Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №5

Поиск максимального паросочетания в двудольном графе

Выполнил:

студент гр. 953504

Кондрашов И.Д.

Проверил:

кандидат физико-математических наук

Дугинов О.И.

Минск 2022

1. **Пример работы программы**

Входные данные:

Text

Description automatically generated

Ответ:

Graphical user interface, text

Description automatically generated

1. **Исходный код программы (C#)**

Реализованные методы:

1. DFS(ref double[][] graph, int graphRowIndex, ref bool[] visited, ref int[] matching) – метод, реализующий поиск в глубину
2. MaximumMatching(Matrix<double> graph)– метод, выполняющий задание лабораторной(поиск максимального паросочетания)

using MathNet.Numerics.LinearAlgebra;

using System;

using System.Collections.Generic;

using System.Linq;

namespace Lr5

{

class Program

{

static void Main(string[] args)

{

Matrix<double> graph = Matrix<double>.Build.DenseOfArray(new double[,] { { 0, 1, 1, 0, 0 },

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

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

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

{ 0, 0, 0, 0, 0 } });

var match = MaximumMatching(graph);

foreach (var item in match)

{

Console.WriteLine($"Максимальное паросочетание: {String.Join(", ", item)}");

}

}

private static List<int[]> MaximumMatching(Matrix<double> graph)

{

int[] matching = new int[graph.RowCount];

Array.Fill(matching, -1);

bool[] visited = new bool[graph.RowCount];

var graphArray = graph.ToRowArrays();

for (int i = 0; i < graph.RowCount; i++)

{

Array.Fill(visited, false);

DFS(ref graphArray, i, ref visited, ref matching);

}

var match = matching.Select((i, j) => new int[] { i, j }).Where(v => !v.Contains(-1));

var matchArray = match.ToList();

return matchArray;

}

private static bool DFS(ref double[][] graph, int graphRowIndex, ref bool[] visited, ref int[] matching)

{

for (int i = 0; i < graph[graphRowIndex].Length; i++)

{

if (graph[graphRowIndex][i] == 1 && !visited[i])

{

visited[i] = true;

if (matching[i] == -1 || DFS(ref graph, matching[i], ref visited, ref matching))

{

matching[i] = graphRowIndex;

return true;

}

}

}

return false;

}

}

}