

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
TAIKOMOSIOS INFORMATIKOS KATEDRA

DISKREČIOSIOS STRUKTŪROS (P170B008)
KURSINIS DARBAS
Užduoties Nr. B16

Atliko: IFF 8-2 gr. studentas *Nedas Šimoliūnas*

Priėmė: dėst. *Martynas Patašius*

KAUNAS

2019

1. Užduotis (nr. 16)

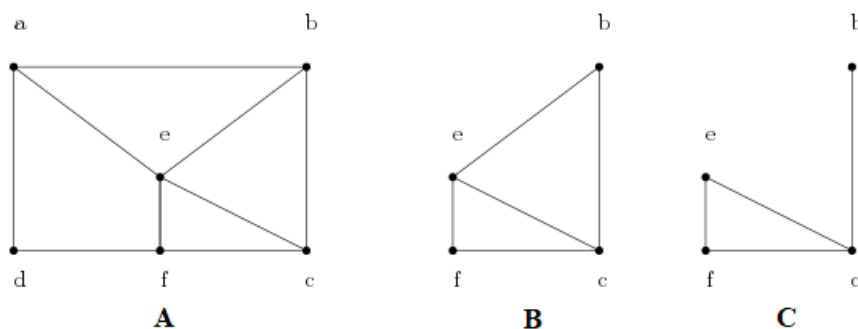
Duotas grafas G ir du jo pografiai G_1 ir G_2 . Sudaryti programą, kuri nustatytų, ar šie pografiai yra indukuoti, šių pografijų bendrų briaunų skaičių.

2. Užduoties analizė

Grafo A pografis B bus indukuotasis, jei jo viršūnių aibė yra grafo A viršūnių aibės poaibis ir pografio B briaunų aibę sudaro tos grafo A briaunos, kurių abu galai priklauso pografio B viršūnių aibei.

Programa gauna tris grafus (pagrindinį ir du pografius), aprašytus briaunų matrica (viena briaunos viršūnė yra viename masyve, kita viršūnė – kitame, masyvų ilgiai vienodi, vienas indeksas per abu masyvus nurodo vieną briauną).

Iš pografio viršūnių aibės padaromos visos galimos grafo G viršūnių kombinacijos. Gavus viršūnių transformaciją, pakeičiamos duotos pografio briaunos. Tikrinama, ar kiekviena briauna priklauso grafiui G . Jei bent viena nepatenka, pografis negalimas ir einama prie kitos kombinacijos. Jei visos briaunos patenka, tikrinama, ar pografis yra indukuotasis. Tai padaroma suskaičiuojant briaunų, kurių viršūnės patenka į kombinacijos aibę, kiekį. Jei šis skaičius lygus pografio briaunų skaičiui, duotasis pografis yra indukuotasis. Masyve *duplicates*, imant pografio viršūnės numerį kaip indeksą, padidinama reikšmė. Įvykdomi tie patys veiksmai su kitu pografium. Tikrinamas masyvas *duplicates*. Kiekviena grafo G briauna, kurios viršūnių reikšmės masyve *duplicates* bus lygios dviem, priklausys tiek pografiui G_1 , tiek G_2 . Visos grafo G briaunos nuspaltintos juodai, tik G_1 briaunos – raudonos, tik G_2 briaunos – žalios, bendros G_1 ir G_2 briaunos – mėlynos. Pografis spalvinamas tik tada, kai jis yra indukuotasis.



1 pav. Grafas A , indukuotasis pografis B ir pografis C

Su 1 pav. pateiktu pavyzdžiu, tiek B , tiek C yra grafo A pografiai, tačiau tik B yra indukuotasis, nes C grafe trūksta briaunos $(e;b)$.

