**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**Taikomosios diskrečiosios matematikos**

Kursinio darbo ataskaita

Studentė: Danil Titkov, IF-9/1

Vadovai: Birutė Jarašiūnienė

**KAUNAS, 2010**

Turinys

[1. Uždavinio sąlyga ir jo analizė 3](#_Toc280099649)

[1.1. Uždavinio sąlyga 3](#_Toc280099650)

[1.2. Uždavinio analizė 3](#_Toc280099651)

[2. Algoritmo aprašymas, iliustruotas pavyzdžiais 3](#_Toc280099652)

[2.1. Sprendimo algoritmas 3](#_Toc280099653)

[3. Programos kodas 3](#_Toc280099654)

[3.1. pagr.m 3](#_Toc280099655)

[3.2. medisPaieskaGilyn.m 4](#_Toc280099656)

[4. Testai 7](#_Toc280099657)

[4.1. Pirmas pavyzdys: 7](#_Toc280099658)

[4.2. Antras pavyzdys: 8](#_Toc280099659)

# Uždavinio sąlyga ir jo analizė

## Uždavinio sąlyga

31. Sudaryti algoritmą ir programą randančią l-viršūninius grafo G indukuotus

pografius, turinčius Oilerio ciklą.

## Uždavinio analizė

Grafas G‘ = (V‘,E‘) vadinamas G = (V,E) pografiu, jeigu V‘ priklauso V ir E‘ priklauso E. Jeigu pografio G‘ briaunų aibėje E0‘ yra visos E briaunos, jungiančios V‘ viršūnes, tai G‘ vadinamas V‘ indukuotoju pografiu.

Oilerio maršrutas - maršrutas grafe, kuriuo galima pereiti visas jo briaunas po vieną kartą. Jei tokiu maršrutu grįžtama į pradinę viršūnę, jis vadinamas Oilerio ciklu, jei ne - Oilerio keliu arba Oilerio grandine. Oileris įrodė, kad Oilerio maršrutas neorientuotame grafe egzistuoja tada ir tik tada, kai pastarasis yra jungusis ir turi ne daugiau nei dvi nelyginio laipsnio viršūnes. Be to, jei tokių viršūnių nėra, egzistuoja Oilerio ciklas, o jei yra dvi - Oilerio kelias.

# Algoritmo aprašymas, iliustruotas pavyzdžiais

## Sprendimo algoritmas

Rekursiškai ieškoma visų galimų pografių ir tikrinama ar jie atitinka Oilerio ciklo salygą, tai yra, ar visų jo viršunių laipsniai yra lyginai. Jei salyga tenkinama pografis pavaizduojamas grafiškai ir įrašomas į pografiu matricą. Du kartus tas pats pografis nevaizduojamas ir neįrašinėjamas i pografių matricą.

# Programos kodas

## pagr.m

clc, close all, clear all ;

% Grafo virsuniu zymejimas ekrane

% % Lango mastelis

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

%

%Grafo virsuniu zymejimas

Vkor=[];

nn = 0;

% Ciklas virsunems zymeti

disp('kairys mygtukas zymi taskus nuo 1 iki (n-1)')

disp('desinys - paskutini taska n')

but = 1;

while but == 1

[xi,yi,but] = ginput(1);

plot(xi,yi,'ro')

nn = nn+1;

Vkor(nn,1)=xi;

Vkor(nn,2)=yi;

end

V = [1 2 3 4 5 6];

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

%breziamas pradinis grafas

figure(1)

title('Duotasis grafas')

plotGraphVU1(V,U,0,0,Vkor,0,10,2,'b')

pografiai = {};

[sk, pografiai] = oilerCiklai(V, U, Vkor, 2, pografiai);

## moilerCiklai.m

function [sk, pografiai] = oilerCiklai( V, U, Vkor, num, pografiai )

%V - grafo virsuniu aibe;

%U - grafo briaunu matrica;

%num - ieskomo ciklo numeris(reikalinga grafiko piesimui);

%sk - grazinamas nupiestu ciklu numeris(reikalina grafiko piesimui);

%pografiai - indukuotuju pografiu aibe, atitinkanciu Oilerio ciklo salyga

%Algoritmas

%rekursiskai ieskoma visu galimu pografiu ir tikrinama ar jie atitinka

%Oilerio ciklo salyga, tai yra, ar visu ju virsuniu laipsniai yra lyginai.

