**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

INFORMATIKOS FAKULTETAS

TAIKOMOSIOS INFORMATIKOS KATEDRA

**DISKREčIOSIOS struktūros (p170b008)**

**KURSINIS DARBAS**

**Užduoties Nr. B03**

Atliko: IF -5/3 gr. stud. *Ugnius Mockus*

Priėmė: dėst. *Audrius Nečiūnas*

KAUNAS

2016

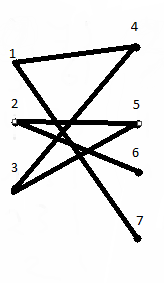
# 

# 1. Užduotis (nr. B3)

Sudaryti algoritmą ir programą, kuri nustatytų, ar grafas yra dvidalis, ir jei taip, tai išskirtų abi grafo puses.

# 2. Užduoties analizė

Dvidalis grafas, tai toks grafas, kurio viršūnių aibę galima išskaidyti į du poaibius A ir B taip, kad kiekvienos briaunos galai priklausytų skirtingiems poaibiams. Grafas yra dvidalis, jei įmanoma visas jo viršūnes nudažyti dviem spalvomis, kai vienodos spalvos nėra gretimos.



1 pav. Dvidalio grafo pavyzdys

# 3. Programos tekstas

package projektas;

import java.util.ArrayList;

import java.util.List;

import java.util.Stack;

public class Projektas {

private List<Integer> kaire=new ArrayList<Integer>();

private List<Integer> desine=new ArrayList<Integer>();

static int[][] G=new int[][]{

{0,0,0,1,0,0,1},

{0,0,0,0,1,1,0},

{0,0,0,1,1,0,0},

{1,0,1,0,0,0,0},

{0,1,1,0,0,0,0},

{0,1,0,0,0,0,0},

{1,0,0,0,0,0,0},

};

public static void main(String[] args) {

Projektas temp=new Projektas();

temp.Ieskoti();

}

public void Ieskoti(){

int ilgis = G.length;

int[][] A=new int[ilgis][ilgis];

int nr = 0;

for(int i = 0; i < ilgis; i++)

{

for(int j = 0; j < ilgis; j++)

{

if(G[i][j] > 0)

{

A[i][nr] = j + 1;

nr++;

}

}

nr = 0;

}

for(int i=0;i<A.length;i++)

{

for(int j=0;j<A.length;j++)

{ //su ju briaunomis

if(A[i][j]!=0)

{

int current = A[i][j];

int prev = i + 1;

if(!kaire.contains(prev))

{

if(!kaire.contains(current))

{

kaire.add(i + 1);

}

}

if(!desine.contains(prev))

{

if(!desine.contains(current))

{

desine.add(A[i][j]);

}

}

}

}

}

boolean dvidalis = true;

for(int i = 0; i < desine.size(); i++)

{

for(int j = 0; j < kaire.size(); j++)

{

if(desine.contains(kaire.get(j)))

{

dvidalis = false;

break;

}

}

}

if(!dvidalis)

{

System.out.print("Grafas ne dvidalis!");

}

else

{

System.out.println("Kaire puse ");

for(int i=0;i<kaire.size();i++){

System.out.println(kaire.get(i)+" ");

}

System.out.println("Desine puse ");

for(int i=0;i<desine.size();i++){

System.out.println(desine.get(i)+" ");

}

}

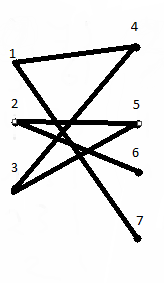
}

}

# 4. Testiniai pavyzdžiai

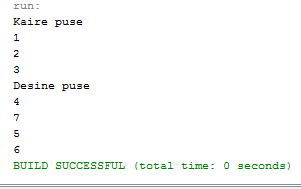
Buvo panaudoti du testiniai pavyzdžiai.

Pirmojo pavyzdžio grafas pavaizduotas 2 pav.

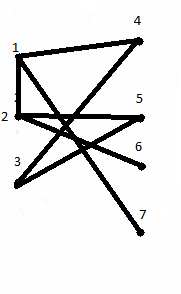


2 pav. Dvidalio grafo pavyzdys

Šis grafas gali būti nudažytas dviem spalvomis, todėl jis nėra dvidalis. Programos rezultatas:

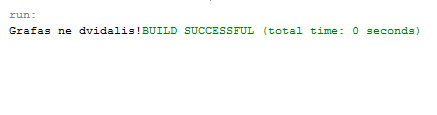


Antrojo pavyzdžio grafas pavaizduotas 3 pav.



3 pav. Nedvidalio grafo pavyzdys

Šio grafo pirma ir antra viršūnės turi briauną todėl jis nėra dvidalis. Programos rezultatas:



# 5. Išvados

Programa veikia teisingai, tol, kol grafo viršūnės esančios toje pačioje dalyje viena nuo kitos skiriasi tik 1 .

# 6. Literatūros sąrašas

1. „Diskrečiųjų struktūrų“ modulis „Moodle“ aplinkoje https://moodle.ktu.edu/course/view.php?id=763 (žiūrėta 2016-12-15)