3. Programos tekstas

Programa parašyta Java programavimo kalba NetBeans aplinkoje.

```
public class Uzduotis16 extends Application {
```

```
    public int[] Ga = {1,2,3,4,2,3,7,6,6,4};
```

```
    public int[] Gb = {2,3,4,5,8,8,8,7,8,6};
```

```
    public int[] G1a = {1,2,1,4,3};
```

```
    public int[] G1b = {2,4,5,5,4};
```

```
    public int[] G2a = {1,1,2,3};
```

```
    public int[] G2b = {2,4,4,4};
```

```
    private Canvas canvas = null;
```

```
    private GraphicsContext gc = null;
```

```
    public int canvasH = 800;
```

```
    public int canvasW = 800;
```

```
    public Color graphColor1 = Color.BLACK;
```

```
    public Color graphColor2 = Color.RED;
```

```
    public Color graphColor3 = Color.GREEN;
```

```
    public Color duplicateColor = Color.BLUE;
```

```
    public int[] duplicates;
```

```
    public boolean t = true;
```

```
    void startProgram() {
```

```

duplicates = new int[maxTwo(Ga, Gb)+1];

for(int i = 0; i < duplicates.length; i++) {
    duplicates[i] = 0;
}

drawGraph(Ga, Gb, graphColor1);

if(Ga.length >= G1a.length)
    checkGraph(G1a, G1b, graphColor2);

t = true;

if(Ga.length >= G2a.length)
    checkGraph(G2a, G2b, graphColor3);

t = true;

int count = 0;

for(int i = 1; i < duplicates.length; i++) {
    for(int j = i+1; j < duplicates.length; j++) {
        if(i != j && duplicates[i] == 2 && duplicates[j] == 2) {

            for(int k = 0; k < Ga.length; k++) {
                if((Ga[k] == i && Gb[k] == j) ||
                    (Ga[k] == j && Gb[k] == i)) {
                    drawEdge(i, j, duplicateColor, maxTwo(Ga, Gb), 5);

                    if(t) {

```

```

        t = !t;

        System.out.println("Indukuotųjų pagrafių bendros briaunos:");

    }

    System.out.print("(" + i + ";" + j + ") ");

    count++;

    }

    }

    }

    }

    }

    if(count > 0) {

        System.out.println("");

        System.out.println("Indukuotųjų pagrafių bendrų briaunų skaičius: " + count);

    }

}

```

```

void checkGraph(int[] n1, int[] n2, Color color) {

    int[] temp = new int[maxTwo(n1,n2)];

    int[] end = new int[maxTwo(n1,n2)];

    for(int i = 0; i < maxTwo(n1, n2); i++) {

        temp[i] = i + 1;

        end[i] = maxTwo(Ga,Gb)-maxTwo(n1,n2)+1+i;

    }

    do {

        generate(temp.length, temp.clone(), color, n1, n2);

        for(int j = maxTwo(n1, n2) - 1; j >= 0; j--) {

```

```

        if(temp[j] <= maxTwo(Ga, Gb) - (maxTwo(n1, n2) - j)) {

            temp[j]++;

            for(int k = j+1; k < temp.length; k++) {

                temp[k] = temp[k-1] + 1;

            }

            break;

        }

    }

    if(!t) {

        break;

    }

} while(!equalArray(temp, end));

```

```

generate(end.length, end.clone(), color, n1, n2);

for(int j = maxTwo(n1, n2) - 1; j >= 0; j--) {

    if(temp[j] <= maxTwo(Ga, Gb) - (maxTwo(n1, n2) - j)) {

        temp[j]++;

        for(int k = j+1; k < temp.length; k++) {

            temp[k] = temp[k-1] + 1;

        }

        break;

    }

}

}

```

```

void check(int[] arr, Color color, int[] n1, int[] n2) {

    int[] g1 = n1.clone();

    int[] g2 = n2.clone();

```

```

int count = 0;

change(arr, g1, g2, n1, n2);

for(int i = 0; i < g1.length; i++) {
    for(int j = 0; j < Ga.length; j++) {
        if((g1[i] == Ga[j] && g2[i] == Gb[j]) ||
            (g1[i] == Gb[j] && g2[i] == Ga[j]))
            count++;
    }
}

if(count == g1.length && count == edges(arr)) {
    for(int i = 0; i < g1.length; i++) {
        drawEdge(g1[i], g2[i], color, maxTwo(Ga, Gb), 5);
    }

    if(color.equals(graphColor2))
        System.out.println("G1 yra indukuotasis pografis, jo viršūnės grafe G:");
    else if(color.equals(graphColor3))
        System.out.println("G2 yra indukuotasis pografis, jo viršūnės grafe G:");

    printArray(arr);
    for(int i = 0; i < arr.length; i++) {
        duplicates[arr[i]]++;
    }
    t = false;
}

}

void drawGraph(int[] n1, int[] n2, Color color) {