%jei salyga tenkinama pografis pavaizduojamas grafiskai ir irasomas i

%pografiu matrica. du kartus tas pats pografis nevaizduojamas ir

%neirasinejamas i pografiu matrica.

if length(V) > 2 && length(U) > 2 %jei i funkcija perduotame grafe/pografyje dar yra pakankamai virsuniu ciklui

for i=1:length(V) %tikrinami atskiri virsuniu atvejai

tempV = V; %padaroma laikina virsuniu aibe(reikalinga tik sitame cikle)

tempV(i) = []; %isvaloma tam tikra virsune

tempU = {}; %sukuriama laikina briaunu aibe

for j=1:length(U) %atreankamos tik tempV virsuniu aibei priklausancios briaunos

if U{j}(1) ~= V(i) && U{j}(2) ~= V(i)

tempU{length(tempU)+1} = U{j};

end

end

nums = []; %virsuniu laipsniu sarasas

for j=1:length(tempV) %visoms virsunems

nums(j) = 0;

for k=1:length(tempU) %suskaiciuojami laipsniai

if tempU{k}(1) == tempV(j) || tempU{k}(2) == tempV(j)

nums(j) = nums(j)+1;

end

end

end

ok = 1; %kintamasis reikalingas salygu tikrinimam, jie visos salygos geros tada jis ir liks 1

for j=1:length(nums) %tikrinama ar visos virsunes lyginio laipsnio

if mod(nums(j),2) ~= 0 || nums(j) == 0 %lyginiu virsuniu salygos tikrinimas

ok = 0;

break; %ciklas nutraukiamas, kad butu sutaupoma laiko

end

end

if ok == 1 %jie virsuniu laipsniai atitinka salyga, tikrinama ar sis ciklas nebuvo irasytas anksciau

for j=1:length(pografiai)

if isequalwithequalnans(pografiai{j}, tempV) == 1 %jei ciklas jau buvo irasytas

ok = 0;

break;

end

end

end

if ok == 1 %jei rastas indukuotasis pografis yra naujas

pografiai{length(pografiai)+1} = tempV;

tempVkor = []; %laikina konkretaus pografio virsuniu koordinaciu aibe

for j=1:length(tempV) %atrenkamos koordinates pografio pavaizdavimui

tempVkor(j,1) = Vkor(tempV(j), 1);

tempVkor(j,2) = Vkor(tempV(j), 2);

end

figure(num) %figuros numeris

grid on

title('Rastas Oilerio ciklas pografyje')

plotGraphVU1(tempV,tempU,0,0,tempVkor,0,10,2,'r') %pografio brezimas

num = num+1;

end

[num, pografiai] = oilerCiklai(tempV, tempU, Vkor, num, pografiai ); %rekursiskai kreipiamasi toliasnem pografiu paieskom

end

end

sk = num; %rastu ciklu skaicius

return

# Testai

## Pirmas pavyzdys:

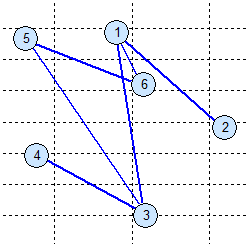
Viršūnių aibė:

V = [1 2 3 4 5 6];

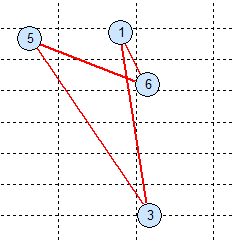
Briaunų aibė:

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

Grafo vaizdas:



Gauti indukuotieji pografiai,turintys Oilerio kelia:



## Antras pavyzdys:

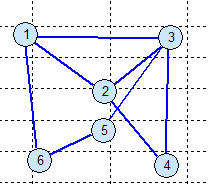
Viršūnių aibė:

V = [1 2 3 4 5 6];

Briaunų aibė:

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

Grafo vaizdas:



Gauti indukuotieji pografiai,turintys Oilerio kelia:

