**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**Diskrečiosios Struktūros**

Kursinio darbo ataskaita

Atliko: Wink Wink IF-X/X

Vadovai: Dėstytojas

**KAUNAS, 2012**

Turinys

[1. Uždavinio sąlyga ir jo analizė 3](#_Toc346493688)

[1.1. Uždavinio sąlyga 3](#_Toc346493689)

[1.2. Uždavinio analizė 3](#_Toc346493690)

[Indukuotas grafas 3](#_Toc346493691)

[Dvipusis grafas 3](#_Toc346493692)

[Jungusis grafas 3](#_Toc346493693)

[Bendra užduotis 4](#_Toc346493694)

[2. Algoritmo aprašymas 4](#_Toc346493695)

[2.1. Sprendimo algoritmas 4](#_Toc346493696)

[3. Programos kodas 6](#_Toc346493697)

[3.1. main.m 6](#_Toc346493698)

[3.2 AtlikimoFunkcija.m: 6](#_Toc346493699)

[3.3 generuotiPoaibius.m: 8](#_Toc346493700)

[4. Testai 8](#_Toc346493701)

[4.1. Pirmas pavyzdys: 8](#_Toc346493702)

[4.2. Antras pavyzdys: 9](#_Toc346493703)

[5. Išvados 10](#_Toc346493704)

# Uždavinio sąlyga ir jo analizė

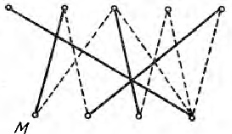
## Uždavinio sąlyga

**Užduotis nr. 22:** Sudaryti algoritmą ir programą, kuri rastų visus indukuotus jungius dvipusius grafus (3,s) (s<n-3) duotame n - viršūniniame grafe

## Uždavinio analizė

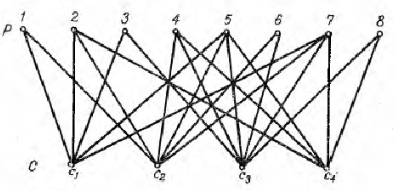
Kursinio darbo užduotis – sukurti programą, kuri sugeneruotų visus indukuotus jungius dvipusius grafus. Užduoties apribojimas – pirmojo poaibio viršūnių kiekis – 3, o antro poaibio bendras viršūnių kiekis – ne didesnis nei (visas viršūnių kiekis – 3).

Indukuotas grafas – tai grafas, turintis NE VISAS pradinio grafo Viršūnes ir NE VISAS jo briaunas. Indukuoto grafo viršūnių aibė turi dalinai sutapti su pradinio grafo viršūnių aibe(ne visos pradinio grafo viršūnės), ir turi turėti visas tas viršūnes jungiančias briaunas.



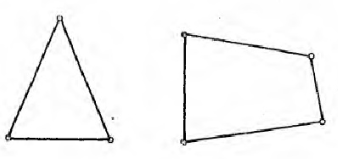
Viršuj - indukuoto pografio pavyzdys.Atkreipti dėmesį, jog punktyrinės linijos – pradinio grafo briaunos, kurių indukuotasis neturi. Tuo pačiu jis neturi ir vienos viršūnės šiam pavyzdyje, kurią pradinis grafas turėjo.

Dvipusis grafas – grafas, kurio viršūnės yra sugrupuotos į du poaibius. Tarpusavy viduj poaibių viršūnės nėra sujungtos jokiomis briaunomis, tačiau tarp abėjų aibių viršūnės yra sujungtos.



Aukščiau – dvipusio grafo pavyzdys. Viršutinės viršūnės priklauso poaibiui Nr. 1, apatinės viršūnės priklauso poaibiui nr. 2. Abejose aibėse viršūnės tarpusavy nesujungtos, tiktais jos sujungtos tarp aibės nr.1 ir aibės nr. 2.

Jungusis grafas – vientisas netrūkus grafas. Nuo vienos viršūnės galima patekti per briaunias iki bet kurios kitos viršūnės.



Viršuj esantis grafas nėra jungus. Jį sudaro 2 pografiai, nesujungti tarpusavyje. Kad grafas būtų jungus, reikia sujungti visus pografius bent 1 briauna.

Bendra užduotis – gauti indukuotą jungų dvidalį pografį. Taip galima bus padaryti atmetus nepageidautinas, dėl kurių grafas negali būti jungus. Tada galima generuoti visus įmanomus pografius, atmetant viršūnes, taip padarant grafą indukuotu. Taip mes gautume dvidalį indukuotą grafą. Ir, galiausiai atmetus briaunas ir viršūnes reikia įsitikinti, jog mūsų pografis yra jungus.

# Algoritmo aprašymas

## Sprendimo algoritmas

Pirmiausiai, mums reikia suskirstyti duotąjį grafą į visas įmanomas aibes. Svarbiausia sąlyga ta, jog pirmojo poaibių elementų kiekis yra 3, antro ne daugiau nei n – 3. Padarome ciklą, kuris suranda visas galimas aibes. Svarbiausia, jog aibėse viršūnės nesikartotų, ir jog jos nesikartotų su ankstesnėmis aibėmis. Pvz. {1 2 3}, {1 2 4}, {2 4 5}, {4 5 6} ir t.t. Suradus visas aibes, mes turim rasti ir likusius poaibius, kuriuoe reikia sudėti viršūnes iš likusių viršūnių, tuo pačiu darant grafą indukuotu. Naudojama rekursinė funkcija, kuriai nurodomas viršūnės aibės dydis. Algoritmas veikia taip, jog generuojamos visos likusios viršūnės antram poaibiui. Kadangi pasitaiko duplikatų, tikrinama, ar vienodi, ir paliekami tik unikalūs, duplikatai išmėtomi. Kai aibės dydis tampa 0, funkcija daugiau nieko negeneruoja, tik grąžina. Taip sulipdžius gauname visus įmanomus indukuoto pografio poaibius. Dabar reikia aibes sujungti galimomis pradinio grafo briaunomis. Sujungę aibes briaunomis, turime įsitikinti, ar tikrai gavom dvidalį grafą. Patikrinam, ar poaibių viduje viršūnės nesijungia. Jei viskas pasitvirtina, gauname indukuotą jungų dvidalį pografį. Grafas bus jungus, nes iš karto atrenkami dvidaliai pografiai, kurie visada yra jungūs. Belieka tik tie pografiai, kurie atitinka mūsų sąlygą.

# Programos kodas

## main.m

clc, close all, clear all ;

V = importdata('Virsunes.txt');

U = {};

T = importdata('Briaunos.txt');

for i=1:length(T),

U{i} = [T(i, 1) T(i, 2)];

end

[rezU] = AtlikimoFunkcija(V, U);

for i=1:length(rezU), %Isvedami musu rezultatai i konsoles langa

AA = rezU{i};

i

for j=1:length(AA),

AA1 = cell2mat(AA(j))

end

end

hold on; axis equal; axis([-1.1,1.1,-1.1,1.1]); grid on

arc=0; poz=0; Fontsize=10; lstor=1; spalva='b';

figure(1)

title('Duotasis grafas')

plotGraphVU1(V,U,0,0,[],1,10,3,'g');

title('Gautas grafas')

plotGraphVU1(V,rezU{1},0,0,[],1,10,3,'r'); %isvedam pati pirma rezultata grafiskai

## AtlikimoFunkcija.m:

function [rezU] = AtlikimoFunkcija( V, U)

rezU = {};

ilgis = length(V);

ilgisCiklui = round(ilgis/2) - mod(ilgis, 2);

KompleksinisKelias = 0;

for i=1:ilgisCiklui,

KompleksinisKelias = KompleksinisKelias + RadimasC(i, ilgis);

end

Poaibiai = {};

%sukuriam poaibius iš 3 elementų

for i=1:ilgis

for j=i+1:ilgis

for k=j+1:ilgis

Poaibiai{length(Poaibiai)+1} = [V(i), V(j), V(k)];

end

end

end

%sukuriam porinius poaibius

visiPoaibiai2 = {};

for i=1:ilgis-4

p = generuotiPoaibius(V, i);

for j=1:length(p)

visiPoaibiai2{length(visiPoaibiai2)+1} = p{j};

end

end

for i=1:length(visiPoaibiai2)

visiPoaibiai2{i} = sort(visiPoaibiai2{i});

end

Poaibiai2 = {};

for i=1:length(visiPoaibiai2)

yra = 0;

for j=1:length(Poaibiai2)

if length(Poaibiai2{j}) == length(visiPoaibiai2{i})

if Poaibiai2{j} == visiPoaibiai2{i}

yra = 1;

end

end

end

if yra == 0

Poaibiai2{length(Poaibiai2)+1} = visiPoaibiai2{i};

end

end

%Sujungiam poaibius su poriniais poaibiais

for i=1:length(Poaibiai)

for j=1:length(Poaibiai2)

tmpU = U;

poaibis = Poaibiai{i};

poaibis2 = Poaibiai2{j};

yraGrafe = 1;

for k=1:length(poaibis)

for l=1:length(poaibis2)

virsune = [poaibis(k), poaibis2(l)];

yraVirsune = 0;

if poaibis(k) ~= poaibis2(l)

for e = 1:length(tmpU)

if (~isempty(find(ismember(poaibis(k), tmpU{e}),1))) && (~isempty((find(ismember(poaibis2(l), tmpU{e}),1))))

yraVirsune = e;

end

end

end

if ~yraVirsune == 0

tmpU{yraVirsune} = [];

else

yraGrafe = 0;

end

end

end

if yraGrafe == 1

for k=1:length(poaibis)

for l=1:length(poaibis)

virsune = [poaibis(k), poaibis(l)];

