1;
            end
        end
    end
end

```

```

        end
    end
end
end

%skaiciavimoLaikas = toc;

figure(1)
title('Grafo virsunes nudazytos pagal algoritma')
Vkor = plotGraphVU1(virsSar, briaunAib, 0, 0, [], 0, 10, 1, 'b');
vspalvos='rgbcmyk';
disp('Viršune    Viršune buvo nudazyta');
for i = 1:virsKiek
    fprintf('%5d %6d\n',i,d(i));
    Vkor1 = Vkor(i,:);
    figure(1);
    r = 0.08; % viršūnės spindulys
    x = Vkor1(1); y = Vkor1(2); % virsSar(i) viršūnės koordinatės
    rectangle('Position',[x-r,y-
r,2*r,2*r],'Curvature',[1,1],'FaceColor',vspalvos(d(i)));
    if abs(i)<10
        shiftx = 0.2*r;
    else
        shiftx = 0.6*r;
    end
    str = sprintf('%d', abs(i));
    text(x-shiftx, y, str);
end

disp('Panaudotu spalvu kiekis(chromatinis skaicius):');
disp(sp);
%disp('Skaiciavimo laikas:');
%disp(skaiciavimoLaikas);

```

Grafo spausdinimui naudojama funkcija:

plotGraphVU1.m

```

function Vkor = plotGraphVU1(V,U,orgraf,arc,Vkor,poz,FontSize,lstor,spalva)
...

```

Funkcijos turinys nekeistas

## 5. Testavimo pavyzdžiai

Buvo panaudoti trys testavimo pavyzdžiai:

### *Pirmas testas*

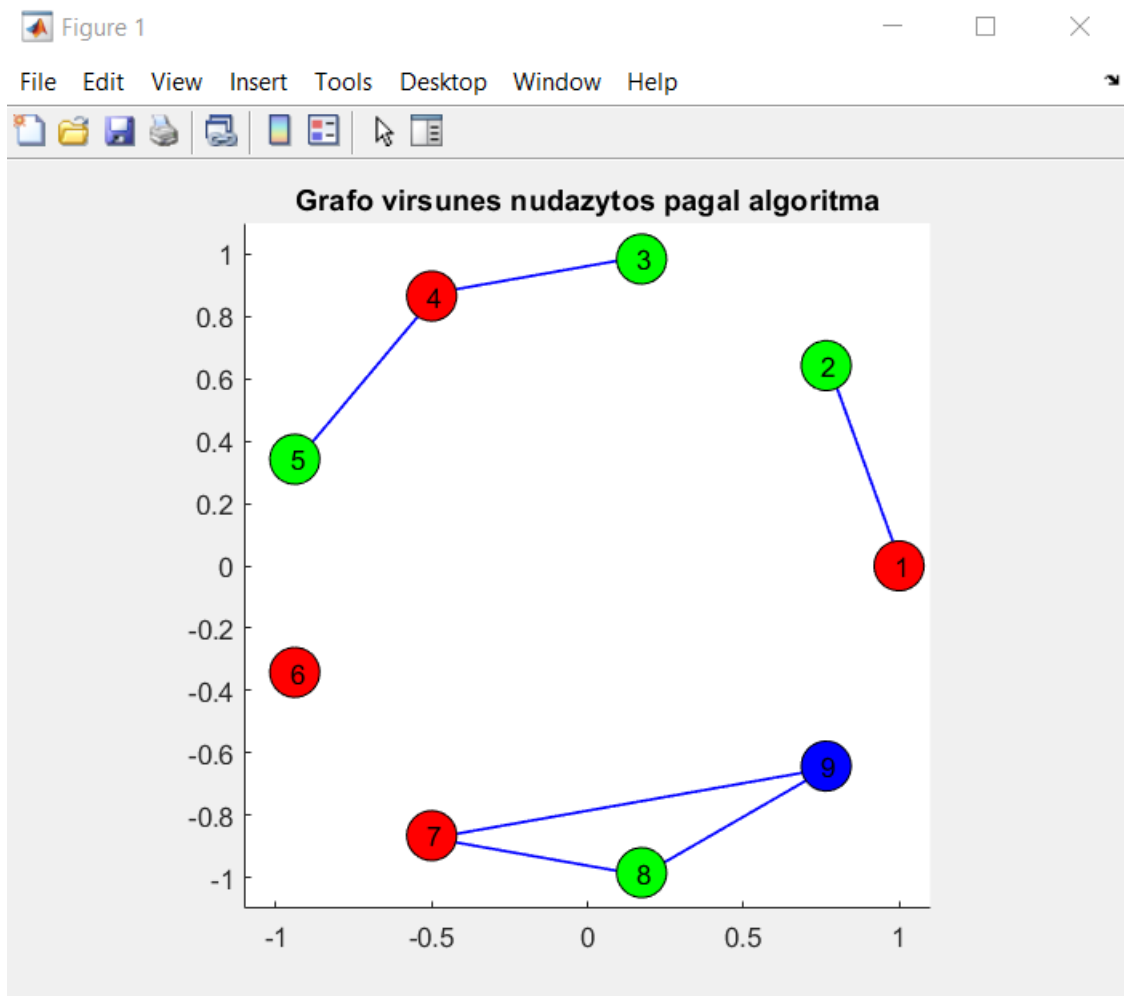
Pirmojo pavyzdžio grafas pavaizduotas 2 pav., kai duota briaunų matrica:

```

briaunM = [ 1 3 5 7 8 7;
            2 4 4 8 9 9 ];

```

Šis grafas buvo nudažytas 3 spalvomis, tad chromatinis skaičius yra 3. Programos rezultatas:



**2 pav.** Nuspalvintas grafas, chromatinis skaičius yra 3

Virsune	Virsune buvo nudazyta
1	1
2	2
3	2
4	1
5	2
6	1
7	1
8	2
9	3

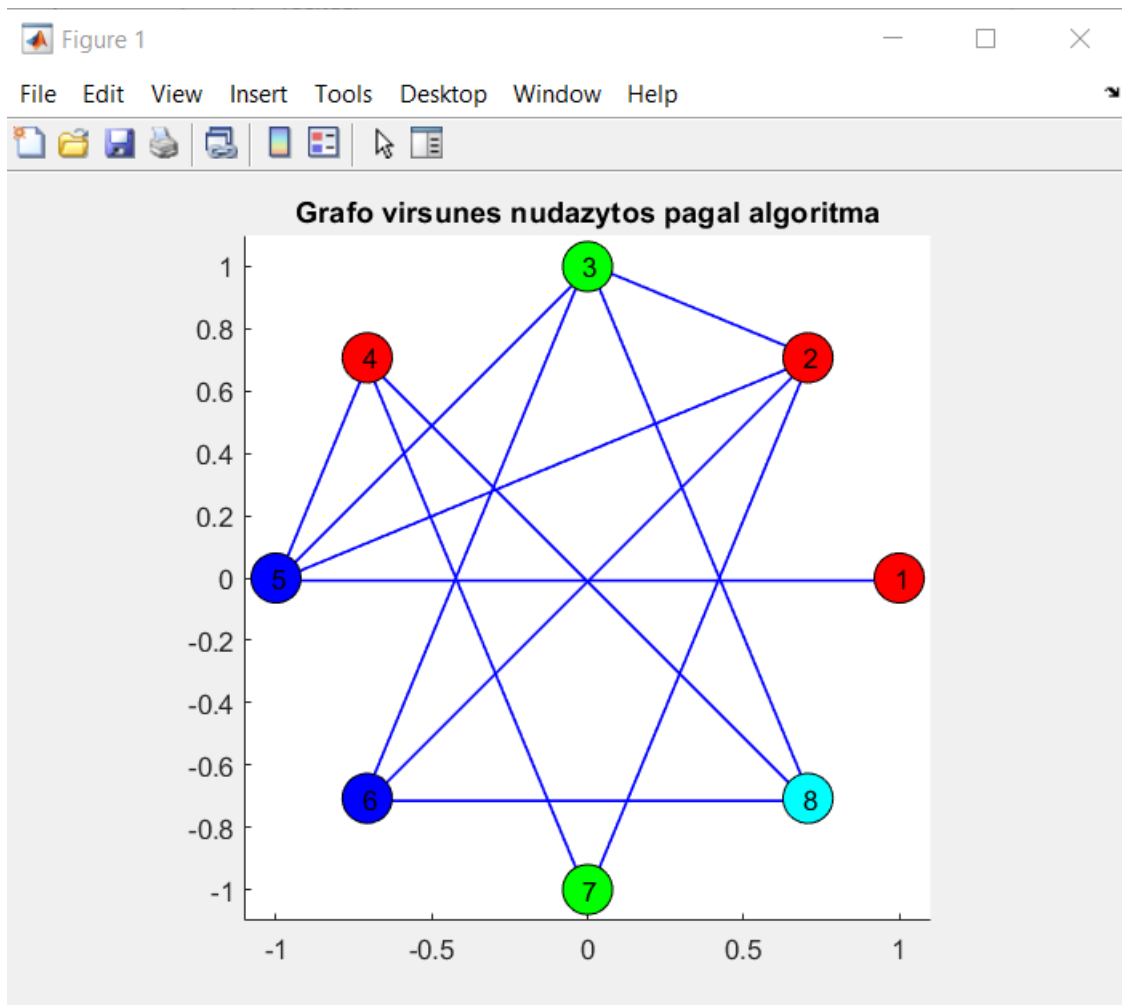
Panaudotu spalvu kiekis(chromatinis skaicius):  
3

### **Antras testas**

Antrojo pavyzdžio grafas pavaizduotas 3 pav., kai briaunų matrica aprašyta:

```
briaunM = [ 2 3 4 5 8 7 4 6 6 5 5 5;
            3 6 7 3 3 2 8 2 8 2 1 4];
```

Šis grafas buvo nudažytas 4 spalvomis, tad chromatinis skaičius yra 4. Programos rezultatas:



**3 pav.** Nuspalvintas grafas, chromatinis skaičius yra 4

Viršune	Viršune buvo nudazyta
1	1
2	1
3	2
4	1
5	3
6	3
7	2
8	4

Panaudotu spalvu kiekis(chromatinis skaičius):  
4

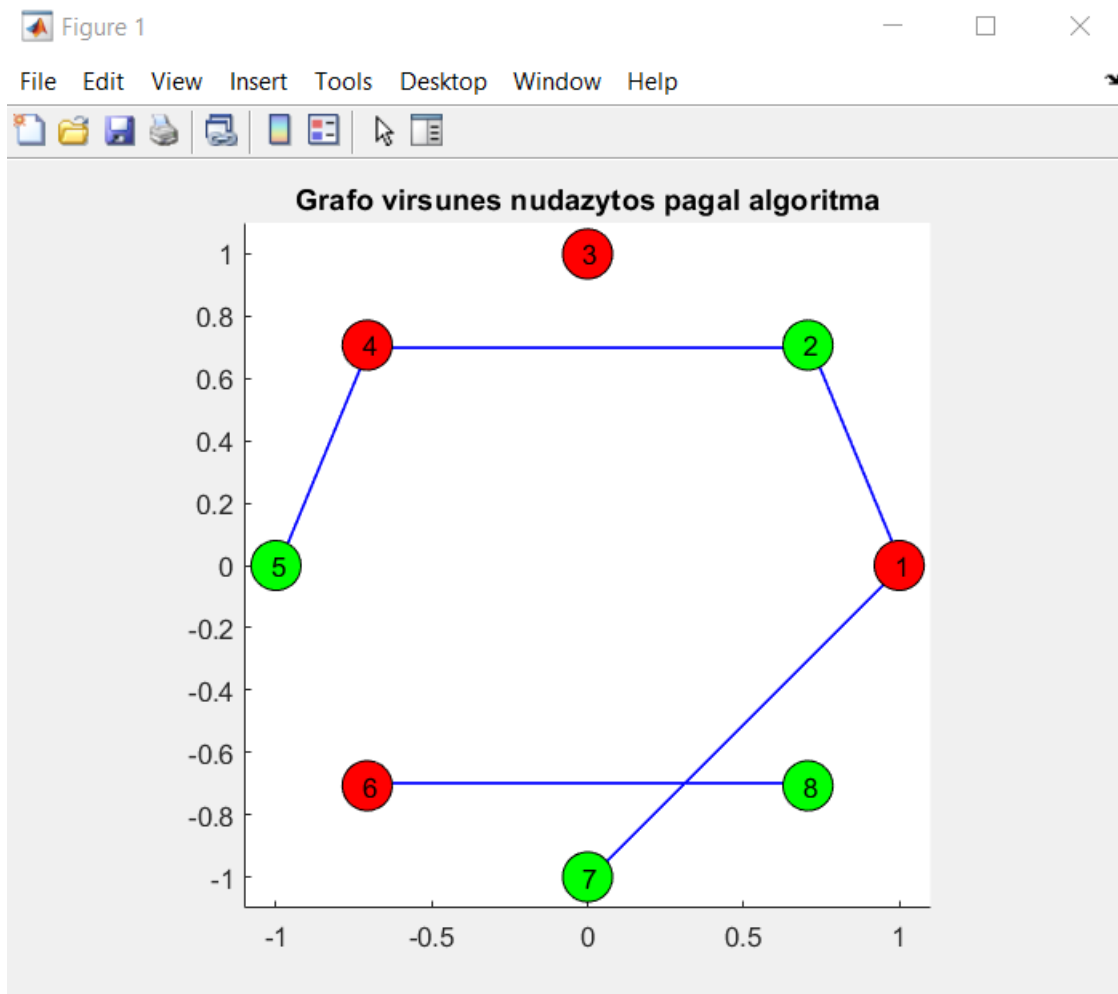
### ***Trečias testas***

Trečiojo pavyzdžio grafas pavaizduotas 4 pav., kai briaunų matrica:

```
briaunM = [1 4 7 8 5;
            2 2 1 6 4];
```

Šis grafas buvo nudažytas 2 spalvomis, tad chromatinis skaičius yra 2. Programos rezultatas:





**4 pav.** Nuspalvintas grafas, chromatinis skaičius yra 2

Viršune	Viršune buvo nudazyta
1	1
2	2
3	1
4	1
5	2
6	1
7	2
8	2

Panaudotu spalvu kiekis(chromatinis skaičius):  
2

## 6. Išvados

Programa veikia teisingai. Nudažomos visos grafo viršūnės laikantis principo „pirma spalva, po to viršūnė“, taip pat randamas ir chromatinis grafo skaičius.

## 7. Literatūros sąrašas

1. Matlab dokumentacija <http://www.mathworks.se/help/index.html> (žiūrėta 2022-11-23)
2. „Diskrečiųjų struktūrų“ modulis „Moodle“ aplinkoje <https://moodle.ad.ktu.lt/course/view.php?id=65> (žiūrėta 2022-11-25)