```

```

for(int i = 0; i < maxTwo(n1, n2); i++) {

    drawNode(canvasW / 2 + Math.cos(2 * Math.PI / maxTwo(n1, n2) * i) * canvasW / 3,
             canvasH / 2 + Math.sin(2 * Math.PI / maxTwo(n1, n2) * i) * canvasH / 3,
             20, color, i + 1);

    gc.setFill(Color.WHITE);

    gc.setTextAlign(TextAlignment.CENTER);

    gc.fillText(String.valueOf(i + 1),
                canvasW / 2 + Math.cos(2 * Math.PI / maxTwo(n1, n2) * i) * canvasW / 3 * 1.2,
                canvasH / 2 + Math.sin(2 * Math.PI / maxTwo(n1, n2) * i) * canvasH / 3 * 1.2);

}

for(int i = 0; i < n1.length; i++) {

    drawEdge(n1[i], n2[i], color, maxTwo(n1, n2), 3);

}

}

void drawNode(double x, double y, double r, Color color, int num) {

    gc.setFill(color);

    gc.fillOval(x - r, y - r, 2*r, 2*r);

}

void drawEdge(int node1, int node2, Color color, int n, int width) {

    gc.setStroke(color);

    gc.setLineWidth(width);

    gc.strokeLine(canvasW / 2 + Math.cos(2 * Math.PI / n * (node1 - 1)) * canvasW / 3,
                  canvasH / 2 + Math.sin(2 * Math.PI / n * (node1 - 1)) * canvasH / 3,
                  canvasW / 2 + Math.cos(2 * Math.PI / n * (node2 - 1)) * canvasW / 3,
                  canvasH / 2 + Math.sin(2 * Math.PI / n * (node2 - 1)) * canvasH / 3);

}

```



```

int edges(int[] arr) {
    int edges = 0;

    for(int i = 0; i < Ga.length; i++) {
        if(contains(arr, Ga[i]) && contains(arr, Gb[i]))
            edges++;
    }

    return edges;
}

boolean contains(int[] arr, int num) {
    for(int i = 0; i < arr.length; i++) {
        if(arr[i] == num)
            return true;
    }
    return false;
}

void change(int[] arr, int[] g1, int[] g2, int[] n1, int[] n2) {
    for(int i = 0; i < g1.length; i++) {
        g1[i] = arr[n1[i]-1];
        g2[i] = arr[n2[i]-1];
    }
}

void generate(int k, int[] arr, Color color, int[] n1, int[] n2) {
    if(!t)
        return;

```

```

if(k == 1) {
    check(arr, color, n1, n2);
}

else {
    generate(k-1, arr, color, n1, n2);

    for(int i = 0; i < k - 1; i++) {
        if(k % 2 == 0) {
            int temp = arr[i];
            arr[i] = arr[k-1];
            arr[k-1] = temp;
        }
        else {
            int temp = arr[0];
            arr[0] = arr[k-1];
            arr[k-1] = temp;
        }

        generate(k-1, arr, color, n1, n2);
    }
}

}

boolean equalArray(int[] arr1, int[] arr2) {
    if(arr1.length != arr2.length)
        return false;

    for(int i = 0; i < arr1.length; i++)

```

```

        if(arr1[i] != arr2[i])
            return false;

    return true;
}

void printArray(int[] mas) {
    for (int i = 0; i < mas.length; i++) {
        System.out.print(mas[i] + " ");
    }
    System.out.println("");
    System.out.println("");
}

int maxTwo(int[] a1, int[] a2) {
    int toReturn = 0;
    for(int i = 0; i < a1.length; i++)
        if(a1[i] > toReturn || a2[i] > toReturn)
            toReturn = Math.max(a1[i], a2[i]);

    return toReturn;
}

public static void main(String[] args) {
    launch(args);
}

@Override
public void start(Stage primaryStage) throws Exception {
    primaryStage.setTitle("Nr. 16");
}

