**Залізчук Данило**

**Група ІП-92**

**Лабораторна робота №7.1**

Ціль

Дослідити кодування дерев за допомогою коду Прюфера.

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 побудови кода Прюфера. Програма визначає код Прюфера для заданого дерева та виводить його на екран.

2. Розв’язати задачу побудови дерева з коду Прюфера. Програма визначає матрицю суміжності для заданого коду Прюфера (який задається на вхід у вигляді рядка 1 3 3 1 4 7 4) та виводить матрицю суміжності на екран.

Контрольні питання

1. Як за заданим деревом побудувати код Прюфера?

2. Як за кодом Прюфера побудувати дерево?

**Код:**

using System;  
using System.Collections.Generic;  
using System.IO;  
using System.Linq;  
  
namespace DS\_IP92\_LR7.\_1\_ZalizchukD  
{  
 class Program  
 {  
 static void Main(string[] args)  
 {  
 string input = "graph.txt", prufer = "prufer.txt", choice;  
 Graph graph;  
 Console.WriteLine("1. Дерево в код Прюфера\n2. Код Прюфера в дерево");  
 choice = Console.ReadLine();  
 if (choice == "1")  
 {  
 Console.Clear();  
 graph = new Graph(input);  
 graph.TreeToPrufer();  
 }  
 else if (choice == "2")  
 {  
 Console.Clear();  
 graph = new Graph(prufer);  
 graph.PruferToTree();  
 }  
 else  
 {  
 Console.WriteLine("Ошибка: неверный ввод.");  
 Environment.Exit(0);  
 }  
 }  
 }  
  
 *// ================ КЛАСС "ГРАФ" ================* class Graph  
 {  
 private int n, m;  
 private int[,] mSmezh;  
 private int[] vertexPowers;  
 private List<int> pruferCode = new List<int>(), vertices;  
  
 *// ================ КОНСТРУКТОР, ЧТЕНИЕ ДАННЫХ О ГРАФЕ ================* public Graph(string path)  
 {  
 StreamReader sr = new StreamReader(path);  
  
 if (path == "graph.txt")  
 {  
 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];  
  
 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]++;  
 }  
 else if (path == "prufer.txt")  
 {  
 string read = sr.ReadLine();  
 string[] temp = read.Split(' ');  
 foreach (var el in temp)  
 pruferCode.Add(Convert.ToInt32(el) - 1);  
  
 n = pruferCode.Count + 2;  
 mSmezh = new int[n, n];  
 vertices = new List<int>();  
 for (int i = 0; i < n; i++)  
 vertices.Add(i);  
   
 }  
  
 }  
   
 *// ================ ПОЛУЧЕНИЕ КОДА ПРЮФЕРА ИЗ ДЕРЕВА ================* public void TreeToPrufer()  
 {  
 for (int i = 0; i < n - 2; i++)  
 {  
 for (int j = 0; j < n; j++)  
 {  
 if (vertexPowers[j] == 1) *// если с вершина - листок с наименьшим номером* {  
 for (int k = 0; k < n; k++)  
 {  
 if (mSmezh[j, k] == 1) *// ищем ей инцидентную вершину* {  
 mSmezh[j, k] = 0; *// удаляем ребро* mSmezh[k, j] = 0;  
 pruferCode.Add(k); *// добавляем инцидентную вершину в код* vertexPowers[k]--;  
 break;  
 }  
 }  
  
 vertexPowers[j]--;  
 break;  
 }  
 }  
 }  
   
 Console.Write("Код Прюфера: ");  
 foreach (var el in pruferCode)  
 {  
 Console.Write($"{el + 1} ");  
 }  
 }  
  
 *// ================ ПОЛУЧЕНИЕ ДЕРЕВА ИЗ КОДА ПРЮФЕРА ================* public void PruferToTree()  
 {  
 for (int i = 0; i < n - 2; i++)  
 {  
 for (int j = 0; j < vertices.Count; j++)  
 {  
 if (!pruferCode.Contains(vertices[j])) *// ищем номер наименьшей вершины, которой нет в коде* {  
 mSmezh[vertices[j], pruferCode[0]] = 1; *// создаем ребро с первой вершиной из кода* mSmezh[pruferCode[0], vertices[j]] = 1;  
 vertices.RemoveAt(j); *// удаляем задействованные вершины* pruferCode.RemoveAt(0);   
 break;  
 }  
 }  
 }  
  
 *// оставшиеся 2 вершины в списке вершин составляют последнее ребро* mSmezh[vertices[0], vertices[1]] = 1;  
 mSmezh[vertices[1], vertices[0]] = 1;  
   
 Console.WriteLine("Матрица смежности полученного дерева:");  
 for (int i = 0; i < n; i++)  
 {  
 for (int j = 0; j < n; j++)  
 Console.Write("{0,4}", mSmezh[i, j]);  
   
 Console.WriteLine();  
 }  
   
 }  
   
 }  
   
}

**Результат:**