yraVirsune = 0;

if poaibis(k) ~= poaibis(l)

for e = 1:length(tmpU)

if (~isempty(find(ismember(poaibis(k), tmpU{e}),1))) && (~isempty((find(ismember(poaibis(l), tmpU{e}),1))))

yraVirsune = e;

end

end

end

if ~yraVirsune == 0

tmpU{yraVirsune} = [];

else

yraGrafe = 0;

end

end

end

for k=1:length(poaibis2)

for l=1:length(poaibis2)

virsune = [poaibis2(k), poaibis2(l)];

yraVirsune = 0;

if poaibis2(k) ~= poaibis2(l)

for e = 1:length(tmpU)

if (~isempty(find(ismember(poaibis2(k), tmpU{e}),1))) && (~isempty((find(ismember(poaibis2(l), tmpU{e}),1))))

yraVirsune = e;

end

end

end

if ~yraVirsune == 0

tmpU{yraVirsune} = [];

else

yraGrafe = 0;

end

end

end

if yraGrafe == 0

virsunes = {};

for q = 1:length(poaibis)

for w = 1:length(poaibis2)

virsunes{length(virsunes)+1} = [poaibis(q), poaibis2(w)];

end

end

rezU{length(rezU) + 1} = virsunes;

end

end

end

end

end

## generuotiPoaibius.m:

function [ Poaibiai ] = generuotiPoaibius( V, virsuniuSkaicius )

Poaibiai = {};

if virsuniuSkaicius == 1

for i=1:length(V)

Poaibiai{length(Poaibiai)+1} = V(i);

end

else

for i=1:length(V)

tmpV = V;

tmpV(i) = [];

tmpPoaibiai = generuotiPoaibius(tmpV, virsuniuSkaicius-1);

for j=1:length(tmpPoaibiai)

poaibis = tmpPoaibiai{j};

poaibis(length(poaibis)+1) = V(i);

Poaibiai{length(Poaibiai) + 1} = poaibis;

end

end

end

end

# Testai

Rezultatai išvedami pro komandos langą. Norėdami pavaizduoti juos, reikia main.m faile pakeisti pografio numerį (Pagal nutylėjimą nustatytas yra 1as).

## Pirmas pavyzdys:

Viršūnių aibė:

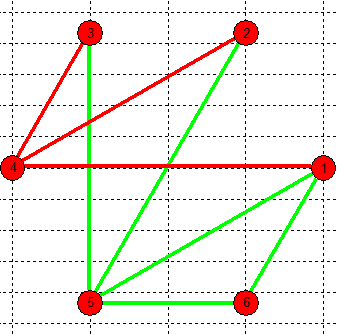
V = [1 2 3 4 5 6];

Briaunų aibė:

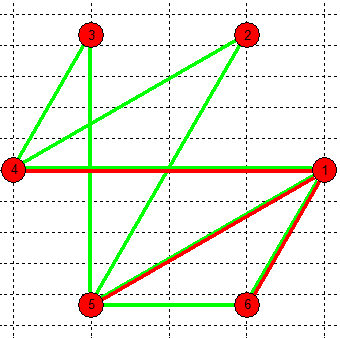
U = {[ 1 4], [1 5], [1 6], [2 4], [2 5], [3 4], [3 5], [5 6]};

Gautas Grafas ir mūsų apskaičiuotas rezultatas (žalios briaunos kartu su raudonom – pradinis grafas, tik raudonos – gautas pografis):

1as pografis:



7as pografis:



## Antras pavyzdys:

Viršūnių aibė:

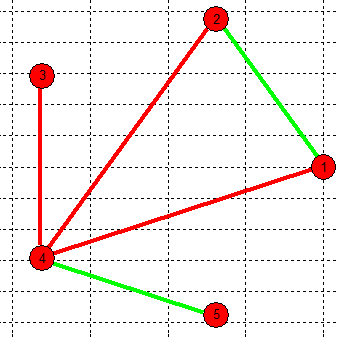
V = [1 2 3 4 5];

Briaunų aibė:

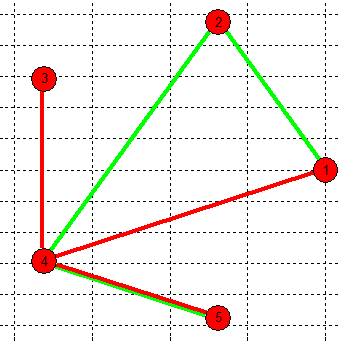
U = {[ 1 2], [1 4], [2 4], [3 4], [4 5]};

Gautas Grafas ir mūsų apskaičiuotas rezultatas (žalios briaunos kartu su raudonom – pradinis grafas, tik raudonos – gautas pografis):

1as pografis:



3as pografis:



# Išvados

Programa veikia. Išanalizavę užduotį, parašėme programą, kuri duoda reikiamus rezultatus.