**Залізчук Данило**

**Група ІП-92**

**Лабораторна робота №6.1**

Ціль

1. Розглянути задачу пошуку ейлеревих циклів у графі.

2. Розглянути задачу пошуку гамільтонових циклів у графі.

Завдання

1. 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». *При реалізації вважати, що заданий граф є неорієнтованим*.

1) Знайти ейлерів цикл в графі. Програма виводить на екран знайдений ейлерів цикл або сповіщає про його відсутність у графі.

2) Знайти ейлерів маршрут в графі. У випадку, якщо ейлеревого циклу в графі не існує, то спробувати знайти ейлерів маршрут. Програма виводить на екран знайдений ейлерів маршрут, якщо він є, або сповіщає про його відсутність.

2. Реалізувати програмне застосування (програму), яке виконує наступні функції. Причому на вхід програми подається вхідний файл з описом графу, зі структурою, яка вказана у практичному завданні №1 «Представлення графів». *При реалізації вважати, що заданий граф є неорієнтованим*.

1. Знайти гамільтонів цикл в графі. Програма виводить на екран знайдений гамільтонів цикл або сповіщає про його відсутність у графі.

2. Знайти гамільтонів маршрут в графі. У випадку, якщо гамільтонового циклу в графі не існує, то спробувати знайти гамільтонів маршрут. Програма виводить на екран знайдений гамільтонів маршрут, якщо він є, або сповіщає про його відсутність.

Контрольні питання

1. Що таке ейлерів цикл?

2. Які умови існування ейлеревого циклу в графі?

3. Що таке ейлерів маршрут?

4. Які умови існування ейлеревого маршруту в графі?

5. Що таке гамільтонів цикл, маршрут?

6. Які умови існування гамільтонового циклу в граф

**Код:**

using System;  
using static System.IO.Directory;  
using System.IO;  
using System.Collections.Generic;  
using System.Linq;  
using System.Threading;  
  
namespace DS\_IP92\_LR6.\_1\_ZalizchukD  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 string path = Directory.GetCurrentDirectory() + "\\input.txt";  
 string choice;  
 Graph graph = new Graph(path);  
   
 Console.WriteLine("1. Проверить наличие Эйлерова пути\n2. Проверить наличие Гамильтонова пути");  
 choice = Console.ReadLine();  
  
 if (choice == "1")  
 {  
 Console.Clear();  
 graph.Eiler();  
 } else if (choice == "2")  
 {  
 Console.Clear();  
 graph.Hamiltonian();  
 }  
 else  
 {  
 Console.WriteLine("Ошибка: неверный ввод.");  
 Environment.Exit(0);  
 }  
 }  
  
 }  
  
 *// ================= КЛАСС "ГРАФ" =================* class Graph  
 {  
 private int n, m, start;  
 private int[,] mSmezh;  
 private int[] vertexPowers;  
 private bool eilerCycle = true, eilerPath, hamiltonianCycle;  
 private bool[] visited;  
 private List<int> hamiltonianPath;  
  
 private Stack<int> stack;  
  
 *// ================= ЧТЕНИЕ ПРОСТЫХ 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];  
 hamiltonianPath = new List<int>();  
  
 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]++;  
   
 CheckEiler();  
 CheckHamiltonian();  
 }  
  
 *// ================= ПРОВЕРКА НА ЭЙЛЕРОВОСТЬ =================* private void CheckEiler()  
 {  
 int count = 0;  
 foreach (var power in vertexPowers)  
 if (power % 2 != 0)  
 {  
 eilerCycle = false;  
 count++;  
 }  
  
 if (eilerCycle == true)  
 eilerPath = true;  
 else if (count == 2)  
 eilerPath = true;  
 else  
 eilerPath = false;  
 }  
  
 *// ================= ПРОВЕРКА НА ГАМИЛЬТОНОВ ЦИКЛ =================* private void CheckHamiltonian()  
 {  
 hamiltonianCycle = true;  
 if (n >= 3)  
 {  
 for (int i = 0; i < n; i++)  
 if (vertexPowers[i] < n / 2)  
 {  
 hamiltonianCycle = false;  
 break;  
 }  
  
 }  
 else  
 hamiltonianCycle = false;  
 }  
  
 *// ================= ЭЙЛЕРОВ ПУТЬ =================* public void Eiler()  
 {  
 int v = 0;  
 if (!eilerCycle)  
 {  
 if (eilerPath)  
 {  
 for (int i = 0; i < n; i++)  
 if (vertexPowers[i] % 2 != 0)  
 {  
 Console.Write("Найден Эйлеров путь: ");  
 v = i;  
 break;  
 }  
 }  
 else  
 {  
 Console.WriteLine("Эйлеров путь отсутствует.");  
 return;  
 }  
 }  
 else  
 Console.Write("Найден Эйлеров цикл: ");  
  
 int[,] rebra = mSmezh;  
 stack = new Stack<int>();  
 stack.Push(v);  
 while (stack.Count != 0)  
 {  
 v = stack.Peek();  
 for (int i = 0; i < n; i++)  
 if (rebra[v, i] == 1)  
 {  
 stack.Push(i);  
 rebra[v, i] = 0;  
 rebra[i, v] = 0;  
 break;  
 }  
  
 if (v == stack.Peek())  
 {  
 stack.Pop();  
 Console.Write($"{v + 1} ");  
 }  
 }  
 Console.WriteLine();  
   
 }  
  
 *// ================= ГАМИЛЬТОНОВ ПУТЬ =================* public void Hamiltonian()  
 {  
 if (hamiltonianCycle)  
 {  
 for (int i = 0; i < n; i++)  
 {  
 start = i;  
 HamiltonianCycle(start);  
 }  
   
 Console.WriteLine("Гамильтонов путь отсутствует.");  
 }  
 else  
 {  
 for (int i = 0; i < n; i++)  
 HamiltonianPath(i);  
   
 Console.WriteLine("Гамильтонов путь отсутствует.");  
 }  
 }  
  
 *// ================= ГАМИЛЬТОНОВ ЦИКЛ, БЕКТРЕКИНГ =================* private void HamiltonianCycle(int v)  
 {  
 if (hamiltonianPath.Count == n && mSmezh[hamiltonianPath[0], hamiltonianPath[n - 1]] == 1)  
 {  
 hamiltonianPath.Add(hamiltonianPath[0]);  
 Console.Write("Найден Гамильтонов цикл: ");  
 foreach (var el in hamiltonianPath)  
 Console.Write($"{el + 1} ");  
   
 Console.WriteLine();  
 Environment.Exit(0);  
 }  
  
 for (int i = 0; i < n; i++)  
 {  
 if (mSmezh[v, i] == 1 && !visited[i])  
 {  
 visited[i] = true;  
 hamiltonianPath.Add(i);  
 HamiltonianCycle(i);  
  
 visited[i] = false;  
 hamiltonianPath.RemoveAt(hamiltonianPath.Count - 1);  
 }  
 }  
  
 }  
  
 *// ================= ГАМИЛЬТОНОВ ПУТЬ, БЕКТРЕКИНГ =================* private void HamiltonianPath(int v)  
 {  
 if (hamiltonianPath.Count == n)  
 {  
 Console.Write("Найден Гамильтонов путь: ");  
 foreach (var el in hamiltonianPath)  
 Console.Write($"{el + 1} ");  
   
 Console.WriteLine();  
 Environment.Exit(0);  
 }  
  
 for (int i = 0; i < n; i++)  
 {  
 if (mSmezh[v, i] == 1 && !visited[i])  
 {  
 visited[i] = true;  
 hamiltonianPath.Add(i);  
 HamiltonianPath(i);  
  
 visited[i] = false;  
 hamiltonianPath.RemoveAt(hamiltonianPath.Count - 1);  
 }  
 }  
  
 }  
  
 }  
}

**Результат:**









