**Залізчук Данило**

**Група ІП-92**

**Лабораторна робота №6.2**

Ціль

Дослідити задачу розфарбування графів.

Завдання

Реалізувати програмне застосування (програму), яке виконує наступні функції. Причому на вхід програми подається вхідний файл з описом графу, зі структурою, яка вказана у лабораторній роботі №1 «Представлення графів».

1. Розв’язати задачу розфарбування графів. Програма визначає мінімальну кількість кольорів, в який можна розфарбувати граф, та виводить на екран список вершин із відповідними призначеними кольорами.

Контрольні питання

1. Як формулюється задача розфарбування графу?

2. Які методи дозволяють знайти мінімальну кількість кольорів для розфарбування графу?

**Код:**

using System;  
using System.Collections.Generic;  
using System.IO;  
using System.Linq;  
  
namespace DS\_IP92\_LR6.\_2\_ZalizchukD  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 string file = "input.txt";  
 Graph graph = new Graph(file);  
 graph.Coloring();  
  
 }  
 }  
  
 *// ================ КЛАСС "ГРАФ" ================* class Graph  
 {  
 private int n, m;  
 private int[,] mSmezh;  
 private int[] vertexPowers;  
 private bool[] visited;  
  
 *// ================ КОНСТРУКТОР, ЧТЕНИЕ ДАННЫХ О ГРАФЕ ================* public Graph(string path)  
 {  
 StreamReader sr = new StreamReader(path);  
 string read = sr.ReadLine();  
 string[] temp = read.Split(' ');  
 n = Convert.ToInt32(temp[0]);  
 m = Convert.ToInt32(temp[1]);  
 mSmezh = new int[n, n];  
 vertexPowers = new int[n];  
 visited = new bool[n];  
  
 for (int i = 0; i < m; i++)  
 {  
 read = sr.ReadLine();  
 temp = read.Split(' ');  
 int a = Convert.ToInt32(temp[0]) - 1, b = Convert.ToInt32(temp[1]) - 1;  
 mSmezh[a, b] = 1;  
 mSmezh[b, a] = 1;  
 }  
  
 for (int i = 0; i < n; i++)  
 for (int j = 0; j < n; j++)  
 if (mSmezh[i, j] == 1)  
 vertexPowers[i]++;  
   
 sr.Close();  
 }  
   
 *// ================ СОРТИРОВКА ВЕРШИН ПО УБЫВАНИЮ СТЕПЕНЕЙ, НО ПО ВОЗРАСТАНИЮ НОМЕРОВ ================* private void sortVertexPowers()  
 {  
 List<int> array = new List<int>();  
  
 int max = vertexPowers.Max();  
  
 while (max > 0)  
 {  
 for (int i = 0; i < n; i++)  
 if (vertexPowers[i] == max)  
 array.Add(i);  
  
 max--;  
 }  
  
 vertexPowers = array.ToArray();  
 }  
  
 *// ================ РАСКРАСКА ВЕРШИН ГРАФА ================* public void Coloring()  
 {  
 sortVertexPowers();  
   
 int trueCount = 0, *// счетчик закрашенных вершин* color = -1, *// номер цвета* count; *// счетчик несмежных вершин* List<List<int>> colors = new List<List<int>>(); *// список цветов и вершин каждого цвета* while (trueCount != n)  
 {  
 color++;  
 for (int i = 0; i < n; i++)  
 {  
 if (visited[vertexPowers[i]] == false) *// если вершина не закрашена* {  
 colors.Add(new List<int>()); *// добавляем новый цвет* colors[color].Add(vertexPowers[i]); *// закрашиваем вершину в этот цвет* visited[vertexPowers[i]] = true;  
 trueCount++;  
 break;  
 }  
 }  
  
 for (int i = 0; i < n; i++)  
 {  
 count = 0;  
 for (int j = 0; j < colors[color].Count; j++)  
 if (visited[vertexPowers[i]] == false)  
 if (mSmezh[vertexPowers[i], colors[color][j]] == 0)  
 count++; *// проверяем, чтобы вершина не была смежна с каждой из закрашенных вершин* if (count == colors[color].Count) *// если не смежна - закрашиваем* {  
 colors[color].Add(vertexPowers[i]);  
 visited[vertexPowers[i]] = true;  
 trueCount++;  
 }  
 }  
   
 }  
   
 Console.WriteLine($"Минимальное количество цветов - {colors.Count}\n");  
 for (int i = 0; i < colors.Count; i++)  
 {  
 Console.Write($"{i + 1} цвет: ");  
 for (int j = 0; j < colors[i].Count; j++)  
 {  
 Console.Write($"{colors[i][j] + 1} ");  
 }  
 Console.WriteLine();  
 }  
   
 }  
  
 }  
   
}

**Результат:**