```

```

Group root = new Group();

canvas = new Canvas(canvasW, canvasH);

gc = canvas.getGraphicsContext2D();

gc.setFill(Color.GRAY.darker());

gc.fillRect(0, 0, canvasW, canvasH);

startProgram();

root.getChildren().add(canvas);

primaryStage.setScene(new Scene(root));

primaryStage.show();

}

}

```

4. Testiniai pavyzdžiai

Buvo panaudoti keturi testiniai pavyzdžiai.

Pirmasis testas: abu pografiai yra indukuoti, turi bendrą briauną.

```

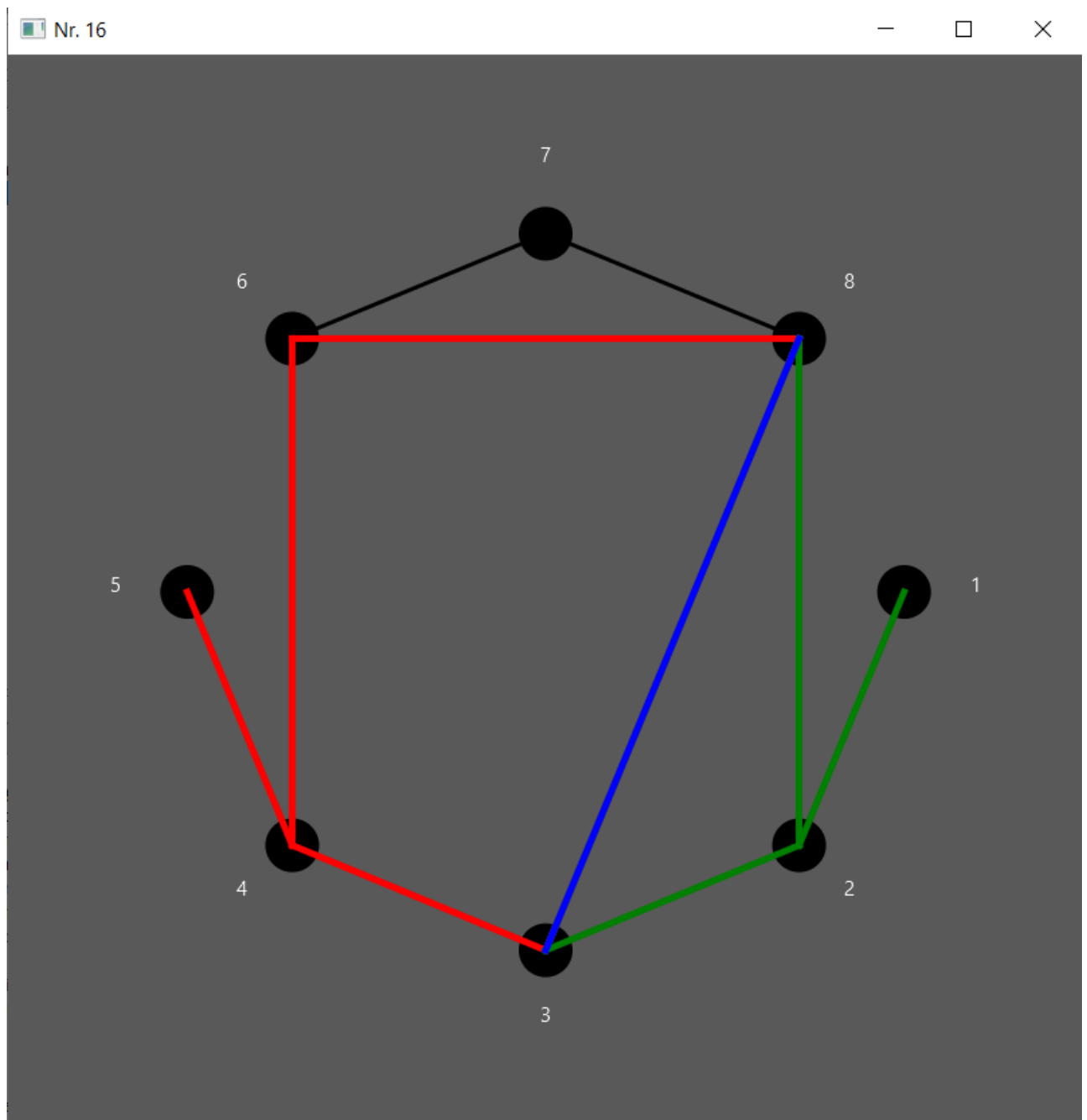
public int[] Ga = {1,2,3,4,2,3,7,6,6,4};
public int[] Gb = {2,3,4,5,8,8,8,7,8,6};

public int[] G1a = {1,2,1,4,3};
public int[] G1b = {2,4,5,5,4};

public int[] G2a = {1,1,2,3};
public int[] G2b = {2,4,4,4};

```

2 pav. Pirmojo testinio varianto aprašyti grafai



3 pav. Pirmojo testinio varianto grafas

G1 yra indukuotasis pografis, jo viršūnės grafe G:
8 3 5 4 6

G2 yra indukuotasis pografis, jo viršūnės grafe G:
3 8 1 2

Indukuotųjų pagrafių bendros briaunos:
(3;8)

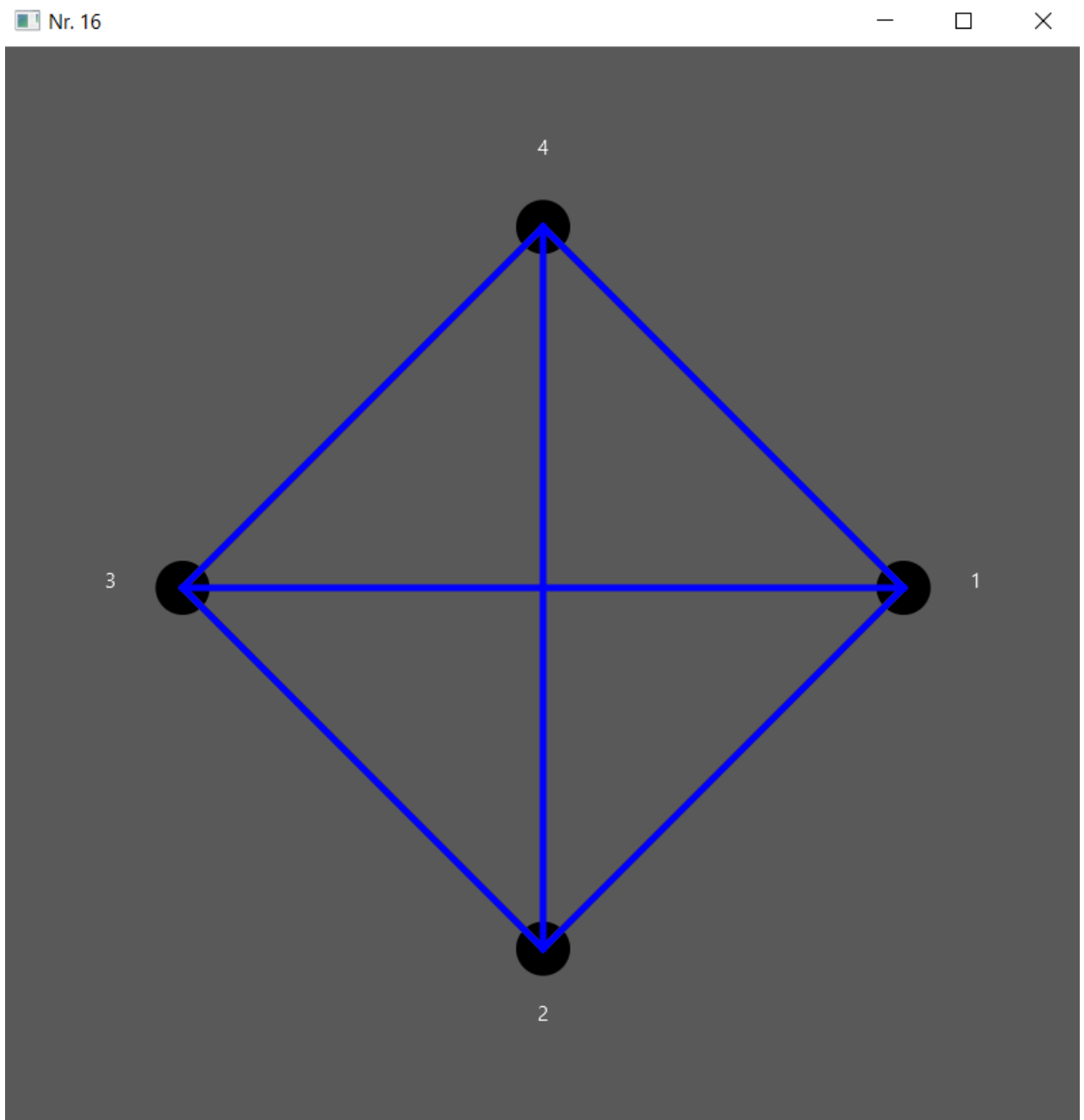
Indukuotųjų pagrafių bendrų briaunų skaičius: 1

4 pav. Pirmojo testinio varianto atsakymų išvedimas

Antras testas: abu pografių lygūs pagrindiniam grafiui:

```
public int[] Ga = {1,2,3,1};  
public int[] Gb = {2,3,4,4};  
  
public int[] Gla = {1,2,3,1};  
public int[] G1b = {2,3,4,4};  
  
public int[] G2a = {1,2,3,1};  
public int[] G2b = {2,3,4,4};
```

5 pav. Antrojo testinio variant aprašyti grafai



6 pav. Antrojo testinio varianto grafas

G1 yra indukuotasis pografis, jo viršūnės grafe G:
1 2 3 4

G2 yra indukuotasis pografis, jo viršūnės grafe G:
1 2 3 4

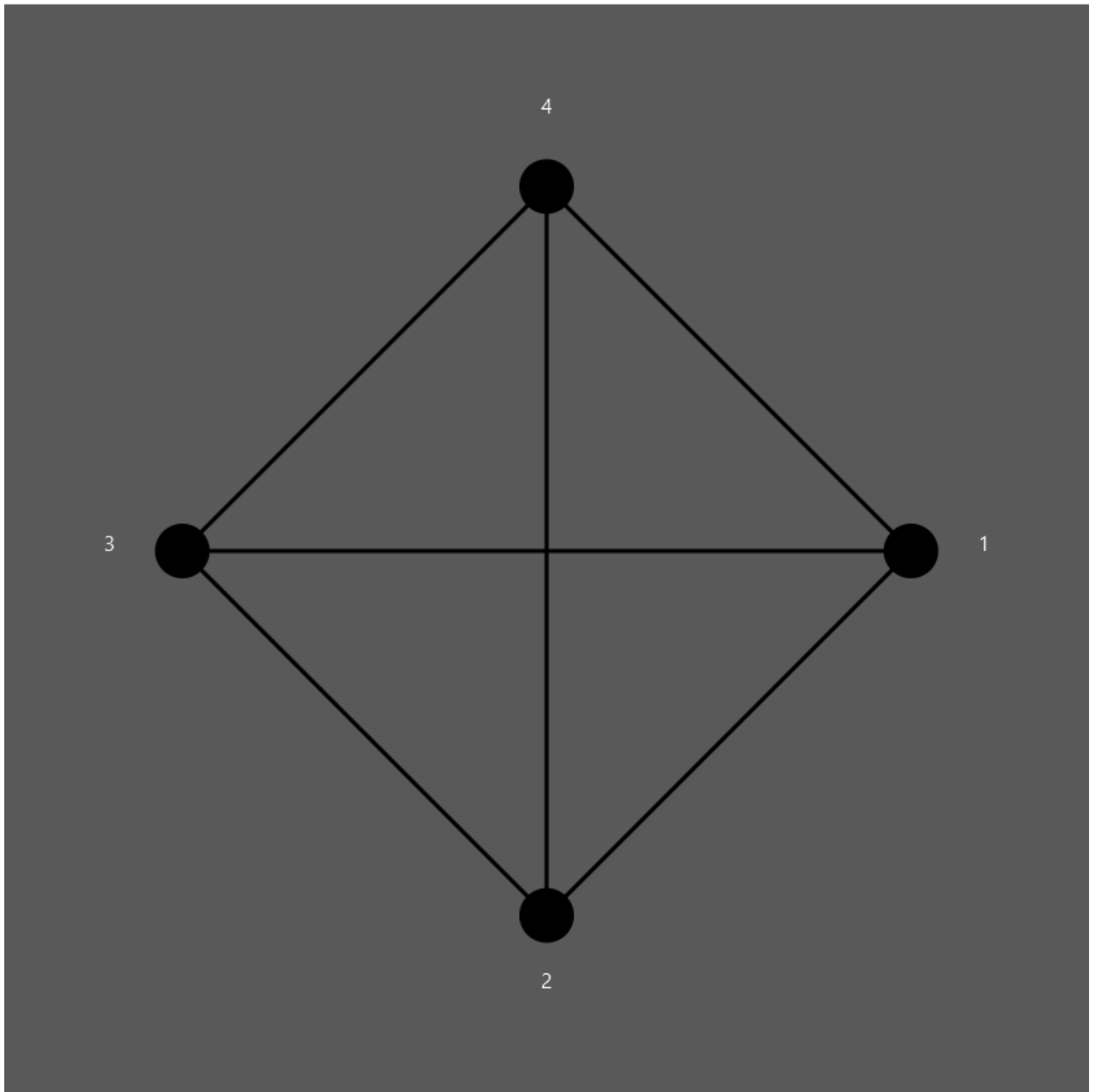
Indukuotųjų pografių bendros briaunos:
(1;2) (1;4) (2;3) (3;4)
Indukuotųjų pografių bendrų briaunų skaičius: 4

7 pav. Antrojo testinio varianto atsakymų išvedimas

Trečiasis testinis variantas: abu pografiai nėra indukuoti.

```
public int[] Ga = {1,2,3,1,1,2};  
public int[] Gb = {2,3,4,4,3,4};  
  
public int[] Gla = {1,2,3,1};  
public int[] Glb = {2,3,4,4};  
  
public int[] G2a = {1,2};  
public int[] G2b = {3,3};
```

8 pav. Trečiojo testinio varianto aprašyti grafai



9 pav. Trečiojo testinio varianto grafas

Programa nerado indukuotųjų pograpių, tad jokių atsakymų neišveda.

Ketvirtasis testinis variantas: du indukuotieji pografiai, neturi bendrų briaunų.


```

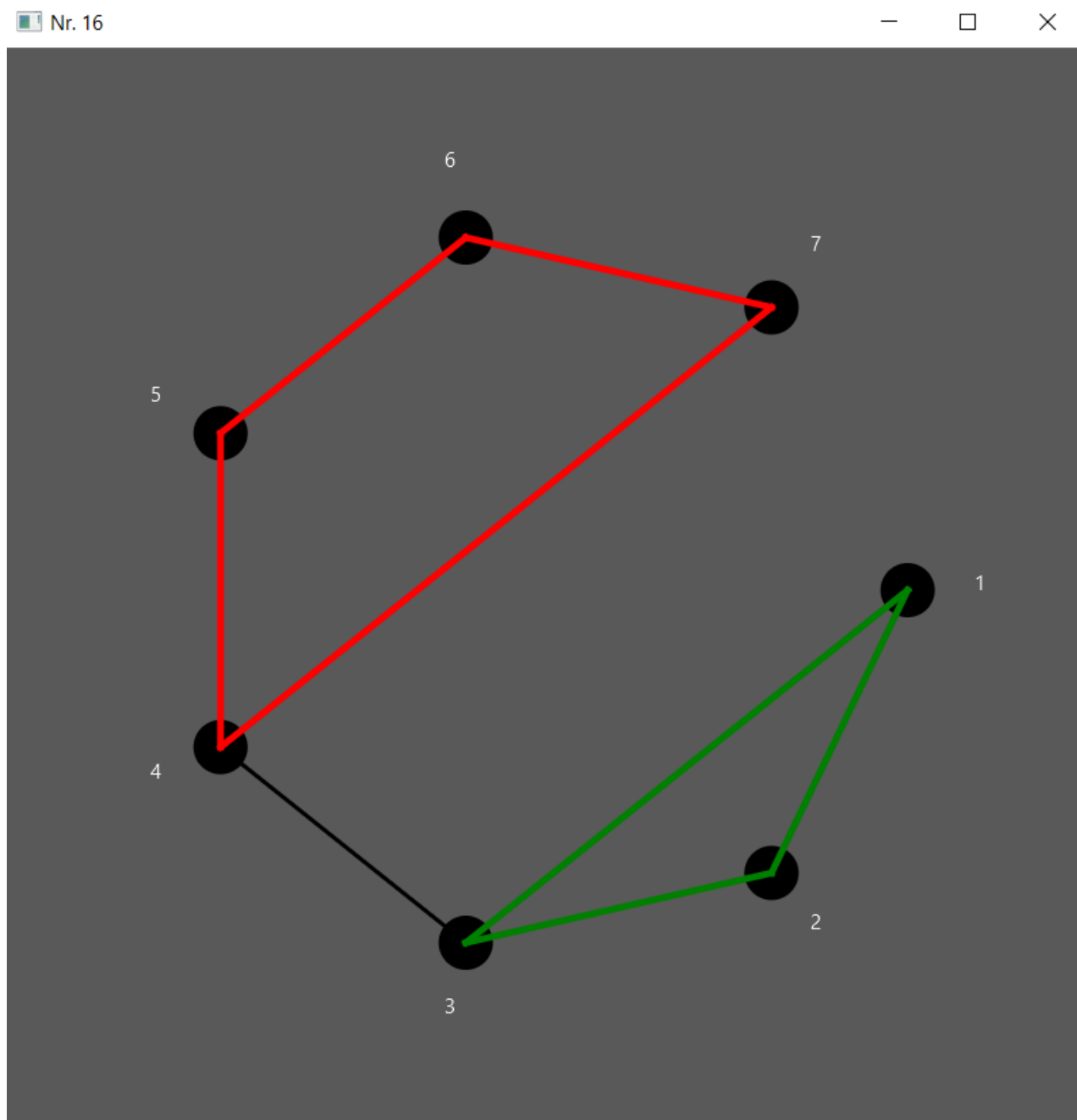
public int[] Ga = {1,2,1,3,4,5,6,4};
public int[] Gb = {2,3,3,4,5,6,7,7};

public int[] Gla = {1,2,3,1};
public int[] Glb = {2,3,4,4};

public int[] G2a = {1,2,1};
public int[] G2b = {3,3,2};

```

10 pav. Ketvirtojo testinio varianto aprašyti grafai



11 pav. Ketvirtojo testinio varianto grafas

G1 yra indukuotasis pografis, jo viršūnės grafe G:
4 5 6 7

G2 yra indukuotasis pografis, jo viršūnės grafe G:
1 2 3

12 pav. Ketvirtąjo testinio varianto atsakymų išvedimas

5. Išvados

Sprendžiant iš pateiktų testinių pavyzdžių, programa veikia teisingai. Randami indukuotieji pografiai, parašomos jų viršūnės, randamos bendros indukuotųjų pograpių briaunos. Grafinėje sąsajoje nubrėžiami grafai, atskirti spalvomis.

6. Literatūros sąrašas

1. „Diskrečiųjų struktūrų“ modulio medžiaga. <https://moodle.ktu.edu/course/view.php?id=39>