Міністерство освіти та науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

ЗВІТ

до лабораторної роботи №1

з дисципліни «Основи Web-програмування»

Виконав: студент 2 курсу групи ІП-64 , ФІОТ

Царук Володимир

Номер залікової: 6429 Номер в списку: 27

Київ – 2017

**Код програми:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace lab1

{

class Program

{

static int[,] Rand(int n) //генерування матриці

{

int[,] M = new int[n, n];

Random rand = new Random();

for (int i = 0; i < n - 1; i++)

for (int j = i + 1; j < n; j++)

{

int x = rand.Next(2);

M[i, j] = x;

M[j, i] = x;

}

return M;

}

static int[,] input(int n) //введеня матриці

{

Console.WriteLine("Введите: \n1 - ввод матрицы с клавиатуры \n2 - сгенерировать матрицу");

int v = Convert.ToInt32(Console.ReadLine());

int[,] M = new int[n, n];

if (v == 1) //введення з клавіатури

{

Console.WriteLine("Введите матрицу:");

for (int i = 0; i < n; i++)

{

string[] s = Console.ReadLine().Split(' ');

for (int j = 0; j < n; j++)

M[i, j] = Convert.ToInt32(s[j]);

}

}

else //генерування випадкової

{

M = Rand(n);

output(n, M);

}

return M;

}

static void output(int n, int[,] M) //вивід матриці в консоль

{

for (int i = 0; i < n; i++)

{

for (int j = 0; j < n; j++)

Console.Write(M[i, j] + " ");

Console.WriteLine();

}

}

static void Delete(int n, int k, int[,] M) //видалення кільця (вершини)

{

for (int i = 0; i < n; i++)

{

M[k, i] = 0;

M[i, k] = 0;

}

}

static void copy(List<int> list1, List<int> list2) //копіювання цепочки(списка), при умові що вона має на данний момент макс довжину

{

list1.Clear();

for (int i = 0; i < list2.Count; i++)

list1.Add(list2[i]);

}

static void Search(int n, int start, int[,] M, List<int> listok, List<int> maxlistok) //пошук нового кільця для цепочки

{

for (int i = 0; i < n; i++)//розглядаємо всі можливі вершини

if (M[start, i] == 1) //якщо є звязок між кільцями

{

int[,] M1 = (int[,])M.Clone(); //робимо копію матриці

listok.Add(i); //додавання елемента(кільця) до цепочки

if (listok.Count > maxlistok.Count) //знайшли нову найдовшу цепочку

{

copy(maxlistok, listok);

}

for (int j = 0; j < n; j++) //видаляэмо всі вершини, окрім тої, в яку переходимо

if (j != i && M1[start, j] == 1)

Delete(n, j, M1);

Delete(n, start, M1);

Search(n, i, M1, listok, maxlistok); //пошук шляхів з нової вершини

}

listok.RemoveAt(listok.Count - 1);

}

static int[,] NewMatr(int n, int[,] M, List<int> list) //створення матриці, що задає цепочку

{

int[,] Matr = (int[,])M.Clone();

int[] arr = new int[n];

for (int i = 0; i < list.Count; i++)

arr[list[i]] = 1;

for (int i = 0; i < n; i++)

if (arr[i] == 0)

Delete(n, i, Matr);

return Matr;

}

static void Main(string[] args)

{

Console.WriteLine("Введите размерность матрицы:");

int n = Convert.ToInt32(Console.ReadLine());

int[,] M = input(n); // вводимо матрицю

List<int> listok = new List<int> { };//цепочка, що розглядається

List<int> maxlistok = new List<int> { };// цепочка макс довжини

for (int i = 0; i < n; i++)

{

listok.Add(i);

Search(n, i, (int[,])M.Clone(), listok, maxlistok);//пошук цепочки

}

Console.WriteLine();

Console.WriteLine("Кол-во удаленных колец: " + (n - maxlistok.Count));

Console.WriteLine("Имеем цепочку:");

for (int i = 0; i < maxlistok.Count; i++)

Console.Write(maxlistok[i] + " ");

Console.WriteLine();

Console.WriteLine("Mатрица даной цепочки:");

output(n, NewMatr(n, M, maxlistok));

Console.ReadKey();

}

}

}

Iлюстрації

